HP 3000 Computer Systems

HEWLETT hp PACKARD

A family of compatible business systems for distributed data processing

General Information Manual

HP 3000 Series 30

HP 3000 Series 33

HP 3000 Series III

HP 3000 Computer Systems
General Information Manual

HEWLETT hp PACKARD

19447 Pruneridge Avenue, Cupertino, CA 95014

Notice

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Preface

Today's business decisions cannot be based on yesterday's data. Immediate access to the most up-to-date information is a necessity for the financial planning, sales forecasting, production scheduling, and other complex tasks which are part of each business day. The HP 3000 interactive business computer systems are uniquely qualified to meet these demands for accurate, timely information. Designed specifically for terminal-oriented business data processing, the HP 3000 systems are based on an integrated hardware and software concept. HP 3000 computer systems are delivered ready to run application programs. Complete data base management and data entry and inquiry facilities are included to make the basic HP 3000 system a fully capable and operational system.

For manufacturers, Hewlett-Packard offers materials requirements planning (MRP) software. These comprehensive on-line application packages are flexible enough to be tailored to your specific needs, yet contain the years of experience HP has gained in refining MRP techniques in its own manufacturing operations. Materials planning and control pay tangible benefits in reducing inventory investment and improving on-time shipments.

For in-house software development, Hewlett-Packard provides tools for efficient interactive program development. With six programming languages, debugging aids, programming utilities, and high level procedures for data base management, data entry, data inquiry and other programming tasks, programmer productivity on the HP 3000 is high.

Advanced distributed processing capabilities through a wide variety of data communications products allow HP 3000 systems to bring information directly to the people who need it, when they need it.

This manual describes the general capabilities and features of the HP 3000 Series 30, Series 33, and Series III computer systems. Detailed specifications of the individual hardware and software products offered with the system are also presented. The manual is divided into two sections: Section I, the System Overview, and Section II, the Reference Sheets.

- Section I is a thorough discussion of the HP 3000 transaction processing environment, manufacturing application programs, data communications, operating system and system architecture.
- Section II contains reference sheet specifications of the hardware systems, Fundamental Operating Software, optional software subsystems and hardware peripherals, software/hardware support services and training course descriptions.
- The appendices outline the various operating system and machine level commands and intrinsics, and expand on the hardware features of the computer systems.

Contents

SECTION I. GENERAL INFORMATION

Chapter 1. System Introduction

Chapter 2. Transaction Processing

Chapter 3. Manufacturing Applications

Chapter 4. Data Communications

Chapter 5. The Operating System—MPE III

Chapter 6. System Architecture

SECTION II. REFERENCE SHEETS

Fundamental Operating Software

Languages

MFG/3000 Manufacturing Systems

Data Communications

Software Subsystems

Hardware

Peripherals

Storage Media

Terminals

Printers

Other Peripherals

Appendices

Index

Chapter 1

SYSTEM INTRODUCTION

MANAGEMENT OVERVIEW
SYSTEM COMPONENTS
FUNDAMENTAL OPERATING SOFTWARE
LANGUAGES
DATA COMMUNICATIONS
MANUFACTURING APPLICATIONS
DOCUMENTATION
TRAINING
SOFTWARE SUPPORT SERVICES
HARDWARE SUPPORT SERVICES
HP GENERAL SYSTEMS USERS GROUP

MANAGEMENT OVERVIEW

HP 3000 computer systems are a compatible family of interactive, data-base oriented business processing systems. They have an integrated hardware and software design ideally suited to multi-purpose business data processing and dedicated manufacturing applications. Their advanced design provides a wide range of user-oriented capabilities that can be utilized with equal ease on both stand-alone systems and those in distributed processing networks.

HP 3000 systems can be configured in three different ways: as a ready-to-run production system, as a program development system, and as an integrated element of a distributed data processing network.

Ready-To-Run Production Systems

HP 3000 computers are delivered as complete systems with all the hardware and software necessary to run and operate applications programs immediately. In addition to hardware, every HP 3000 computer system includes Fundamental Operating Software (FOS) which makes it a completely functional production system. Included in FOS is the Multi-Programming Executive operating system (MPE) that schedules, monitors, and controls processing, and a complete set of software subsystems for data management, data entry and other capabilities needed in business processing applications. The Fundamental Operating Software enables you to immediately run application programs developed and compiled on any HP 3000 without installing any additional software on the executing system other than the program itself. One HP 3000 can satisfy the program development needs of an entire network of HP 3000 Computer Systems.

Program Development Systems

Optional software subsystems, such as language compilers, are available for program development. Highly compatible interfaces between subsystems have been created so that programmers can use the Fundamental Operating Software, in conjunction with the compilers, as a high-level program development tool. The subsystems of FOS include a wide range of high-level procedures for data management and data entry that are callable from any of the HP 3000 programming languages. This enables programmers to give application programs comprehensive capabilities and makes program development on the HP 3000 quick and convenient.

Distributed Data Processing Systems

Data communication subsystems are also available to enable you to set up distributed processing networks that conform to the way you want to do business. A variety of terminal communications products allow you to place terminals wherever you wish without regard to distance or location. HP Distributed Systems software permits sharing of resources between HP 3000 computer systems and other Hewlett-Packard computers. HP 3000 systems can also provide batch access to IBM mainframe computers. These capabilities are supplied as high level services which free the programmer from complex tasks and allow you to place the processing power where you need it—regardless of the level of expertise at the remote site.

Combined Function Systems

The above three HP 3000 configurations are by no means mutually exclusive. All HP 3000 computer systems can simultaneously perform transaction processing operations, interactive program development, batch processing and data communications (see Figure 1-1).

You can distribute your processing according to functional needs or geographical location while retaining the degree of control and security you desire. At each location, you can specify the function and capability you want your HP 3000 to have: a single dedicated application, multiple applications, program development or a combined general use system.

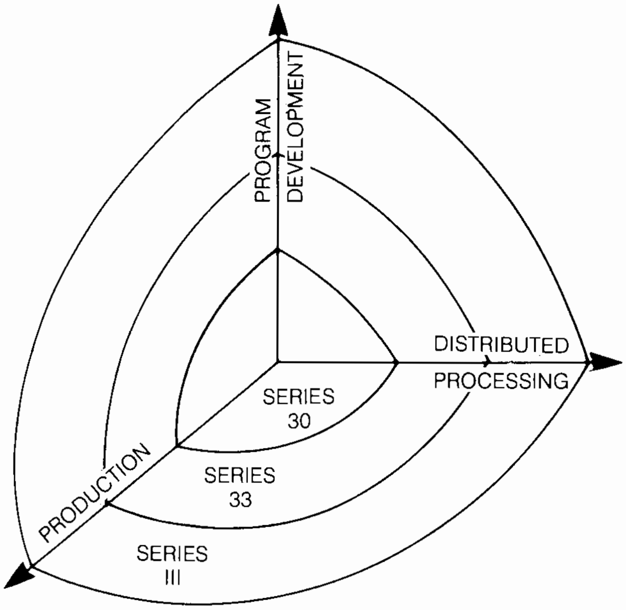
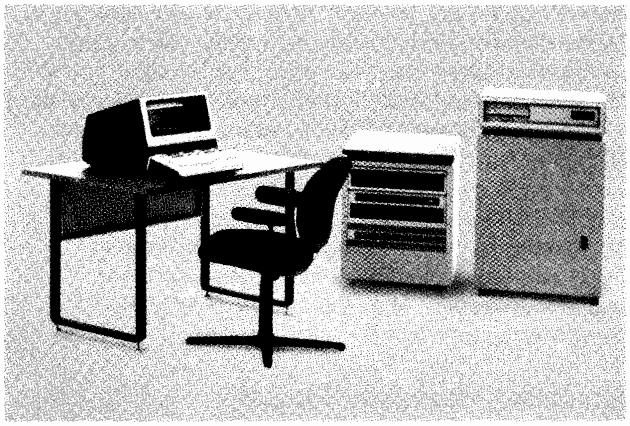
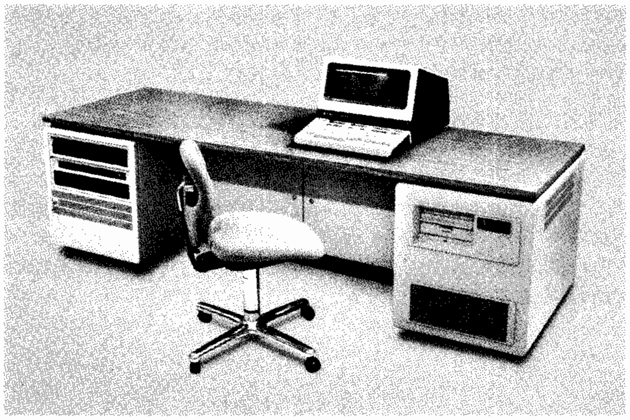


Figure 1-1. Each HP 3000 computer system can be configured to provide the exact mix of capabilities you require along three functional dimensions: Production, Distributed Data Processing, and Program Development.

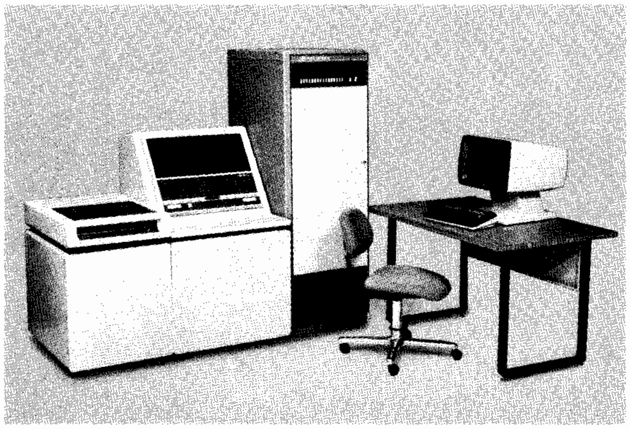
HP 3000 computer systems are available in three models:



Series 30



Series 33



Series III

A Commitment to Your Success

Hewlett-Packard is committed to making our products easy to use by non-data processing professionals, whether through HP supplied application programs or the interactive data access facilities for terminal users. Also, the wide range of program development tools makes it easy for programmers to create user-oriented programs.

Superior hardware and software is not enough. Comprehensive support services are available to assist you in maintaining all phases of your operation. Among them are the Remote System Verification Program (RSVP) available on the Series 30 and 33 that permits these systems to be diagnosed remotely from an HP office by system expert. Another innovative service is the Phone-In Consulting Service (PICS) that enables your System Manager to receive immediate assistance via telephone to resolve software related problems. Personal attention from a well trained Systems Engineer is available whenever you need it. A full range of documentation, training, consulting, and support programs are offered to insure that you have the assistance you need to be successful. Hewlett-Packard supplies a full range of support services and gives you the flexibility to tailor them to your needs.

Hewlett-Packard documentation, training and support services are discussed at the end of this chapter and in the reference sheets in Section II.

SYSTEM COMPONENTS

An HP 3000 computer system consists of the system hardware, the fundamental operating software, and additional software subsystems, all of which contribute to the easy development and operation of user applications. Hewlett-Packard also provides a set of ready-to-use applications developed for a manufacturing environment. These components and their relationships are illustrated in Figure 1-2.

An Integrated System

The fundamental HP 3000 computer system consists of the system hardware, and the Fundamental Operating Software that includes the MPE operating system, utility programs, and data management, and data entry subsystems.

The HP 3000 software is integrated with an advanced hardware design that includes stack architecture, variable length code segmentation, a hardware-assisted virtual memory scheme, user protection, and dynamic storage allocation. Hardware and software work together, with hardware performing many of the operations conventionally performed by software, such as interrupts or subroutine calls.

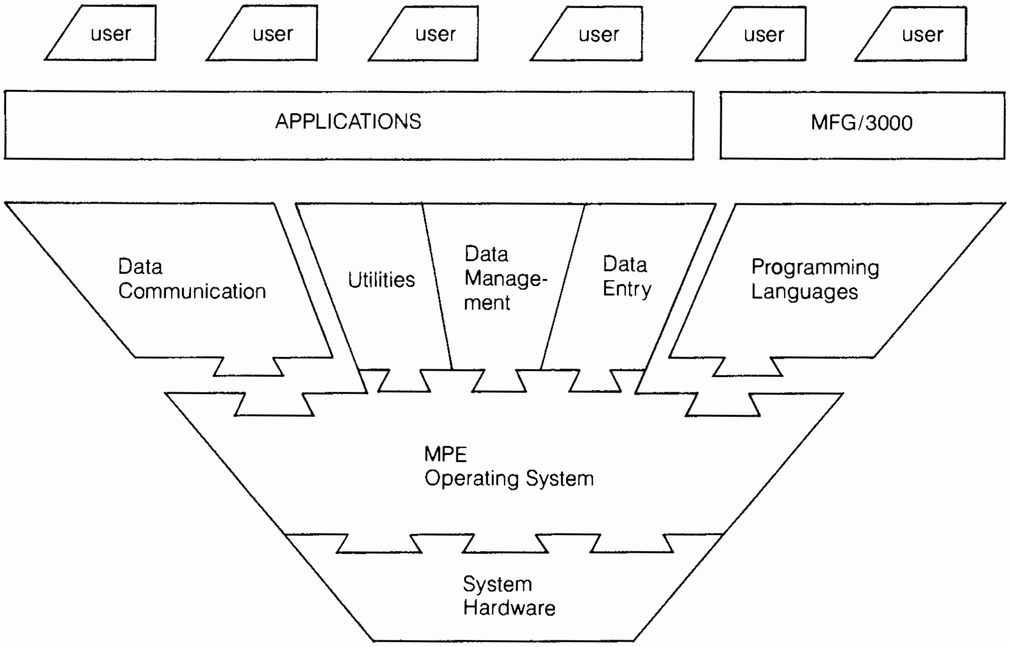


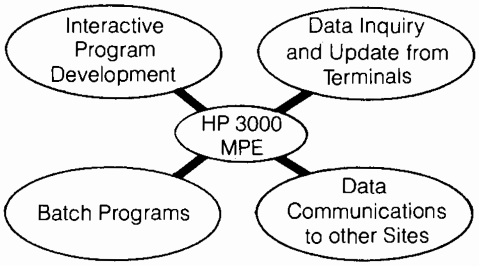
Figure 1-2. HP 3000 Computer System: System Components

System Hardware

The HP 3000 incorporates many features otherwise found only on very large computer systems, including:

- Stack Architecture—provides private, hardware protected data storage for each user plus automatic movement of this data to and from the central processor.
- Virtual Memory—consists of main memory plus an extensive storage area on magnetic disc to provide a total memory far exceeding the main memory size.
- Separation of Code and Data—strictly separate domains for code and data permits code sharing and re-entrant execution while maintaining data privacy.
- Fault Control Memory—high speed semiconductor memory modules provide automatic fault detection and single-bit correction.
- Microcode—more than 200 instructions are microcoded operations in read-only memory, plus a full set of microcoded system operations.
- Concurrent I/O and CPU operation—allows input/output to be performed concurrently with CPU operation.

Although the same software is used by all HP 3000 computer systems, specific system hardware differs for the Series 30, Series 33, and Series III. Series 30 and Series 33 are described in Appendix D, Series III in Appendix F. Generally, HP 3000 system hardware includes the central processor unit (CPU), main memory, and various peripheral devices available on each system series. Refer to Chapter 6 for a full discussion of the system architecture, and the HP 3000 Price/Configuration Guide for configuration details.


Interactive Program Development
Data Inquiry and Update from Terminals
HP 3000 MPE
Batch Programs
Data Communications to other Sites

All Simultaneously

Figure 1-3. HP 3000 MPE Operating System controls and supervises all processing.

FUNDAMENTAL OPERATING SOFTWARE

MPE Operating System

The Multiprogramming Executive operating system (MPE III) is the disc-based software system which supervises the processing of all programs that run on the HP 3000. MPE dynamically allocates such system resources as main memory, the central processor, and peripheral devices to each program as needed. In addition, MPE coordinates all user interaction with the system, providing an easy to use command language interface and a powerful programmatic interface in the form of intrinsics and a versatile file system.

MPE monitors and controls program input, compilation, run preparation, loading, execution and output. It also controls the order in which programs are executed and allocates and maintains usage records of the hardware and software resources they require. By relieving you of many program control, input/output, and other housekeeping responsibilities, MPE makes the HP 3000 extremely easy to use.

The major features of the operating system are:

- Multiprogramming: concurrent transaction processing, data communications, on-line program development, and batch processing
- Virtual memory
- Concurrent multilingual capability
- Sharable code
- Device independent file system
- Input and output spooling
- Complete system security and automatic accounting of resources
- Friendly, powerful command language
- Complete interactive terminal management, both local and remote
- Automatic system restart after power failure

Details of the MPE operating system are presented in Chapter 5.

Data Entry

Standard in the Fundamental Operating Software of each HP 3000 is:

- **HP VIEW/3000**—A comprehensive data entry subsystem that provides both a ready-to-use data entry program, ENTRY, and a programmatic interface for your terminal-oriented application programs. It includes facilities for immediate on-line entry and modification of data, a wide range of data editing and validation, record reformatting, and interactive forms design and forms management.

Data Management

The HP 3000 has extensive data management facilities with both sequential and random access methods included as part of the MPE III operating system. Additional access methods and data management capabilities included in the Fundamental Operating Software are:

- **KSAM/3000**—Keyed Sequential Access Method, an indexed file subsystem providing one primary and up to 15 alternate keys, with retrieval based upon the value of the keys.

- **IMAGE/3000**—Data Base Management System allows information to be logically related between data sets (files). It facilitates data independence, data security and integrity and minimizes data redundancy. A transaction logging facility provides an audit trail of data base activity, and a security feature permits locking at the item level. Remote data base access through HP data communication software is also included.

- **QUERY/3000**—Complements IMAGE by supplying an English-like inquiry language for entry, updating and reporting with IMAGE data bases.

Utilities

A set of utility programs, standard on each HP 3000, eases program development and file manipulation and aids in system administration. The utilities included in the Fundamental Operating Software are:

- **EDIT/3000**—A powerful and easy to use text editor
- **FCOPY/3000**—A program for general file copying
- **SORT-MERGE/3000**—A facility for ordering records in a file and merging sorted files
- **System Utilities**—provide administrative controls, reports on system resources, and other special purpose capabilities.
- Facility to execute compiled programs without the source language compiler on the system (except those written in APL\3000).

The utilities, data entry and data management software subsystems are described more fully in Chapter 2.

ADDITIONAL SOFTWARE SUBSYSTEMS

In addition to the Fundamental Operating Software, a full set of programming languages and extensive data communication facilities are offered for use on HP 3000 computer systems.

Languages

Designed to provide you with language flexibility, the HP 3000 offers six high-level programming languages which let you select the language best suited to the task. Programs can be written in:

- **COBOL**
- **RPG**
- **FORTRAN**
- **BASIC**
- **APL** (only on Series III)
- **SPL** (Systems Programming Language—a high level machine dependent language that takes full advantage of HP 3000 design features)

Applications written in different languages, or a combination of languages, can be run simultaneously. Data files and peripheral devices can be used in common by programs in any language without program changes.

Within MPE and the other subsystems of the Fundamental Operating Software reside high level procedures for terminal handling, data entry, and data management. These procedures, or intrinsics as HP refers to them, can easily be called from HP 3000 programming languages. Thus, the addition of a compiler is all that is needed to equip the HP 3000 as a program development machine. Because FOS is standard on all HP 3000's, compiled programs run on all HP 3000 computer systems and can take full advantage of the data entry, data management, and other high level program calls.

The languages are discussed in detail in Chapter 2.

Data Communications

HP 3000 Computer Systems employ a network architecture called Hewlett-Packard Distributed Systems Network (HP-DSN). HP-DSN allows diverse computer systems to be linked into information processing networks. Through the HP-DSN communications product family, data processing can be distributed among geographically or functionally dispersed computers and terminals. Flexible enough to support either a latticework of autonomous computer systems or a hierarchical configuration of satellite processors to a central computer, HP-DSN gives you the freedom to design a data processing network that uniquely fits your organizational structure. HP-DSN stresses interactive communication among processors and is implemented as a set of high level services that free the programmer from complex coding tasks and the concerns of hardware interfaces.

Data communication subsystems extend the basic asynchronous terminal communications under MPE to include:

- **MTS/3000**—synchronous multipoint terminal communications that allow users to have up to 32 terminals share a single remote or local line at speeds up to 9600 bps.

The variety of communications subsystems that link HP 3000 computers to IBM-type mainframe systems include:

- **RJE/3000**—IBM 2780/3780 emulation for remote job entry.
- **MRJE/3000**—IBM HASP/JES/ASP multileaving workstation emulation for remote job entry by multiple users simultaneously.

In addition, data communications between Hewlett-Packard computer systems in an HP Distributed Systems Network is implemented through:

- **DS/3000**—Distributed Systems/3000, provides remote file and peripheral access, interactive processing and resource sharing among HP 3000 computers and other Hewlett-Packard computer systems.

These data communications subsystems are discussed in Chapter 4.

Manufacturing Application Software

MFG/3000 is an integrated on-line system developed by Hewlett-Packard for managing the materials planning and control function of a manufacturing operation. MFG/3000 software is applicable to most manufacturing applications, however, it is ideally suited for a discrete manufacturer who assembles standard, multi-piece products in lots.

An important feature of MFG/3000 software products is the ease and speed with which the user can enter, retrieve, and modify data through an interactive terminal. The user has the facility to call up a variety of "menu-like" screens on the terminal to perform many tasks associated with materials planning and control.

With these products, most transaction processing operations employ a technique by which edits are stored in tables. These edit tables are used to enforce update security restrictions, to add or change screen format definitions, and to modify the editing rules which apply to any particular data field. Changing the edit tables requires no programming modifications and can be done by the system administrator through an on-line program.

All MFG/3000 products are supplied with predefined screen formats, transaction edit criteria, and retrieval and report formats. They can be run "as is" or, if required, many external features can be modified by non-programmers. Items affecting terminal users that are easily modifiable include:

—Default values for certain transactions

—Screen sequencing

—Certain field editing characteristics

—Most transaction/screen formats

—Originator numbers and capabilities

—Documentation screens

MFG/3000 consists of the following products:

- **EDC/3000 (Engineering Data Control)**—maintains an IMAGE/3000 data base that contains bills of materials, current and standard cost data, standard routings, and descriptive information about each item.
- **IOS/3000 (Inventory and Order Status)**—maintains an IMAGE/3000 data base that contains information about current inventory balances, scheduled issues from inventory, and scheduled receipts from purchase and work orders.
- **MRP/3000 (Material Requirements Planning)**—uses information from EDC/3000 and IOS/3000 to generate the materials plan with recommendations on what and how much material to order and when to order it.
- **SPC/3000 (Standard Product Costing)**—uses information from EDC/3000 to calculate current product costs and to set standard costs.

Users

The final element in an HP 3000 computer system is the user. You interact directly with the computer system through the software subsystems and the operating system which in turn uses the hardware and the system peripherals. You may also interact indirectly through an application program. The tasks you perform usually fall into one of the following three categories:

- On-line interaction with application programs
- Program development
- System management

All of the system resources previously described are at your fingertips when you sit down at a terminal and access an HP 3000 computer system. Hewlett-Packard is committed to making the time you spend at the terminal friendly and highly productive.

DOCUMENTATION

Included with the HP 3000 computer system is a comprehensive set of user manuals. Complete documentation is provided for the operating system, and for special tasks such as system installation and program conversion, in addition to documentation for individual subsystems.

Figure 1-4 shows the three functional levels of documentation provided and how the manuals relate to one another. The full set of manuals available for the HP 3000 is presented in Appendix A.

Introductory Level Manuals

These manuals introduce all the major concepts of the HP 3000 computer system and provide an introduction to the use of the system. Particularly informative are the "Using" series of manuals (Using the HP 3000, Using Files, Using HP VIEW/3000, Using COBOL) that contain practical examples showing the new user how to perform a variety of common tasks. These include how to build files, design terminal screens, compile and run programs, equate files to input/output devices, and obtain hardcopy listings.

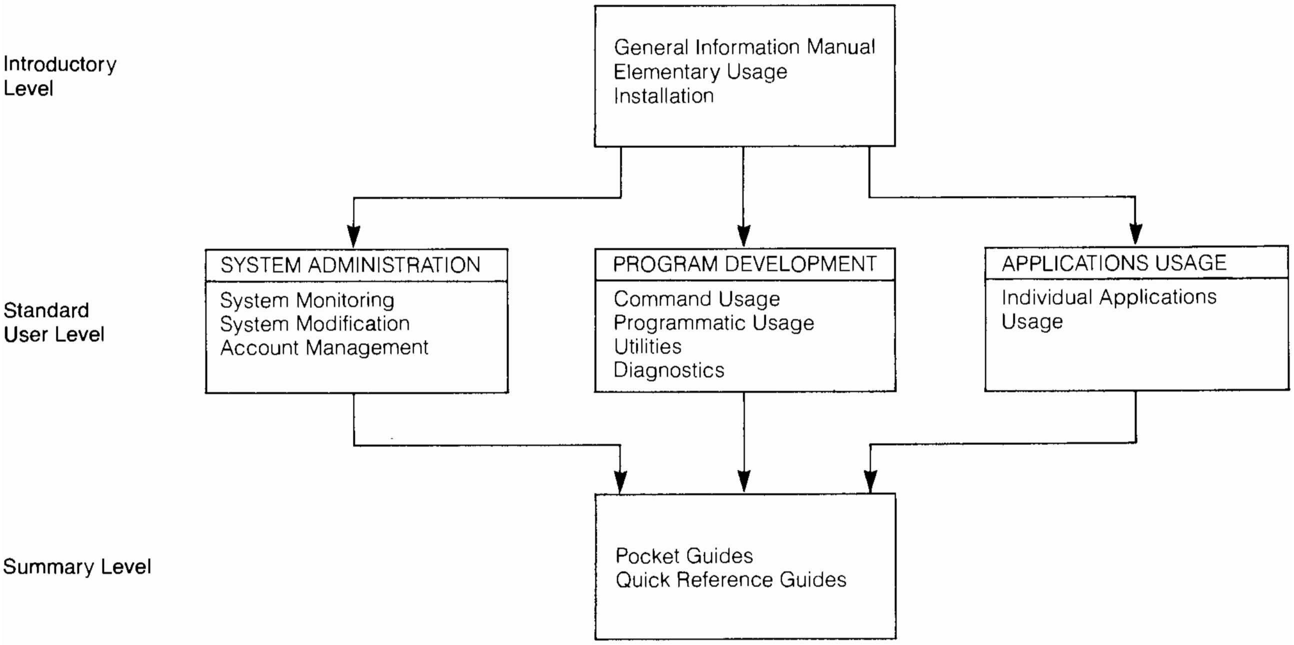


Figure 1-4. HP 3000 documentation levels

Standard User Level Manuals

These manuals provide complete reference documentation for the full line of HP 3000 software. They fall into three general categories:

- Administrative manuals—provide a guide for day-to-day monitoring of the system by a computer operator and, on a higher level, provide the more complex information required for system management.
- Program Development manuals—provide complete reference specifications for the programmer on using the operating system, the various language compilers, the data base and file management subsystems, as well as the data editing and data entry subsystems, and all the HP 3000 utility programs. An Error Messages and Recovery Manual summarizes the error messages and recovery actions for operating system, language, and utility subsystem errors. Another useful manual, the Index to the MPE Reference Documents, provides a master index to the entire set of MPE operating system manuals.
- Applications manuals—provide full instructions for non-programmers on using the Hewlett-Packard application programs available for use on an HP 3000 computer system.

Summary Level Manuals

These are a set of convenient pocket size manuals that summarize the reference specifications for the MPE operating system, and for various programming languages and subsystems. These guides are primarily memory aids and assume a familiarity with the effects of the syntax they summarize. An essential pocket guide is the Software Pocket Guide that summarizes the MPE commands and system procedures, the file system error messages, and the commands that control the more commonly used utility programs.

A quick reference guide is available for the data entry operator who uses the HP VIEW/3000 data entry application program, ENTRY.

TRAINING

Hewlett-Packard offers a variety of training courses that enable you to derive maximum benefit from the system's capabilities. A full curriculum of training courses spans the needs of a variety of users—from the system administrator, to the program developer, to the non-technical user of a software application.

Among the courses offered are:

- A Programmer's Introduction
- System Management and Operation
- IMAGE/3000—Data Base Management Training
- KSAM/3000—Keyed Sequential Access Method
- HP VIEW/3000—Screen Design and Data Entry Programming
- SPL/FILE System Introduction
- MPE III Special Capabilities
- DS/3000—Using Distributed Systems
- MFG/3000—System Administration

Courses may last from one to five days, and in most cases, can be presented at your facility or at an HP Technical Training Center. Experienced professional instructors and hands-on computer time for students combine to make the training experience an invaluable asset for your operation.

Full descriptions of the courses offered are given in Section II.

SOFTWARE SUPPORT SERVICES

A well-defined set of software support services is offered with HP 3000 computer systems. Since customer support needs can differ considerably, these support services are available to provide a flexible range of support. The software services are divided into two main categories:

- **CSS—Customer Support Service,** for customers who choose a close support relationship with Hewlett-Packard.
- **SSS—Software Subscription Service,** for customers who prefer to rely on their own resources for software support.

Documentation Distribution Services

In addition to the software support services described above, Hewlett-Packard offers two types of Documentation Distribution Services that are appropriate for customers with a large programming staff who wish to be individually informed of software problems or keep their documentation up to date. These services are:

- **SNS—Software Notification Service,** provides one copy of the Software Status Bulletin and the Communicator
- **MUS—Manual Update Service,** provides one copy of updates or new editions to manuals automatically whenever they are issued. The various sets of manuals that can receive this support are specified in the Price/Configuration Guide.

The reference sheets in Section II of this manual provide a full description of the Software Support Services outlined above.

HARDWARE SUPPORT SERVICES

A range of maintenance service tailored to your needs is available to ensure that your system runs smoothly and reliably. The services provided by the Customer Maintenance Agreement provide preventive maintenance visits and emergency repairs.

For the first 90 days after installation, an on-site warranty provides parts and labor. After the warranty period, service is continued under the Customer Support Services Agreement. The cost is determined by your configuration and the type and frequency of service best suited to your company's needs.

For more information on Hardware Support Services, refer to the reference sheet in Section II of this manual.

HP GENERAL SYSTEMS USERS GROUP

The Users Group is an independent world-wide organization for the purpose of exchanging techniques and ideas among HP 3000 users. Hewlett-Packard's Customer Relations Manager works closely with the Users Group to promote communication between Hewlett-Packard, the General System Division in particular, and HP 3000 users. Membership is open to any interested individual using an HP 3000. You will find that meeting with other users and exchanging ideas provides a stimulating environment in which to sharpen your programming and operational techniques.

Chapter 2

TRANSACTION PROCESSING

CREATING A TRANSACTION PROCESSING SYSTEM
PRODUCTION TOOLS: FUNDAMENTAL OPERATING SOFTWARE
IMAGE/3000
QUERY/3000
KSAM/3000
HP VIEW/3000
UTILITIES
PROGRAM EXECUTION FACILITY
DEVELOPMENT TOOLS: LANGUAGES
DEBUGGING AIDS
COBOL/3000
RPG/3000
FORTRAN/3000
BASIC/3000
SPL/3000
APL\3000

Today, many businesses are moving away from large batch processing computer systems toward networks of smaller computer systems running innovative on-line transaction processing applications. These installations are typified by a relatively large number of users who communicate with the computer system through terminals and who present the computer with an uneven processing load, heavy terminal and disc I/O demands, and the need to share code and data among the many users. The HP 3000 computer system is designed to meet the needs of such a transaction processing environment.

In a general sense, the expression "transaction processing" can be applied to virtually any interaction between a computer system and its users. A transaction is a series of steps in a predefined logical sequence which accepts input, performs some sort of data processing or manipulation, and generates output. The term "transaction" implies that there is an exchange taking place, and this is certainly true of an interactive transaction processing system where transactions are used to access information stored in a computer's data bases and files.

There are two major advantages of an interactive transaction processing system:

- It allows the entry, manipulation and retrieval of data at various dispersed locations—usually the locations where the data is actually generated and used for business decisions. This allows the people who are most familiar with the data to enter and interact with it.
- It speeds up the business cycle. Rapid and accurate data input and retrieval can be performed by relatively inexperienced personnel to make up-to-date information constantly available. This results in faster response to customer demands, thereby providing a major competitive edge.

CREATING A TRANSACTION PROCESSING SYSTEM

The creation of a transaction processing system involves the skillful blending of user requirements and system capabilities. First, the needs of all system users from data entry personnel, through system management and corporate staff are identified. Next the system designer or analyst translates these needs into logical specifications based on the resources and capabilities of the computer system. Finally, the programmer interprets the specification into a language the computer can understand.

In effect, four sets of requirements must be examined and met in the creation of a transaction processing system: those of the user, the system designer, the programmer, and the computer system itself.

The User:

- wants to see a conversational interface to the application program
- wants the ability to provide one-time reports without additional programming effort
- doesn't want to get involved in learning the computer system software.

The system designer:

- needs to determine the system requirements of the end user
- needs to implement three important aspects of a transaction processing system—fast-response time, rapid information access, and high transaction throughput
- should be provided with design aids to help his analysis of the application needs.

The programmer:

- needs to be familiar enough with the host computer to translate the system designer's specifications into an actual application program
- should be provided with system characterizations which define performance considerations and design tradeoffs of the computer system
- should be provided with programming tools which increase his efficiency in writing, debugging and maintaining the application program.

The computer system:

- must provide hardware and software which operate well in a transaction processing environment
- must provide all of the design aids, system characterizations, software development tools and system software which will allow the system designer and programmer to design and implement a successful transaction processing system.

Transaction Processing on The HP 3000

There are several approaches that the designer of an HP 3000 based on-line transaction processing system may use to facilitate user interaction with the system. In the most commonly used approach, the system is designed so that each user initiates a separate session and invokes an application program with a RUN command, as illustrated in Figure 2-1.

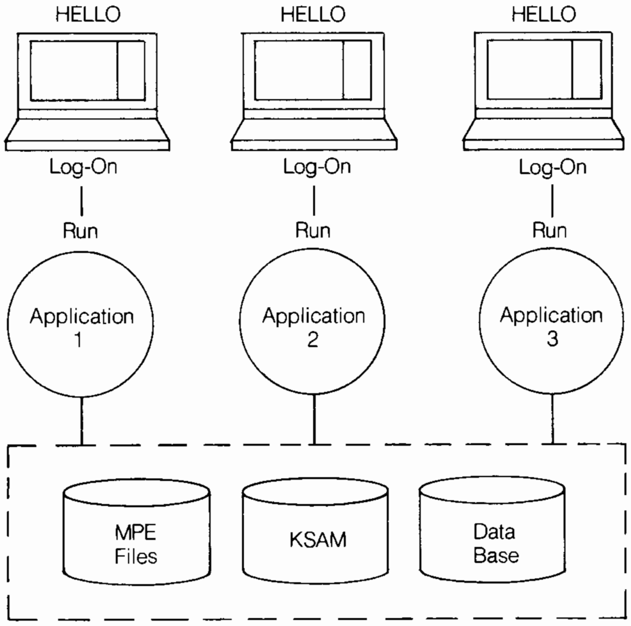


Figure 2-1. On-line transaction processing

With an alternative approach, which also runs one process per terminal, the user is completely isolated from the HP command language. As illustrated in Figure 2-2, a single process (father) manipulates multiple processes (sons), which have no connection with the MPE command language during their execution. Each son process opens a terminal and prompts the user to perform a specific transaction via a menu or screen display. The terminal user simply enters the required data, without ever having to issue a RUN command. When the user is finished with the application, the son process relinquishes control of the terminal, which can then be used in normal interactive mode.

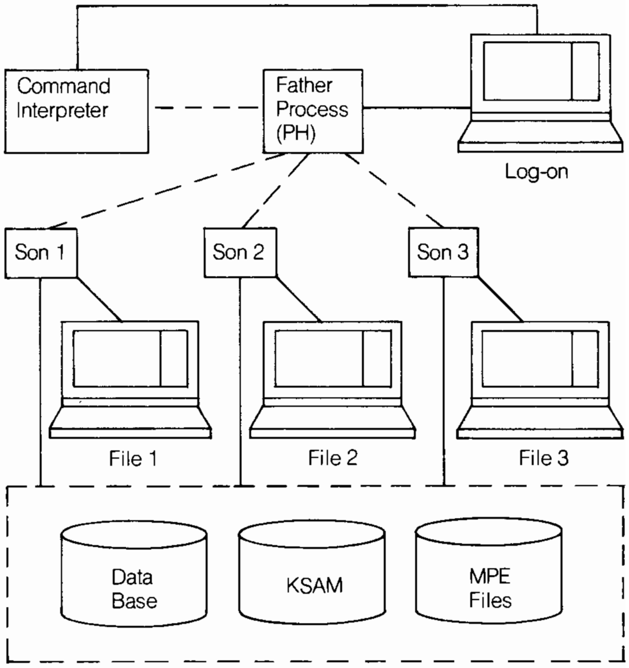


Figure 2-2. Alternative approach

PRODUCTION TOOLS: FUNDAMENTAL OPERATING SOFTWARE

A transaction processing system must achieve good system throughput and resource utilization. The design, development, and maintenance of the programs should be relatively simple and straight-forward. The HP 3000 is equipped with a complete set of tools, depicted in Figure 2-3, which meet the needs of a transaction processing system and make the implementation of that system an easy task.

Every HP 3000 is provided with Fundamental Operating Software (FOS) consisting of:

- MPE Operating System and File System
- IMAGE/3000: a comprehensive data base management system
- QUERY/3000: an English-like inquiry facility used with IMAGE
- KSAM/3000: a keyed sequential file access method
- HP VIEW/3000: a self-contained data entry system and a programmatic terminal interface
- Utilities for system administration and control
- Facility to execute compiled programs without the source language compiler

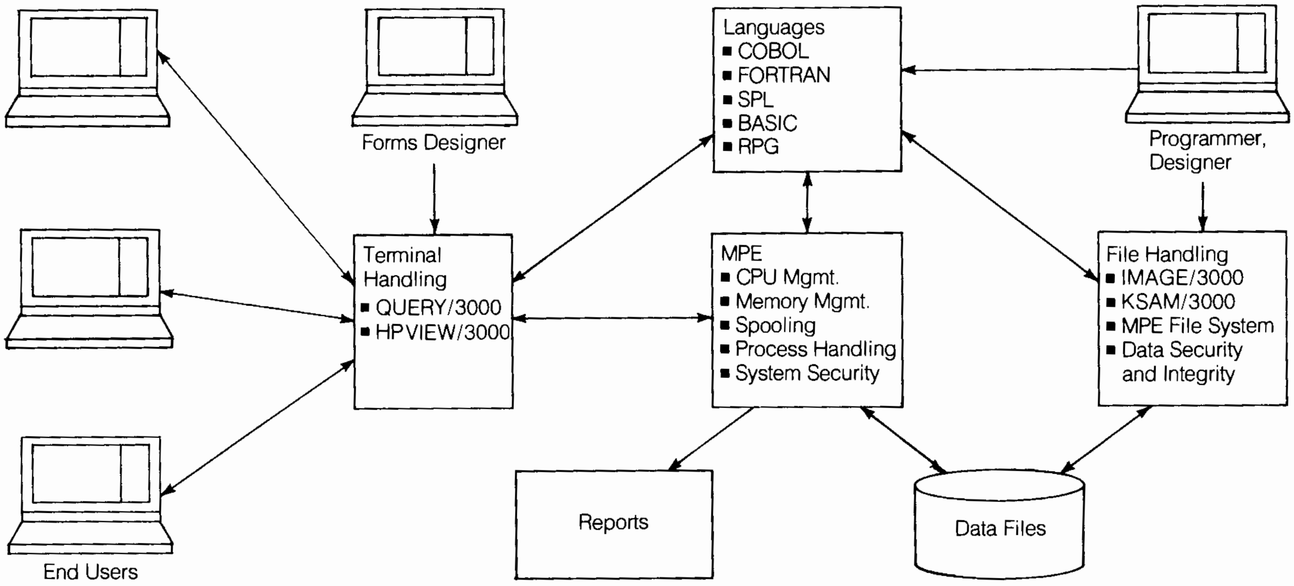


Figure 2-3. Resources for a transaction processing system on the HP 3000. Except for the Language compilers, all the above resources are included in the Fundamental Operating Software that is part of every HP 3000 computer system.

The Fundamental Operating Software equips the HP 3000 with the functions it needs to be a ready-to-run application system. FOS includes a complete set of high level procedures for data entry and data management, callable from application programs. It also provides many interactive facilities to terminal users for screen design and data access.

With the addition of a language compiler, FOS becomes a powerful program development system. Programs developed and compiled on one HP 3000 execute on any other HP 3000 without installing any software other than the program itself.

In addition, HP 3000 transaction processing systems can make use of the Hewlett-Packard Distributed System Network (HP-DSN), a powerful tool for collecting, reporting, and sharing data among physically remote locations, as described in Chapter 4.

IMAGE/3000

IMAGE/3000 is a powerful data base management system. Included in the Fundamental Operating Software, it handles multiple files and helps you define and create a data base tailored to your special requirements. IMAGE is especially designed for data handling situations where a large number of files are needed and the inter-relationships of data within the files are complex.

A data base is a collection of logically-related files containing both data and structural information. Pointers within the data base allow you to access related data and to index data across files.

The primary benefit of the IMAGE data base management system is the time savings which result from the file consolidation and the ease with which programs are developed and maintained.

Most information processing systems that serve more than one application area contain duplicate data. The consolidation of multiple files into a single data base eliminates most of the data redundancy. Through the use of pointers, logically related items of information are chained together, even if they are physically separated, allowing the data to be used by any program needing it. Since there is only one record to retrieve and modify, the work required for data maintenance is greatly reduced, and all reports using those items of information are consistent.

IMAGE allows the data base to be structured independently of the application programs that use it, so that programs only need to define those data items actually used, rather than the entire file. As a result, changes can be made to other parts of the data base without modifying the program that accesses the data base. Coding can begin before the data base structure is finalized, as long as the format of the individual data items accessed by the program is known. Future enhancements to the data base can be implemented without impacting existing programs. The logical separation of the structure of the data base from the applications results in greater flexibility and substantial time savings for both the data base designer and the programming staff.

Data Base Structure

IMAGE is primarily a path oriented or chained approach to data retrieval. Pointers are maintained which logically connect those records with common attributes into chained lists. This allows cross-referenced access to collections of data down to the smallest unit and makes it possible to access related data quickly.

An IMAGE data base consists of data items, data entries, and data sets. A data item (field) is the smallest accessible data element. It consists of a value referenced by a data item name. In general, many data item values are referenced by the same data item name.

A data entry (record) is an ordered set of related data items. The order of the data items within a particular data entry is specified when designing the data base. The length of a data entry is the combined lengths of all data items within it.

A data set (file) is a collection of data entries. Each data set is referenced by a unique data set name.

A data base is a named collection of related data sets. It is defined in terms of data items and data sets. A data base may be defined with up to 255 data item names and 99 data set names.

An IMAGE data set is either a master or a detail data set.

Master data sets:

- are used to keep information relating to a uniquely identifiable entity; for example, information relating to a customer (such as account number, name, address, and credit rating)
- allow for rapid retrieval of a data entry since one of the data items in the entry, called the search item, determines the location of the data entry
- can be related to detail data set containing similar search items and thus serve as indexes to the detail data set.

Detail data sets:

- are used to record information about related events; for example, information relating to each sale for each customer in the master data set
- allow retrieval of all entries pertaining to a uniquely identifiable entity; for example, an account number may be used to retrieve all data entries containing information about sales to that customer.

As illustrated in Figure 2-4, each master data set may serve as an index to a maximum of 16 detail data sets, and each detail data set may be indexed by up to 16 master data sets (or none).

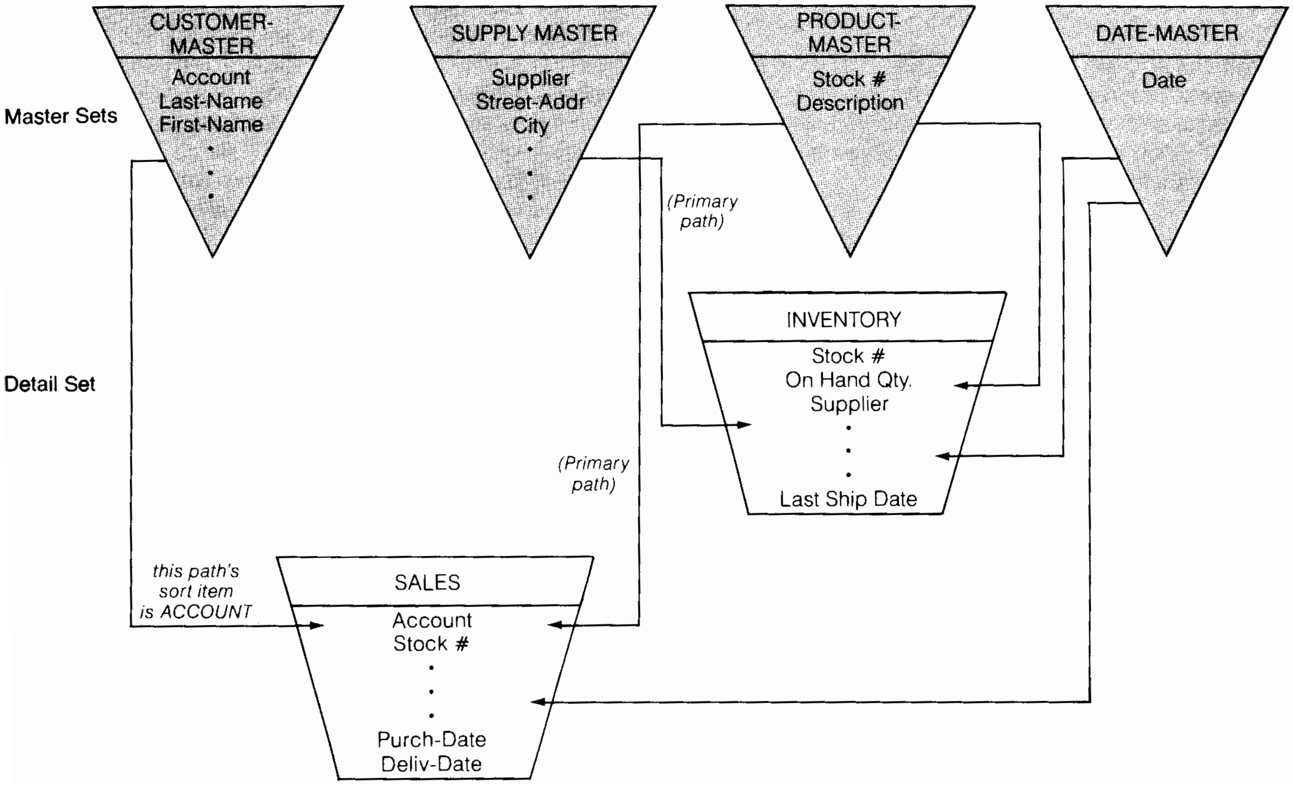


Figure 2-4. Example of master-detail data set relationships.

Accessing the Data Base

You access a data base through call statements to IMAGE Data Base Management Subsystem (DBMS) procedures in COBOL, FORTRAN, BASIC, or SPL programs, or through compiler generated calls in RPG programs. These options give you the ability to select the language most appropriate to each application. The IMAGE procedures locate data, maintain pointer information, manage the allocated file space, and return status information to you. The tasks performed by the IMAGE procedures relieve you of the "bookkeeping" normally associated with file management and allow you to concentrate on applying the processing power of whatever programming language you are using.

Some of the functions performed by IMAGE procedures are:

- Adding a new entry to a data set.
- Deleting an entry from a data set.
- Reading some or all of the data items of an entry.
- Changing the values of data items of an entry.

There are four modes of access available:

Serial—in this mode IMAGE starts at the most recently accessed record and sequentially examines successive records without regard to their key value. Forward and backward serial access is available.

Directed—in this mode the calling program specifies the record address of the data entry from which the desired data items are to be retrieved.

Calculated—this type of access involves retrieval of master data set entries based upon the search item value.

Chained—this type of access consists of successive retrieval of all entries in a detail data set which contain the same search item value.

With Distributed Systems/3000 (DS/3000) software, you may also access a data base on a remote HP 3000. The same call statements used to access a local IMAGE data base can be used with DS/3000 to access remotely located data bases across a communication line. Both the local and remote HP 3000's must have DS/3000 software.

Data Base Security

In addition to the security safeguards of MPE, IMAGE provides the data base administrator with further protection for the data base. The key to the strength of IMAGE/3000 security is the ability to control access not only to specific data sets (files), but also to each data item (field), as shown in Figure 2-5. You can define up to 63 user classes, each with a related password, and associate each class with either read or write access to specific data sets and items.

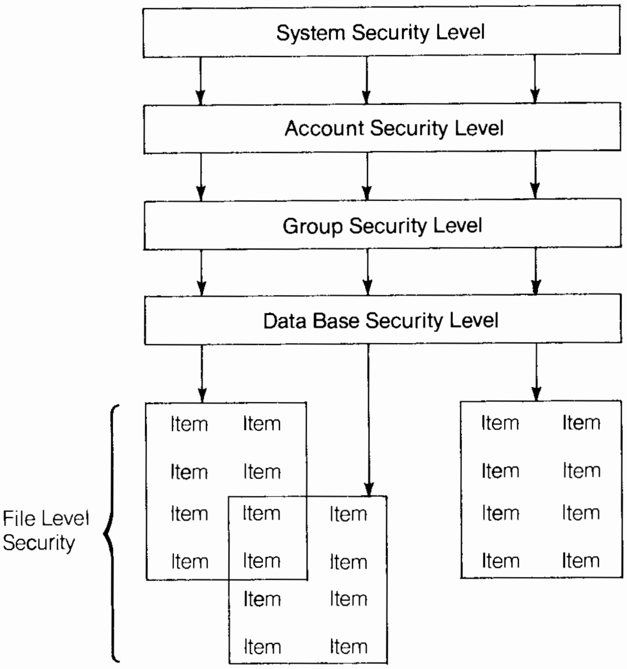


Figure 2-5. Total system and data base security

Therefore, the data base is protected down to the smallest unit of information, for example, a data item specifying an employee's pay rate. This ensures that the only users who access and/or change specific items are those whose job functions require it.

Because multiple users can be simultaneously accessing and updating an IMAGE data base, data integrity must be assured. You may specify the type of access you need—e.g., read, modify, update, etc., directly from the terminal or through a program. IMAGE will allow that access unless it conflicts with the functions already being performed by the other concurrent users. This approach results in compatible sharing of the data base and ensures data integrity.

Transaction Logging and Recovery

IMAGE/3000 offers a logging and recovery system which is based on MPE III user logging and is designed to restore data bases to a consistent state, both logically and structurally.

The IMAGE/3000 logging facility provides the means to log data base transactions to a log file on tape or serial disc. The log file is a record of all modifications to data base items, including information about previous entries as well as the current state of the data base. User text can be logged in order to facilitate future access and interpretation of the log files; it may also be a useful tool for auditing. The recovery system reads the log file to re-execute transactions against a data base back-up copy in the event of a system failure.

The data base administrator is responsible for enabling or disabling the logging and recovery processes, making logging a global function controlled at the data base level rather than at the individual user level.

Query/3000

QUERY is a companion data base inquiry facility for IMAGE and is also part of the Fundamental Operating System. A self-contained language, QUERY provides a simple means of accessing the data base through the use of English language key words and other character strings. You communicate with QUERY through 17 unique commands to store, modify, retrieve, and report on data in an IMAGE data base. Commands can be entered either from an interactive terminal or a batch input device such as a card reader.

With Distributed Systems/3000 (DS/3000) software, the same QUERY commands used to access a local data base may be used for direct access of remote HP 3000 data bases, across a communications link.

Applications

Since QUERY is designed for the non-programmer, it can be employed in a variety of applications after only minimal training. Both novices and experienced programmers find it extremely valuable. Some of the major application areas include:

- Casual Inquiry of the Data Base—facilitates searching a data base for information without writing a program or waiting for a periodic batch run.
- On-line Data Updates—permit modification or deletion of data on-line, directly from a terminal to the IMAGE data sets (files).
- Report Generation—formats reports with header and column labels, page numbers, group labels and summary statistics. Pre-defined report formats can be catalogued in the system to aid inexperienced users.
- Application Program Debugging—aids in program development. QUERY can be used to build test data as well as to interrogate the results of program and system tests. This feature eliminates the requirement that file-related programs be completed before meaningful functional programs can be written.

Features

In addition to the applications listed, QUERY performs several other valuable functions.

- Selection of Data Through Logical Comparisons—locates specific entries for processing based on logical criteria specified in a FIND command.
- Creation and Storage of Procedures—stores frequently used or lengthy commands as individual procedures in a command file, avoiding the necessity of retyping them when needed.
- Display Data Base Structure—displays structural information about the data base and shows the relation between data items and the data sets in which they are located.

Security Provisions

QUERY adheres to all of the security provisions included in the IMAGE data base. After QUERY is invoked, a security password must be entered which determines which data entries and data items may be accessed.

Data Types

QUERY manipulates many of the various IMAGE data types. Each specific data type has a length ranging from two digits plus a sign to 132 characters. The types of data are:

- One-word integer numbers
- Two-word integer numbers
- Two-word real numbers
- Four-word real numbers
- One-word logical values as absolute numbers
- Zoned decimal numbers
- Packed decimal numbers
- ASCII character strings
- ASCII character strings containing no lowercase alphabetics
- One-word integer numbers corresponding to COBOL computational data
- Two-word integer numbers corresponding to COBOL computational data.

Inquiries

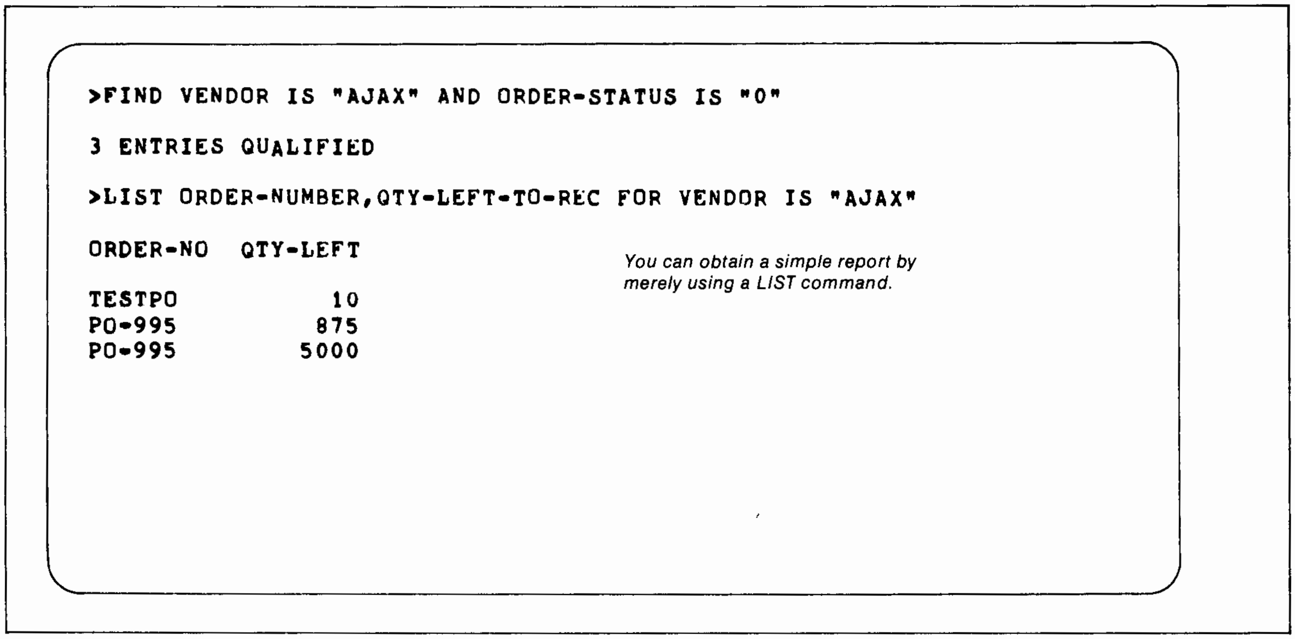
The FIND command locates entries in a data base. Logical selection criteria, included in the command, allow only the entries pertaining to a given inquiry to be returned. Up to 50 logical relationships can be specified in one command.

Reporting

Data entries can be displayed or reported according to a user-specified format. After the data records have been located by the FIND command, a REPORT command is used to specify which items within those records QUERY is to display. REPORT also specifies such items as titles, column headings, page numbers, and line spacing. Figure 2-6 illustrates a QUERY report.

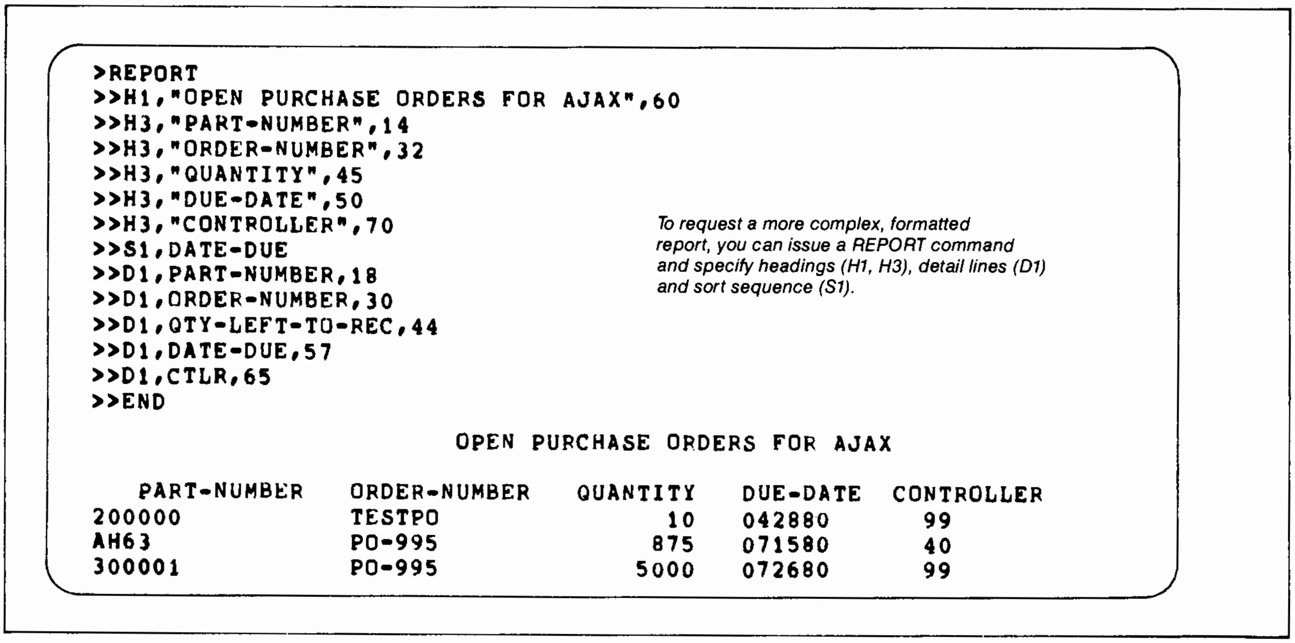
Updating

A data base may be updated by adding, deleting, or modifying an entry using the UPDATE command which operates within the limits established for a password. QUERY prompts for the values needed to complete a transaction.



```
>FIND VENDOR IS "AJAX" AND ORDER-STATUS IS "O"

3 ENTRIES QUALIFIED

>LIST ORDER-NUMBER,QTY-LEFT-TO-REC FOR VENDOR IS "AJAX"

ORDER-NO  QTY-LEFT

TESTPO          10
PO-995         875
PO-995        5000
```

You can obtain a simple report by merely using a LIST command.



```
>REPORT
>>H1,"OPEN PURCHASE ORDERS FOR AJAX",60
>>H3,"PART-NUMBER",14
>>H3,"ORDER-NUMBER",32
>>H3,"QUANTITY",45
>>H3,"DUE-DATE",50
>>H3,"CONTROLLER",70
>>S1,DATE-DUE
>>D1,PART-NUMBER,18
>>D1,ORDER-NUMBER,30
>>D1,QTY-LEFT-TO-REC,44
>>D1,DATE-DUE,57
>>D1,CTLR,65
>>END
                         OPEN PURCHASE ORDERS FOR AJAX

   PART-NUMBER    ORDER-NUMBER   QUANTITY   DUE-DATE  CONTROLLER
200000            TESTPO               10   042880       99
AH63              PO-995              875   071580       40
300001            PO-995             5000   072680       99
```

To request a more complex, formatted report, you can issue a REPORT command and specify headings (H1, H3), detail lines (D1) and sort sequence (S1).

Figure 2-6. Typical QUERY report procedure

KSAM/3000

KSAM/3000 (Keyed Sequential Access Method) is a part of Fundamental Operating Software that provides sophisticated file access by key values within the data record. Each data record contains one primary key field and may include up to 15 alternative ones. Access to these records can be sequential or random by either primary or alternative key value. KSAM also supports key access by physical or logical record number, or by chronological order.

You may use KSAM to retrieve data with part of a key value rather than the entire key. Called partial or generic key search, this approach is ideal for values that share a common beginning. Suppose you wish to find all the customers in a certain geographic area. By specifying only the common first three characters of the zip code—like 950, it is possible to read quickly 95050, 95060, 95065, etc. Records in a KSAM file can also be located through approximate searches. For instance, you can retrieve the first record whose key value is greater than or equal to "01/01/79".

If you have existing indexed sequential files, KSAM/3000 can facilitate their conversion to the HP 3000. For example, RPG programs on an IBM SYSTEM/3 using indexed access method require no coding changes to run on the HP 3000.

File Structure

A KSAM file consists of two physical disc files—one for data and one for keys. The data file contains all the data records while the associated key file holds one or more sets of entries that maintain the primary and alternate logical sequences of the data records. When constructing a file, key entries are dynamically added in ascending order. Any key file restructuring required to accommodate additional data file records is performed automatically by KSAM in an operation that is invisible to you. Figure 2-7 illustrates a KSAM file with a primary key and one alternative key.

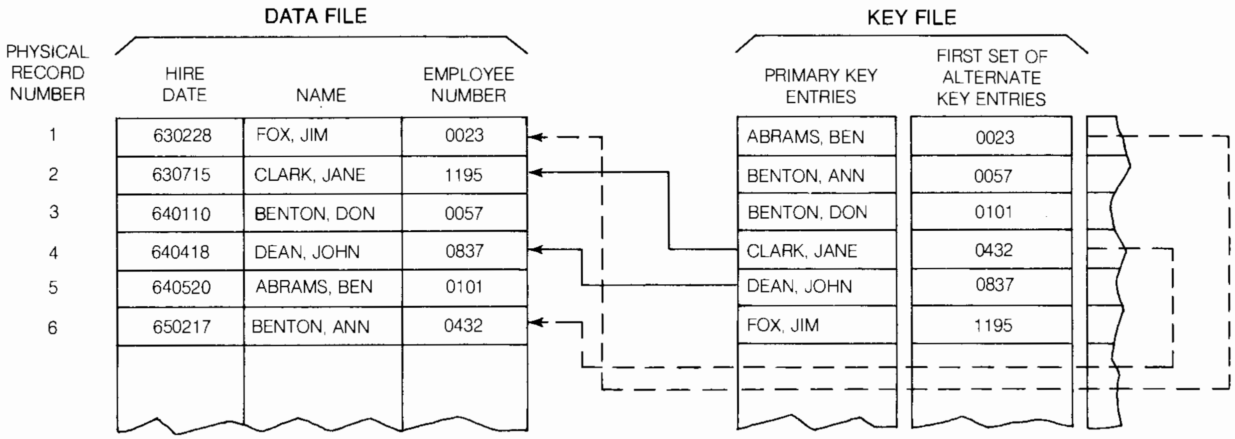
Data fields, including key fields, may contain these types of data:

- signed binary integer,
- double-word signed binary integer,
- real,
- four-word long real,
- ASCII character string,
- packed decimal,
- zoned decimal.

Features

KSAM/3000 provides a number of features, including:

- Multiple Keys—One primary key and up to 15 alternate keys may be specified for any KSAM data file. Each key is ordered in sequence by its value with no relation to other keys.
- Duplicate Key Values—While KSAM normally expects unique values for keys, some key types (such as zip codes) may logically contain duplicate values. You may optionally define these keys to allow non-unique data.



DATA FILE

PHYSICAL RECORD NUMBER	HIRE DATE	NAME	EMPLOYEE NUMBER
1	630228	FOX, JIM	0023
2	630715	CLARK, JANE	1195
3	640110	BENTON, DON	0057
4	640418	DEAN, JOHN	0837
5	640520	ABRAMS, BEN	0101
6	650217	BENTON, ANN	0432

KEY FILE

PRIMARY KEY ENTRIES	FIRST SET OF ALTERNATE KEY ENTRIES
ABRAMS, BEN	0023
BENTON, ANN	0057
BENTON, DON	0101
CLARK, JANE	0432
DEAN, JOHN	0837
FOX, JIM	1195

CHRONOLOGICAL ORDER: 1, 2, 3, 4, 5, 6

PRIMARY KEY (Last Name) ORDER: 5, 6, 3, 2, 4, 1

FIRST ALTERNATE KEY (Employee Number) ORDER: 1, 3, 5, 6, 4, 2

Figure 2-7. KSAM file example.

- Retrieval by Generic Key—You may elect to retrieve records based on a match with the first part of the key only. For example, all stock numbers for similar parts can be read if they share a common prefix.
- Retrieval by Approximate Match—Records may be retrieved based on their relation to a specified key value such as all dates greater than 770401.
- Fixed or Variable Length Data Records—For added flexibility, both fixed and variable length record formats are allowed for KSAM files.

These features make KSAM an appropriate tool for applications where the variety of file access options is more important than flexibility in data base design.

HP VIEW/3000

HP VIEW/3000 is a comprehensive software subsystem of the Fundamental Operating Software that implements and controls source data entry, and also provides an interface between terminals or files, and any transaction processing application.

Source Data Entry

As a source data entry system, HP VIEW/3000 provides easy forms design with data editing and validation built into the forms. It also provides a ready-to-use data entry program so that you can enter data into the system with no programming effort. This same program allows you to "browse" the entered data and modify it as it is entered.

Thus source data entry through HP VIEW/3000 can be done without the use of a compiler. If, however, you need additional or different capabilities, you can use the HP VIEW/3000 procedures to modify the existing data entry program or write your own application.

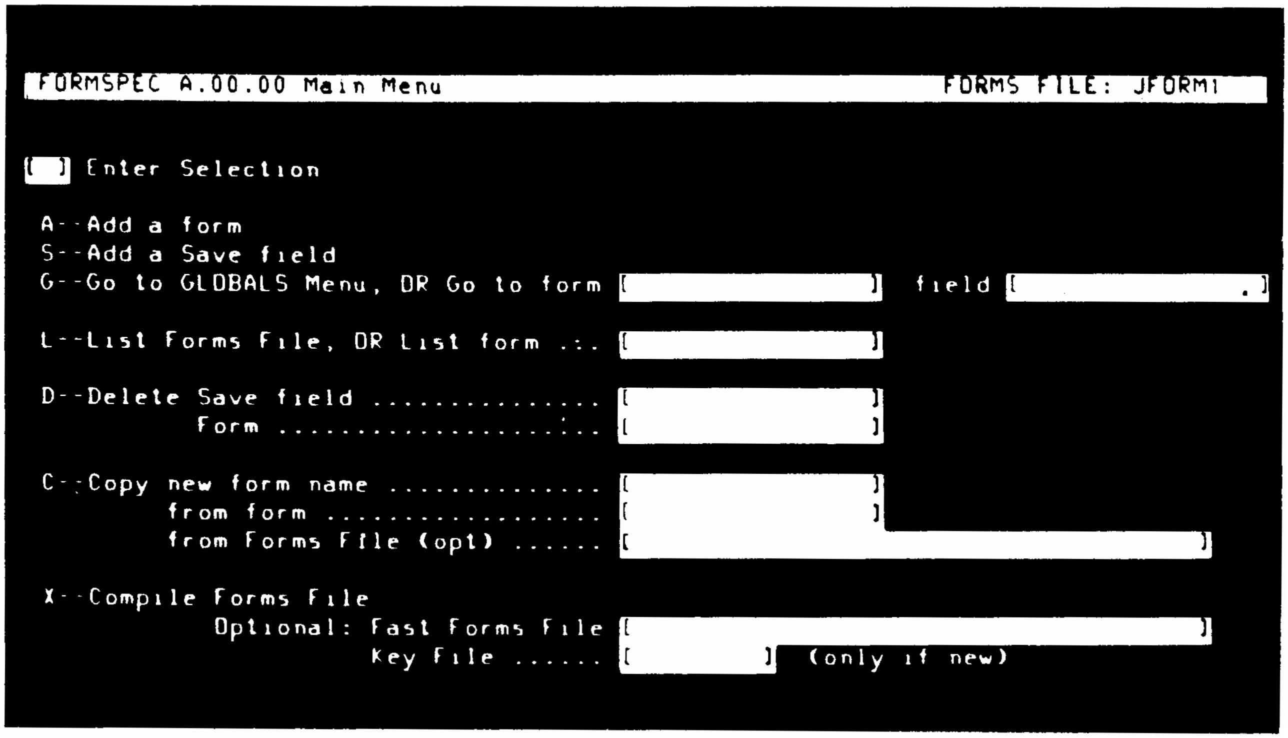
Programmatic Interface

As an interface to transaction processing applications, HP VIEW/3000 provides a set of procedures that allow you to control terminals, forms, and files from an application program. These procedures are available to programs in COBOL, RPG, FORTRAN, BASIC, and SPL.

HP VIEW/3000 also provides a reformatting capability. You can enter specifications to control how data in a batch file is to be reformatted, and then run a program to actually reformat the data.

The HP VIEW/3000 procedures and the reformatting capability, either singly or in combination, provide a "front end" to existing transaction processing applications. Thus, HP VIEW/3000 allows you to concentrate on processing problems rather than on data editing or interface to terminals.

A combination of menu type screens and terminal function keys assist you in manipulating forms design. For example, with the FORMSPEC facility you can design forms, change existing forms, add new forms, and delete forms or fields as shown in Figure 2-8.



```
 FORMSPEC A.00.00 Main Menu                                     FORMS FILE: JFORM1

[ ] Enter Selection

 A--Add a form
 S--Add a Save field
 G--Go to GLOBALS Menu, OR Go to form [               ]  field [              .]

 L--List Forms File, OR List form .:. [               ]

 D--Delete Save field ............... [               ]
           Form ..................... [               ]

 C-:Copy new form name .............. [               ]
        from form ................... [               ]
        from Forms File (opt) ....... [                                  ]

 X--Compile Forms File
          Optional: Fast Forms File [                                  ]
                     Key File ...... [        ] (only if new)
```

Figure 2-8. Main menu screen FORMSPEC facility.

Features

The following are the main features of the HP VIEW/3000 system:

- A forms design program (FORMSPEC)—allows quick and easy forms design at a terminal using "menus."
- Advanced forms design (through FORMSPEC)—provides data editing, formatting, and validation of data as it is entered.
- A data entry program (ENTRY)—provides immediate data entry and modification with no programming effort.
- A flexible data reformatting design program (REFSPEC) specifies reformatting of data to suit an existing application.
- A batch program (REFORMAT)—reformats the data according to the REFSPEC formatting specifications and writes it to a file for use by an application.
- A set of procedures that provide a powerful language interface to terminals, forms, and files from user programs written in COBOL, RPG, BASIC, FORTRAN, or SPL.

Designing Forms

The FORMSPEC program allows you to design forms in as simple or as complex a manner as you desire. Form design with FORMSPEC can be thought of as having four levels of complexity:

- ASCII Collection—You draw the form on the terminal screen and accept any ASCII data entered by the operator.
- Simple Editing—You draw the form on the screen and specify edits based on field type (optional, required, display only) or data type (character, numeric, or date). No special language is required for these edits.
- Full Field Edits—You specify a full range of field edit statements that apply to individual fields in a form. These include: minimum length, range checks, and pattern checks. A subset of the FORMSPEC field specification language is used for these edits.
- Advanced Processing—You specify movement of data between fields and forms, formatting of data (JUSTIFY, FILL, STRIP), alteration of forms sequence, and conditional processing based on the result of editing statements. This level uses the full range of the FORMSPEC language.

Once designed, forms are stored in a forms file and compiled for use in data entry. Forms can be modified during initial design, or even after the forms file is compiled.

Data Entry Program

A stand-alone data entry program, ENTRY, displays forms at the terminal, accepts data entered on the forms, and writes the data to a batch file.

ENTRY operates in two modes: Data Collection and Browse/Modify. The mode when ENTRY is first run is always data collection; the operator must request browse/modify by pressing a terminal function key. As indicated by their names, data collection mode is used to collect data from the terminal, and browse/modify to look at the collected data and modify it if necessary.

In Data Collection mode, a set of terminal function keys allows the operator to:

- Request the first (or head) form.
- Terminate a repeating form and display the next different form.
- Clear the current form to its initial values.
- Print the current form with its initial values on the line printer.
- Request browse/modify mode to view and/or change data already written to the batch file.
- Terminate the ENTRY program.

In Browse/Modify mode, a set of terminal function keys allows the operator to:

- Display data from the first batch record on its form.
- Display data from the previous batch record on its form.
- Display data from the next batch record on its form.
- Clear the current form to the values displayed before any current modifications were entered.
- Delete the data currently displayed at the terminal (in effect, this deletes the current batch record).
- Print the current record, before modification, on the line printer.
- Continue entering data after interrupting data collection.
- Terminate operation of ENTRY.

Reformatting Data

Data entered through HP VIEW/3000 forms is usually written to a batch file. This file can then be used as input to an application program. Sometimes it is necessary to reformat the data in the batch file so that it meets the input requirements of the application program. HP VIEW/3000 meets this need with a reformatting capability that allows you to:

- Combine data entered on several forms into one output record.
- Separate data entered on a single form into several output records.
- Rearrange data within a record, inserting constants, and generating check digits before writing it to the output file.
- Format data within fields by justifying, filling, stripping characters, or adjusting the sign of a numeric value.

Program Interface

HP VIEW/3000 provides a library of high-level procedures for use by all terminal-oriented applications. These procedures can be called by RPG, COBOL BASIC, FORTRAN, or SPL programs. The program interface provides:

- Terminal Interface—Procedures to open and close a terminal file, to display a form at the terminal, and read data entered in fields of the displayed form.
- Forms File Interface—Procedures to open and close a forms file, to get the next form in sequence from the forms file, and to print the current form, with its contents, on the line printer.
- Data Manipulation—Procedures to initialize a form to its initial values, to perform any FORMSPEC or user-defined edits, and to perform any final form processing as defined by FORMSPEC.
- Data Entry—Procedures that open and close a batch file, write data to a batch file, or read data from the batch file.
- Access to Data—Procedures to read entered data to a program buffer, or write data from a program buffer; data from an entire form can be read or only from selected fields; similarly, data can be written for an entire form or only for selected fields. Data passed to or from fields can be converted to or from a variety of data types.
- Status/Error Control and Custom Error Messages—Procedures to set error flags, display custom error messages, and enhance fields in error.

Utilities

Program development on the HP 3000 is made easy through a powerful editor and other commonly-used utility routines.

EDIT/3000

EDIT/3000, the HP 3000 text editor, allows you to create and manipulate files and upper- and lower-case ASCII characters with great ease. Lines, strings, and individual characters can be located, inserted, deleted, and replaced. The files to be edited may contain source language programs or text material such as reports.

You interact with EDIT/3000 through a set of editor commands that includes commands common to many other text editors throughout the computer industry. As illustrated in Figures 2-9 and 2-10, experienced users can write programs directly on the terminal, bypassing both coding sheets and punched cards.

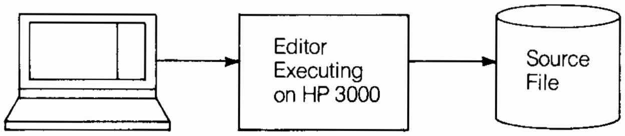
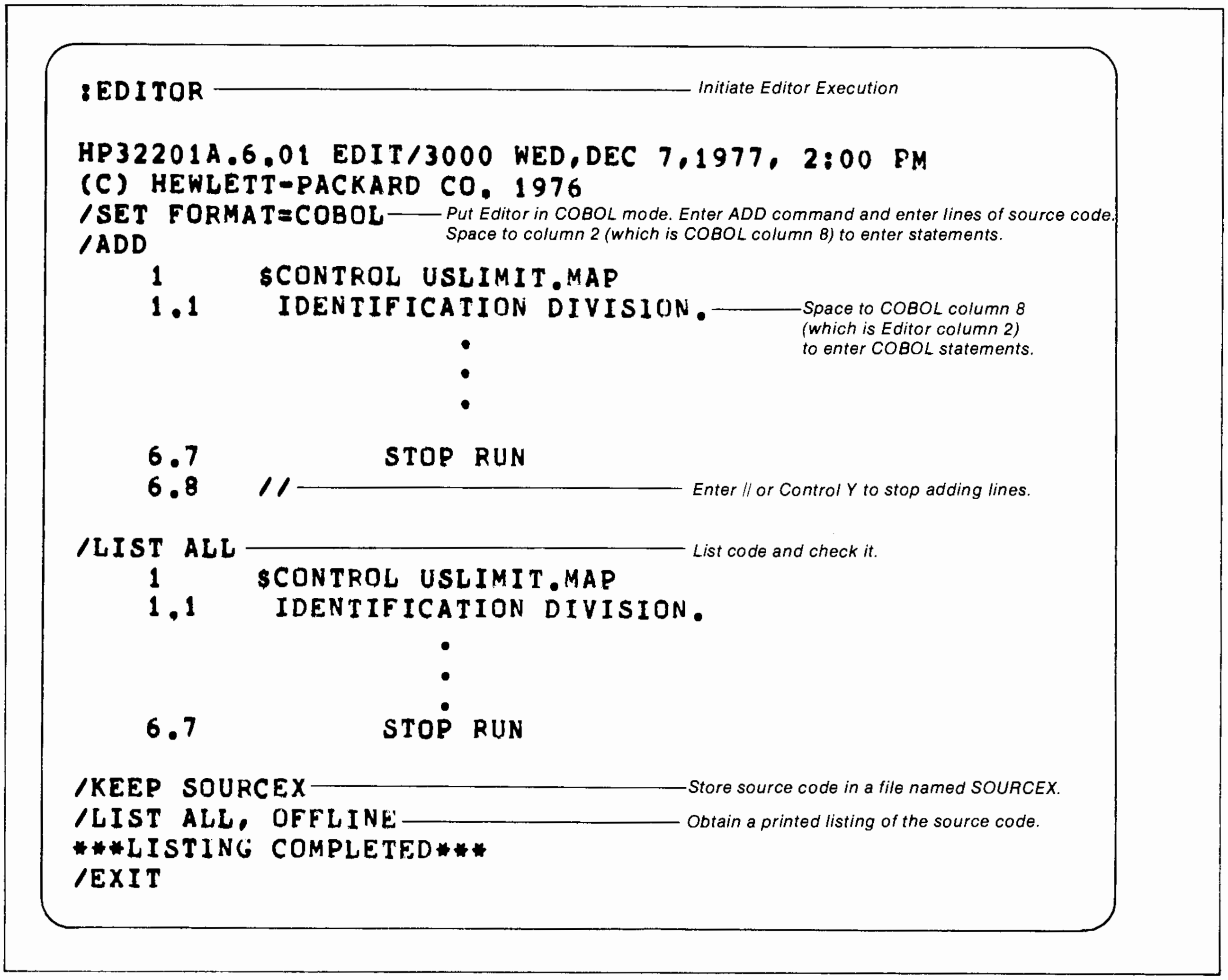


Figure 2-9. Creating a source file at your terminal

After initiating a session with the :HELLO command, you can use the Editor to create a source file. Figure 2-10 is an example of the simplest method:



```
:EDITOR

HP32201A.6.01 EDIT/3000 WED,DEC 7,1977, 2:00 PM
(C) HEWLETT-PACKARD CO. 1976
/SET FORMAT=COBOL
/ADD
    1      $CONTROL USLIMIT,MAP
    1.1     IDENTIFICATION DIVISION.
                  .
                  .
                  .
    6.7            STOP RUN
    6.8    //

/LIST ALL
    1      $CONTROL USLIMIT,MAP
    1.1     IDENTIFICATION DIVISION.
                  .
                  .
                  .
    6.7            STOP RUN

/KEEP SOURCEX
/LIST ALL, OFFLINE
***LISTING COMPLETED***
/EXIT
```

Initiate Editor Execution

Put Editor in COBOL mode. Enter ADD command and enter lines of source code. Space to column 2 (which is COBOL column 8) to enter statements.

Space to COBOL column 8 (which is Editor column 2) to enter COBOL statements.

Enter // or Control Y to stop adding lines.

List code and check it.

Store source code in a file named SOURCEX.

Obtain a printed listing of the source code.

Figure 2-10. Using the HP 3000 text editor

Basic editing functions such as adding, locating, changing, deleting and listing lines are provided using single-word commands (which can even be abbreviated to a single, unique letter), whether the edits are applied to one line or many lines. Most functions operate either on specified line numbers or on lines containing specified character strings. Groups of lines may be moved or copied within a file or from one file to another.

EDIT/3000 lets you write complex edit command sequences where editing is based upon conditions found within the text itself. In such command sequences the editor language assumes an ALGOL-like structure with the commands WHILE, NOT, and OR acting upon statements that can be simple or compound in nature.

Through the use of the Z::= and USE commands, you can store sequences of editor commands in the system and then invoke them at will to act upon text in the editor work file. In this manner, commonly-used sequences of editor commands, such as complex WHILE sequences, can be activated with a single USE command rather than by reentering the entire sequence.

A very powerful feature of EDIT/3000 is that it allows you to write procedures in COBOL, FORTRAN, or SPL that perform editing functions not otherwise possible with the standard repertoire of editor commands. You write, prepare, and store your procedures on disc and then invoke them with the PROCEDURE editor command. The editor passes the lines of text specified in the PROCEDURE command, one at a time, from its work file to the designated procedure, which then manipulates them and passes them back to the editor. EDIT/3000 substitutes the modified lines for the original ones in the work file. After the procedure is finished, the editor resumes processing standard editor commands. This feature allows you to perform any editing function that can be done a line at a time.

SORT-MERGE/3000

SORT-MERGE/3000, the HP 3000 sort/merge program, allows you to sort records in a file into a prescribed order and to merge records by combining two or more previously sorted files into one sorted file. SORT-MERGE can be used as a free-standing subsystem through a few simple commands in either an interactive session or a batch job. It can also be accessed by programs written in COBOL, FORTRAN, or SPL.

FCOPY/3000

FCOPY, the HP-3000 file copier, may be used for all file copying operations on the HP 3000, including KSAM files. With FCOPY you can transfer files from disc, magnetic tape, or any other valid input device to disc, magnetic tape, or any other valid output device.

Facility to Execute Compiled Programs Without the Source Language Compiler

Included as part of the Fundamental Operating Software of each HP 3000 system is the facility to execute compiled programs without the source language compiler (except for those written in APL\3000). With this facility, programs compiled on one HP 3000 may be run on other HP 3000 computer systems without requiring any additional software other than the program itself. Thus one compiler on one system can supply the program development needs of an entire network.

Many other utilities are included in the Fundamental Operating Software to help administer the HP 3000 resources and assist in program development.

For a more detailed explanation of all HP 3000 utilities, turn to Section II of this manual, where each product is described in a product reference sheet.

DEVELOPMENT TOOLS: LANGUAGES

In addition to the components of the Fundamental Operating Software, six high-level programming languages are available for applications program development. The HP 3000 provides highly compatible interfaces between these languages and the subsystems of the Fundamental Operating Software. Thus, the data entry, data management, and utility subsystems can be combined with one or more of the language compilers to provide a highly sophisticated program development tool.

The six programming languages available with the HP 3000 are:

- COBOL/3000
- RPG/3000—Compatible with the industry standard RPG
- FORTRAN/3000—ANS FORTRAN X3.9-1868 with extended features
- BASIC/3000—Compatible with industry standard with extended capabilities
- SPL/3000—A combination high-level and machine-dependent language especially developed for programming the HP 3000
- APL\3000—Industry standard with significant extensions (available only on the HP 3000 Series III)

Once a program written in any language (except APL\3000) is developed and compiled, it can run on any HP 3000 system equipped only with the Fundamental Operating Software.

Program development on HP 3000 computer systems is interactive and can be done concurrently with production data processing.

HP 3000 languages may be used concurrently in any combination by many users or, for multiple compilations, several languages may be incorporated within a single job or session.

All of the compilers on an HP 3000 are invoked in the same general way. This reduces the amount of training required, since learning to use one compiler is essentially learning to use them all. The HP 3000 also preserves your programming investment by incorporating subroutine call conventions in the machine's microcode. This enables programs in all languages, except APL, to call subroutines written in other languages.

A program written in any of the available programming languages (except APL) may also manipulate an IMAGE data base, access a KSAM file, or use HP VIEW/3000 forms—making the HP 3000 data base management and data entry capabilities accessible wherever they are needed.

All programs which run on an HP 3000 access data through the MPE file system. Thus, the programmer need not be concerned with the physical type of device on which the data resides. No changes need to be made to a program to switch input or output from one device to another, for example, from cards to an interactive terminal. Programs used in batch jobs can also be used without modification in interactive sessions, and vice versa. In addition, files created by one language can be accessed by any other language. See Chapter 5 for more details on MPE and the file system.

DEBUGGING AIDS

The HP 3000 provides a wide variety of aids for debugging programs at many different levels. MPE commands such as SHOWJOB, LISTF, or SHOWOUT are available to inquiry about file status of a job or a file. Compiler subsystem $CONTROL commands may be used to specify whether to print source code, object code, warning messages, or a symbol map. The #IF compiler subsystem command permits conditional compilation of portions of the program, allowing debugging statements to be inserted permanently in the code and compiled only when needed.

Two powerful capabilities available during execution of a program are Debug and Stack Dump. The Debug facility is an intrinsic procedure which provides an interactive debugging facility to enable you to set breakpoints, display the contents of memory locations, modify memory locations containing data, display and/or modify the contents of registers, and so forth.

The Stack Dump facility is composed of an intrinsic that enables a program to selectively dump any part of the stack and a stack analysis mechanism that is activated when a program aborts. If the program is running interactively, an automatic call to the debug procedure is generated. If the program is running in a batch environment, part or all of the stack is dumped.

COBOL/3000

HP 3000 computer systems can be supplied with the ANSI 68 standard COBOL. The American National Standard COBOL X3.23-1968, as approved by the American National Standards Institute, has eight processing modules. Of these, COBOL/3000 provides a full Level Two implementation of:

- Nucleus
- Table Handling
- Sequential Access
- Random Access
- SORT
- Segmentation
- Library

RPG/3000

RPG, the Report Program Generator, is a specialized "high level" programming language developed to provide the business community with a convenient means of producing a wide variety of printed reports. RPG can handle jobs ranging from simple tasks such as printing address labels to very complex ones such as an entire payroll process (printing paychecks, payroll registers, and various allied reports). RPG is ideal for producing such documents as inventory lists, billings, invoices, insurance benefit notices, or summaries of sales, profits and losses, and customer transactions. It is also excellent for updating the master files used in producing these documents.

Like all "high level" languages RPG is problem oriented and relatively free of hardware constraints. It is easy to learn, easy to use, and easy to code. It allows you to specify many important operations with a minimum of effort; you merely make simple entries on specially-formatted coding sheets. Because RPG is a standard language available on many different machines the user can submit programs coded in another manufacturer's RPG directly to the RPG/3000 Compiler with little or no re-coding.

FORTRAN/3000

FORTRAN, the FORmula TRANslator, is one of the oldest and most widely used "high level" programming languages. The initial specifications for the language date back to 1954. Although it was originally designed for use in engineering applications FORTRAN is now also used heavily in business environments.

The standard for this language is American National Standard FORTRAN X3.9-1966 as approved by the American National Standards Institute. FORTRAN/3000 is a full implementation of that standard. To provide a more powerful programming language FORTRAN/3000 includes some extended features not covered by the ANSI standard. Some of these extensions are as follows:

- Source programs may be written in free-field as well as fixed-field format.
- Symbolic names may consist of up to 15 characters (instead of 6).
- FORTRAN/3000 programs may call subroutines written in other programming languages (most notably subroutines written in SPL/3000).
- Character-type data may be used to facilitate string manipulation.
- Arrays may have up to 255 dimensions (instead of 3).

BASIC/3000

BASIC, the Beginner's All-purpose Symbolic Instruction Code, is a "high level" programming language developed at Dartmouth College in 1963-64. Originally BASIC was used primarily for numerical work, but the introduction of string and file handling (not originally available) has made it an appropriate language for use in industry, commerce, universities, and research establishments.

In addition to all features commonly found in BASIC interpreters throughout the industry, the BASIC/3000 interpreter contains a number of extended capabilities, some of which are:

- Four numeric data types (real, integer, complex, extended precision).
- Mixed-mode arithmetic.
- Formatted output.
- Program chaining with common storage.
- External subroutine calls.
- Strings and string arrays.
- File creation and purging under program control.

The BASIC/3000 Compiler converts BASIC programs, which have previously been debugged and stored using the BASIC/3000 Interpreter, into a machine-executable form. A compiled BASIC program exists in the system as an actual code segment rather than as data in a data file and can be run under the operating system rather than through line-by-line interpreting by the BASIC/3000 Interpreter.

Typically a compiled BASIC program runs many times faster than when run interpretively. Of course, it should be understood that actual speed improvement depends on the type of program and the resource demands on the operating system. CPU bound programs, for example, will realize a speed improvement on the order of 10 to 30 times faster, with certain extraordinary programs running up to 100 times faster. I/O bound programs will run 1 to 4 times faster.

APL\3000

APL, A Programming Language, is a unique "high level" programming language developed for scientific and mathematical applications. In recent years it has been used in commercial areas such as statistics, modeling, and finance. With APL, the user has the ability to express complex calculations in a simple and concise manner.

In order to create a more complete APL product, Hewlett-Packard has added many significant features to the APL language as follows:

- Virtual Workspaces and arrays—a virtual memory scheme is used which allows extremely large virtual workspaces and array size.
- Dynamic Incremental Compiler—APL\3000 compiles and saves the code necessary to execute an APL function which means that there can be a significant speed-up of execution on subsequent runs of a function.
- Report Generation—The commercial formatter provides a versatile and easy to use function for the design of a wide variety of formats and reports.
- APLGOL—A structured programming extension to APL has been added which uses ALGOL-like control structures, with the power of APL operators. APLGOL makes the program simple to read and, therefore, to support and modify.
- File Handling—APL\3000 has full access to the MPE file system through the use of shared variables, which means that from within APL, you can use disc files, tape files, card readers and other devices. The component file system allows simplified file manipulation.
- Error Handling—Secure application environments can be created with programmatic handling of errors.
- Extended Control Functions—A set of functions is provided in APL\3000, which allows you to organize the relationship between user-defined functions in a flexible manner, obtain access to local variables and branch to labels in other functions.
- Friendly Editor—The APL function editor provides the power of a full text editor so that text can be entered, edited, and converted to a matrix or vector of characters for later use.
- Other powerful features—Procedure Calls, Programmatic Access to System Commands, Distributed Systems Extensions, and Double Word Integers.

Note: APL\3000 is available only on HP 3000 Series III computer systems.

SPL/3000

SPL, the Systems Programming Language, is a language designed by Hewlett-Packard especially for writing systems software for the HP 3000. SPL combines the best features of both "high level" and machine-dependent programming languages—it allows the programmer to write software quickly, easily, and efficiently, while producing object programs with good code compression and efficient execution times.

Because of the inherent efficiency of machine-dependent languages, most operating systems, monitors, compilers, and subsystems are written in these languages. However, because SPL combines the efficiency of machine-dependent languages with the simple structure of "high level" languages, all HP 3000 software packages—the operating system, the language compilers, the system utilities, the data base management programs, and the telecommunications programs—are written in SPL.

A further benefit is that programmers who do not know the details of the HP 3000 architecture can still use SPL because of its high-level nature. Also, commonly-executed routines can be written in SPL and called from COBOL, BASIC, RPG, FORTRAN, or APL programs.

See the product reference sheets in Section II for more complete descriptions of the HP 3000 languages.

Chapter 3

MANUFACTURING ENVIRONMENTS

ENGINEERING DATA CONTROL
STANDARD PRODUCT COSTING
INVENTORY AND ORDER STATUS
MATERIAL REQUIREMENTS PLANNING

Available in North America, Europe, and Australia

MFG/3000 is an application software product which helps manage the material planning and control functions of a discrete manufacturing company. The product maintains information that is used to plan material requirements, establish standard product costs, and to recognize and plan priorities effectively. The primary objective of the product is to minimize inventory investment while maximizing customer satisfaction.

As shown in Figure 3-1, MFG/3000 consists of four products:

EDC/3000—Engineering Data Control software which maintains descriptive, cost, and planning information and Bill of Material and routing data about the parts in your manufacturing operation.

SPC/3000—Standard Product Costing software which provides the capability to roll-up current costs and to accurately set standard costs.

IOS/3000—Inventory and Order Status software which tracks planned issues (allocations) and planned receipts (workorders and purchase orders), and maintains stockroom inventory balances.

MRP/3000—Material Requirements Planning software which generates the materials plan with recommendations about what and how much material to order and when to order it.

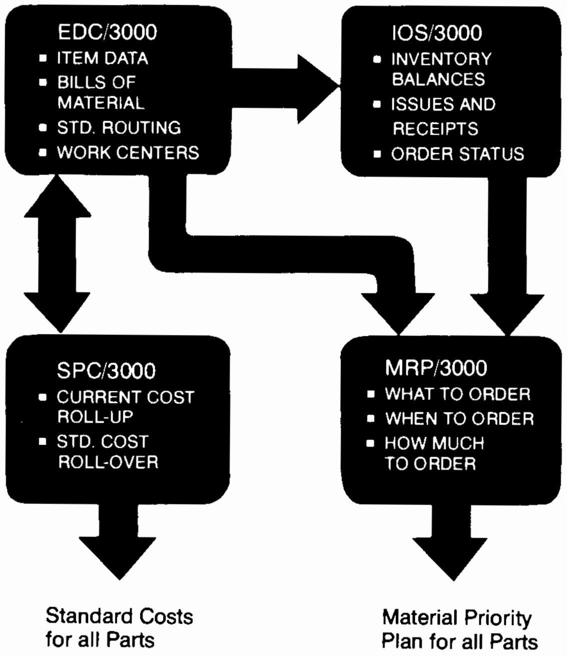


Figure 3-1. MFG/3000

MFG/3000 is designed to be used with Hewlett-Packard terminals. By placing terminals in the user's work area, information is available when and where it is needed. Data is input by the people responsible for its generation. Timely and complete information for operations and decision-making is available for those who need it, when they need it.

The user is guided through MFG/3000 by a simple menu selection approach. This technique eliminates the need to learn a new language or set of commands. The user simply selects the function (i.e., retrieve Bill of Material, review stock status, etc.) from the menu and MFG/3000 displays the desired information on the screen.

MFG/3000 products are supplied with predefined screen formats, transaction edit criteria, and retrieval and report formats. They can be run "as is," or, if required, many external features of the software can be modified by nonprogrammers. The items that affect terminal users and are easily modified include:

- Default values for EDC addition transactions
- Screen sequencing
- Certain field editing characteristics
- Most transaction/screen formats
- Originator numbers and capabilities
- Documentation screens
- Maximum number of records in each data base.

In addition to the implementation consulting and comprehensive customer training for each product, Hewlett-Packard offers a full range of services and documentation for each of the products.

MFG/3000 utilizes Hewlett-Packard's award-winning data base management system: IMAGE/3000. In addition, Hewlett-Packard's English-based QUERY/3000 is available to generate reports and retrievals not specifically provided by MFG/3000. QUERY/3000 allows users to quickly produce reports themselves without involving a programmer. This increases the productivity of your manufacturing operation by providing quick response for required information.

EDC/3000

Engineering Data Control (EDC/3000) maintains information about every item in the materials inventory, including part numbers, product descriptions, cost data, bills of material, bills of labor, and engineering change information.

EDC/3000 is designed so that data is easily entered and changed by the users. With formatted data screens, they can quickly enter and review information on parts and bills of material. Terminals can be located in the work areas of the people responsible for maintaining part description, cost and planning information.

Information concerning bills of material, where used, and routing for parts can be reviewed on-line. Printed reports supplied on demand by EDC/3000 include: Single Level Bills, Indented and Summarized Parts Lists, Where Used Lists, and Routing Lists.

Using EDC/3000

In a typical EDC/3000 installation, interactive CRT terminals are placed in all areas that require bill of material and engineering documentation. Requests for documentation are satisfied directly by on-line retrievals at the terminals, which are located, perhaps, in production engineering, R&D, production control, manufacturing specifications, and other appropriate departments. Requests for lengthy reports entered by the terminal operator are serviced in regularly scheduled batch runs.

Data entry and editing using on-line CRT terminals can be accomplished either by a central group responsible for manufacturing documentation or by each individual responsible for specific portions of the data. Production control, for example, might be responsible for the content of lead time fields, while production engineering controls bills of material. EDC/3000 employs regularly scheduled batch computer runs to actually update the data base with the transactions collected and edited by the CRT data entry. This batch updating facilitates control, security, and synchronization of inputs.

SPC/3000

Standard Product Costing (SPC/3000), in conjunction with data maintained in the EDC/3000 data base, provides a manufacturer the capability to accurately calculate the costs associated with each product and to easily validate these costs before establishing them as standards.

All current cost information for material, labor, and overhead may be reviewed on-line; providing timely and accurate access to vital information.

Using SPC/3000

SPC/3000 uses customer supplied current cost information entered through the EDC/3000 data entry screens and maintained in the EDC/3000 data base. This information might normally be entered by purchasing personnel for material costs, production control personnel for workcenter and routing information, and accounting personnel for labor and overhead rates.

Cost editing identifies current costs that have not been properly initialized or have logical inconsistencies, such as a purchased part with no material cost. A Cost Roll-up Edit Report is printed that identifies all detected conditions that could cause an unreliable or inconsistent Cost Roll-up.

Cost Roll-up is a major function of SPC/3000. Roll-up may be stopped at the completion of each assembly level, or it may be done non-stop through all levels. Specific part selection, using one of four different criteria, provides a high degree of flexibility. Costs at each level are accumulated and then combined with the costs for all lower levels before being rolled-up to the next higher level. The Roll-up continues until the current cost is determined for the selected part. The Roll-up is accomplished without affecting the standard cost for the selected part. A product cost report is printed for each selected part and each of its fabricated components. These reports can then be used by the finance or accounting department to determine the acceptability of the calculated costs for their use as standard costs.

Cost roll-over converts the results of the latest Cost Roll-up to standard costs by replacing the old standard cost values with the existing current cost values. Standards are established only after all information is edited, accumulated, and verified. Standard costs are carefully determined costs and may be used for setting goals, establishing budgets, measuring performance, or determining product prices.

IOS/3000

Inventories can be classified in three ways: stockroom, work in process, and finished goods. Inventory and Order Status (IOS/3000) controls the stockroom inventory by maintaining complete and accurate records of all actions that affect inventory balances:

- Receipt of purchase orders and workorders
- Backorder filling
- Material issue (planned and unplanned)
- Adjustments

All record keeping and updating is done on-line, providing immediate and accurate information.

Using IOS/3000, the user can create, modify, and maintain records on workorders and purchase orders. When a workorder is partially issued, a backorder is automatically created.

IOS/3000 automatically allocates all open workorders. Using the EDC/3000 Bill of Material, IOS/3000 explodes the workorder one level to determine the components required to manufacture it. By allocating the workorder, IOS/3000 is able to identify potential parts shortages before they occur. This aids in eliminating backorders.

IOS/3000 notes exceptional conditions and reports them to the people responsible for action (buyers, schedulers, stockroom personnel). The timely notification of exceptions to the Inventory Plan, represented by planned receipts (orders) and planned issues (allocations), can allow corrective action before the results become disastrous. The on-line nature of IOS/3000 helps foster accuracy of inventory and order data which can be used by MRP/3000 for generation of a total materials plan. IOS/3000 is thus responsible for the implementation and control of your material plan.

Using IOS/3000

IOS/3000 offers the capability to control stockroom inventory movement. Orders to replenish stock are issued either to outside vendors or a customer's own production facility.

A vendor purchase order is entered at a CRT terminal in the purchasing department, and includes the necessary tracking information such as quantities and due dates, as well as descriptive data. Once such an entry passes the customer defined edits, it is immediately added to the order file. When material is received, a CRT terminal with attached hard copy printer in the receiving area is used to update the order to reflect the receipt, print an accounts payable material receipt document, and increment on-hand inventory or inspection inventory as appropriate. If backorders exist for the part being received, a document is generated indicating the quantity of parts back ordered and the department requiring them. This insures prompt filling of backorders once an out-of-stock item is received. All receipts to the stockroom are processed in this manner.

A workorder to an in-plant production facility is entered at a CRT terminal by the production control department using procedures similar to those for purchase orders. This workorder, since it is an authorization to build an assembly, part, or product, requires withdrawal of component parts from stock. These component withdrawal requirements are obtained in a batch run (usually done daily) which "explodes" the workorder quantity by the bill of material from EDC to create allocations which, on the appropriate date, will become requisitions for the correct amounts of each component.

By having allocations available before the actual issue of materials, pre-shortage reports which match all allocations for a particular part to the balance on hand are produced to point out possible parts shortages.

Just prior to the start date of the workorder, picking lists are produced to control the issue of material from the stockroom. Individual requests for parts are serviced directly at a terminal in the stockroom, as are responses to the picking lists.

When the workorder is completed, the material is received back into the stockroom in the same manner described earlier for purchase orders.

In addition to on-line control over all issues and receipts to the stockroom, a cycle counting system based on total usage value of the parts (ABC value classification) helps insure inventory record accuracy.

MRP/3000

MRP/3000 is a material requirements planning system which simulates the complex flow of material in a manufacturing company. If the material is short, it suggests an order for an appropriate quantity of the material, or expedites an existing order. Current and anticipated demand for a part is matched with the current and anticipated supply for that part to find potential conflicts, and to suggest corrective actions whenever supply and demand get out of agreement.

MRP/3000 starts with up-to-date information about current status provided by EDC/3000 and IOS/3000. Information about future requirements is provided by means of the master schedule. Defined time and quantity information is used to calculate material requirements for each inventoried part according to the planning horizon desired. A material plan is generated which can be used to evaluate priorities, anticipate potential problems, adjust future plans, and control material costs.

MRP/3000 is a net regenerative system—regenerative in that a complete materials plan is generated every time MRP/3000 is run, net in that demand is balanced against available and projected supplies to determine net requirements. These characteristics combine to produce a fully documented, visible materials plan on every run.

MRP/3000 provides visibility of the source of all demands by pegging the requirement to the order that created it. A specific due date is also assigned to each order or suggested order, making MRP/3000 a "bucketless" material requirements planning system.

Using MRP/3000

The bill of material information from the EDC/3000 data base and the current inventory levels and order status from the IOS/3000 data base are combined by MRP/3000 to produce a series of reports used by production control and purchasing to plan inventory procurements.

MRP/3000 takes the independent demand for products represented by the customer's master schedule and calculates time-phased demand for component parts. Once all demands for a part or assembly have been determined, MRP/3000 will allocate current inventory and orders to these demands and then suggest new orders based on the part's order planning algorithm. The current and suggested orders are then offset by the assembly leadtime, modified by yield factors, and "exploded" via the bill of material to form dependent demand for this assembly's component parts.

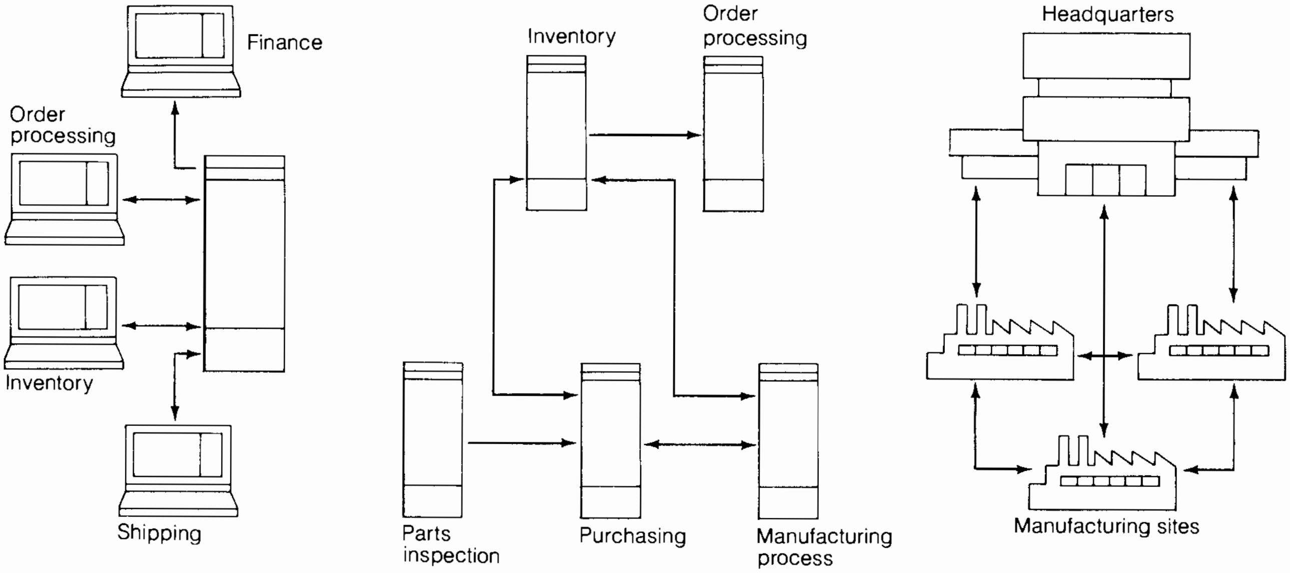
The parts controller receives an exception report that highlights all MRP/3000 suggestions for "push outs" and "pull ups" of existing orders, as well as any new suggested orders that should be looked at before the next MRP/3000 run. Exceptions are detected according to individual controller parameters that are maintained via a CRT terminal.

The controller may take appropriate action based directly on the exception report, or may prefer to investigate the situation that triggered the suggested action. An action report, which displays all supply and demand entries for each part, can be consulted to determine the cause of the exception. The parts controller (inventory planner) has the ability to override any MRP/3000 actions. Once decisions on the actions required have been made, the controller may update IOS/3000 to reflect his plan for order reschedules and new order releases.

Chapter 4
DATA COMMUNICATIONS

HEWLETT-PACKARD DISTRIBUTED SYSTEMS NETWORK (HP-DSN)

Hewlett-Packard Distributed Systems Network (HP-DSN) is the high-level, user-oriented network architecture that allows HP 3000 computer systems to communicate in distributed data processing networks. As shown in Figure 4-1, HP-DSN allows processing to be distributed either geographically or functionally in your organization. However, it is flexible enough to allow for either centralized or decentralized control. HP-DSN networks need not adhere to any particular configuration; hierarchical, linear, or circular networks of HP 3000 nodes in any combination are accommodated by HP-DSN. There is no absolute limit to the number of HP computer systems that may communicate in an HP-DSN network. HP-DSN allows communication between the HP 3000 and other HP computer systems, such as the real-time, interrupt-driven, HP 1000. Further, batch communication between HP 3000's and IBM or IBM- compatible mainframe systems are also included in HP-DSN (Fig. 4-2). These capabilities allow you to choose the HP system which exactly meets your requirements—be it for general accounting or factory data collection—and be assured of compatible communications between systems. The IBM communication capabilities allow you to optimize the trade-off between totally centralized and totally decentralized processing, and provide a smooth transition from strictly centralized to distributed processing.



If you need more timely information or want to improve information accuracy in your manufacturing operations or if you need to integrate a comprehensive manufacturing information network into the operational aspects of managing your business, or if you want to exchange information interactively among a number of manufacturing sites...with HP-DSN you can define and implement a distributed processing system to handle your current requirements with the appropriate level of control while providing a growth path to meet your future needs.

Figure 4-1. HP's range of distributed processing solutions lets you put computer power where the work is done.

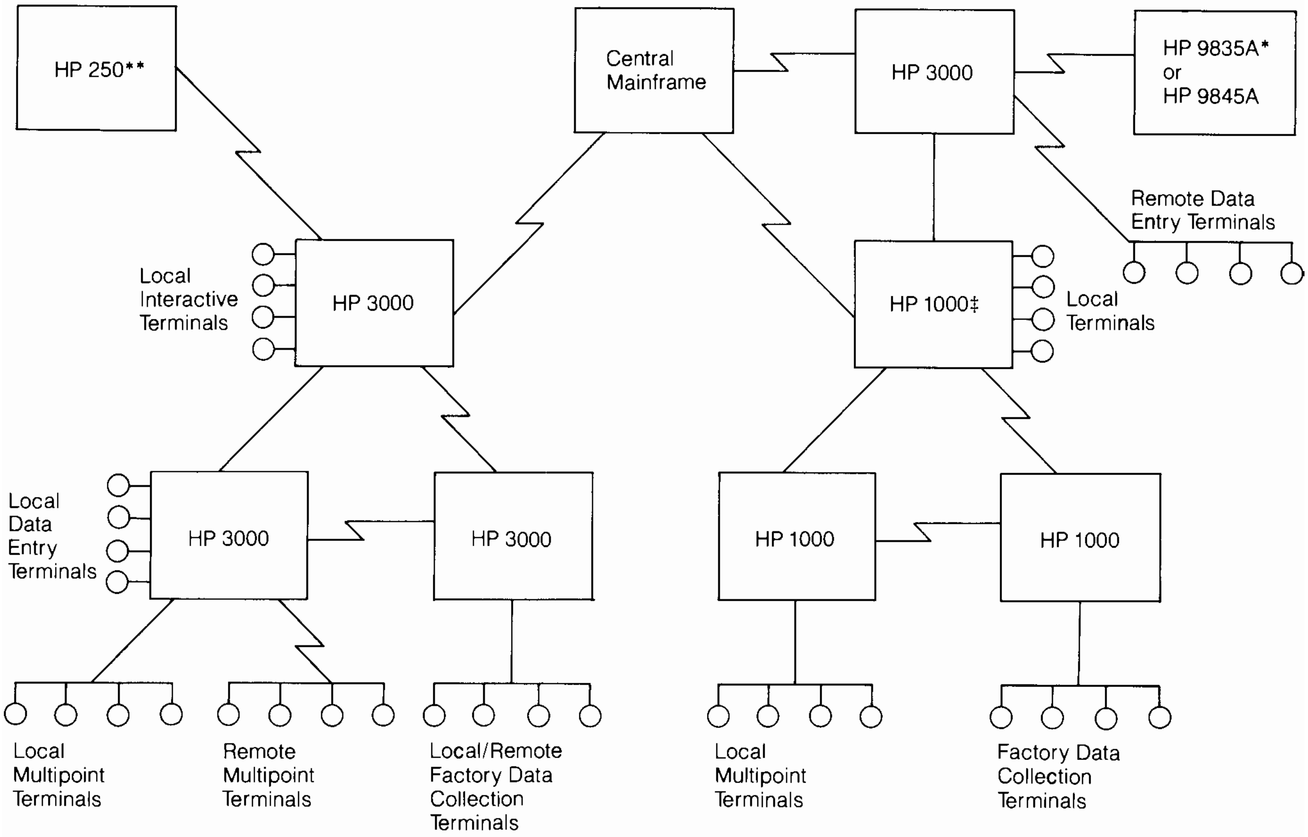

HP 250**
Central Mainframe
HP 3000
HP 9835A* or HP 9845A
Remote Data Entry Terminals
Local Interactive Terminals
HP 3000
HP 1000‡
Local Terminals
Local Data Entry Terminals
HP 3000
HP 3000
HP 1000
HP 1000
Local Multipoint Terminals
Remote Multipoint Terminals
Local/Remote Factory Data Collection Terminals
Local Multipoint Terminals
Factory Data Collection Terminals

Figure 4-2. Hewlett-Packard Distributed Systems Network (HP-DSN)

* HP 9835A and 9845A are desk-top computers that can act as terminals to HP computer systems.

** The HP 250 system can act as a terminal to the HP 3000.

‡ The HP 1000 is a scientific, real-time, interrupt-driven computer system, widely used in factory data collection and instrumentation environments. Devices that are compatible with IEEE Std. 488-1975 (HP-IB) are supported by the HP 1000.

HP-DSN Objectives and Implementation

HP-DSN consists of a set of design objectives, and a set of hardware and software products that implement these objectives. The objectives of HP-DSN may be summarized as follows:

- high-level user interfaces (user freedom from having to know communications protocols or line characteristics)
- device and network independence
- sharing of resources among systems (hardware, software, and data)
- communication with both HP and non-HP equipment
- network diagnosis and recovery
- data and communication integrity and security

The Hewlett-Packard Distributed Systems Network is implemented as a number of functional layers in software with only the high level system services visible to you. The layered approach, depicted in Figure 4-3 provides flexibility for additions and enhancements at each layer. Advantages of this approach include both stability and flexibility: stability because changes can be made to the internal layers without requiring you to alter application programs, and flexibility because the modular design of the network easily accommodates changes as a result of technological improvements. The specific functions assigned to each layer are described below.

HEWLETT-PACKARD DISTRIBUTED SYSTEM NETWORK

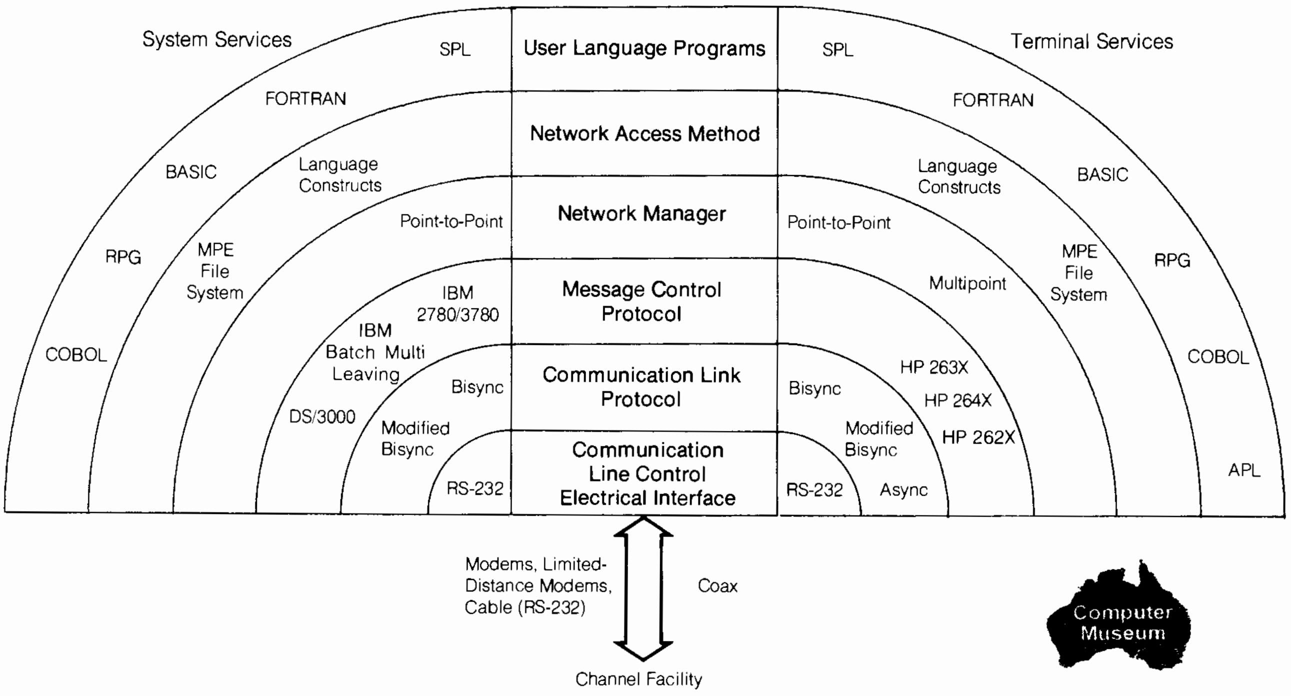


Figure 4-3. HP DSN is implemented as a series of software layers. Improvements can be made in any layer without affecting the highest level, your application programs.

Network Access Method

The network access method gives you the network capabilities at the application level through callable MPE intrinsics, system services, and specific language I/O constructs. The services include:

- Remote file access
- Remote data base access
- Remote command execution (virtual terminal access)
- Remote peripheral access
- Remote program management

Network Manager

The network manager layer is aware of the network topology and is responsible for managing network error recovery and the various topology dependent functions such as polling.

Message Control Protocol

This layer provides control functions, addressing information, message type, and other requirements to effect end-to-end transmission.

Communication Line Protocol

The communication line protocol layer takes care of the "grammar" or protocol by which two or more systems can exchange information in an efficient and reliable manner.

Communication Line Controller

This is the physical interface that defines the number of circuits and the meaning of the signals on those circuits.

HP-DSN Products

HP-DSN products fall into two categories: software and hardware. HP-DSN software subsystems, all of which run under the control of the MPE operating system, are:

- MTS/3000—Multipoint Terminal Software/3000—synchronous, multipoint or multidrop terminal communications (asynchronous terminal communications are a capability inherent in MPE and do not require an additional HP-DSN software subsystem).
- DS/3000—Distributed Systems/3000—interactive data communications between HP computer systems
- RJE/3000—Remote Job Entry/3000
  —IBM 2780/3780 emulation
- MRJE/3000—Multileaving Remote Job Entry/3000—emulation of a multileaving batch workstation (IBM 360/30), providing job management services for HASP II, JES 2, JES 3, and ASP job entry subsystems on the mainframe

HP-DSN hardware includes:

Synchronous Communications Interfaces—

- Synchronous Single Line Controller (SSLC)
- Hardwired Serial Interface (HSI)
- Intelligent Network Processor (INP)

Modems—

- 4800 bits/second, dial up or leased lines
- 9600 bits/second, leased lines

and the Asynchronous Repeater for multipoint communication. The Reference Sheets section of the manual should be consulted for further information on the hardware products.

The HP-DSN software and hardware products afford a complete solution to your distributed processing requirements from Hewlett-Packard—except for the telephone line!

TERMINAL COMMUNICATIONS

The terminal handling capabilities of the HP 3000 provide you with a number of alternatives for use in designing the most cost-effective and functional communication links to terminals. Asynchronous or synchronous communication may take place over point-to-point or multipoint links. Links may use modems, or be hardwired, with several transmission speeds possible.

Point To Point Terminal Communications

Point to point terminal communications is provided on the HP 3000 Series III by the Asynchronous Terminal Controller (ATC). Up to 4 ATC's can be connected to give a maximum of 64 terminals including the system console. Modems supported by the ATC are Bell types 103A, 113B, 202S*/T, 212A and Vadic VA3400.

For the Series 30 and 33, the Asynchronous Data Communication Controller (ADCC) allows 4 terminals to be connected. Each ADCC main can have one ADCC extender to allow a total of 8 terminals. The maximum number of terminals is 32 for the Series 33 and 16 for the Series 30. Modems supported by the ADCC are Bell types 103A, 212A, and 202S*/T.

The maximum speed for an asynchronous terminal on an ATC is 2400 bps and 9600 bps on an ADCC. For dial-up connection, speed sensing is automatic up to 2400 bps. Terminals can be configured to the operating system (MPE) as data entry terminals under your program control, or as log-on terminals accessing all the capabilities of the HP 3000. Terminals normally operate in character mode, except when accessed via HP VIEW/3000 when block mode is employed. If you wish to access terminals in block mode directly (i.e., not using HP VIEW/3000), you must provide the detection and correction facilities for transmission errors by calling operating system routines.

Multipoint Terminal Communications

Multipoint Terminal Software (MTS/3000) permits 9600 bps data transmission between the HP 3000 and multiple HP 2640 series multipoint terminals using a single communication line. The terminals may be connected to the computer by means of a modem (remote access) or may be hardwired to the computer (local access). With MTS/3000 you use MPE commands and file system intrinsics to communicate with the terminals. You may also initiate sessions from the terminals and thus access all the resources of the MPE operating system.

**For performance reasons the use of 202S models is not recommended.*

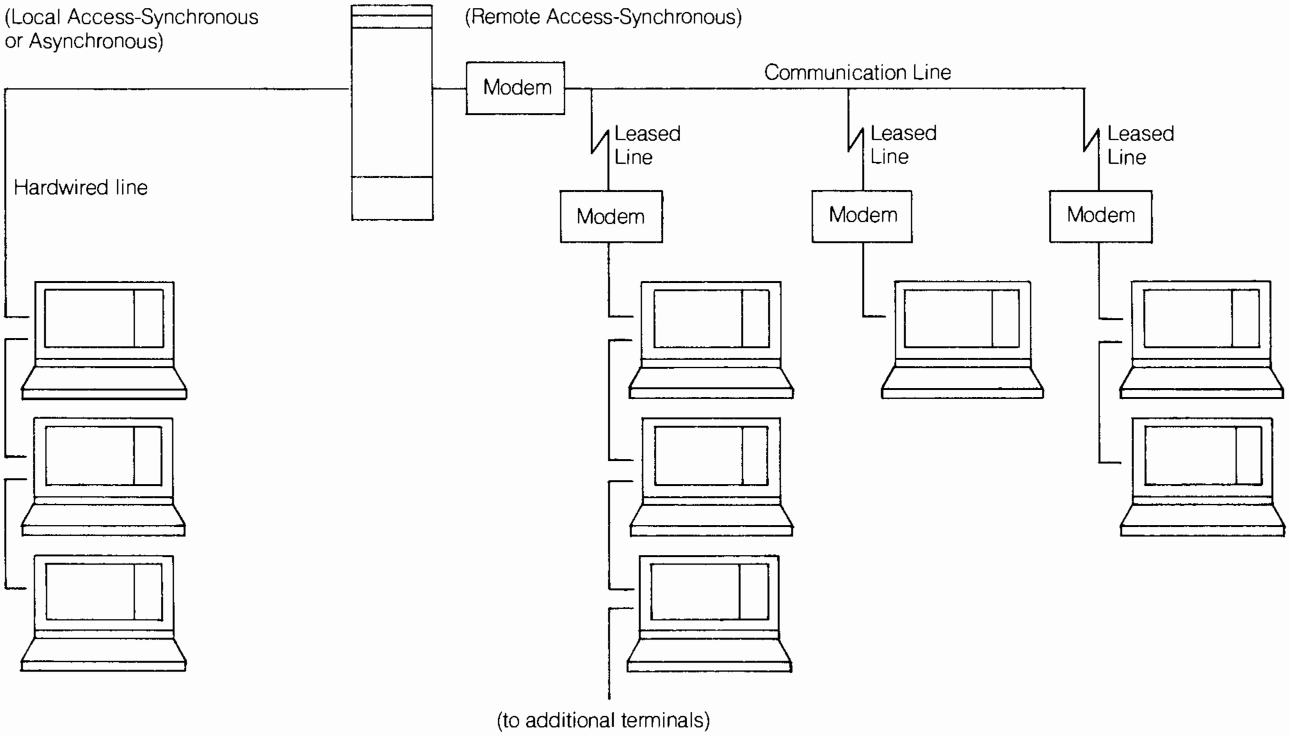

(Local Access-Synchronous or Asynchronous)
(Remote Access-Synchronous)
Modem
Communication Line
Leased Line
Leased Line
Leased Line
Hardwired line
Modem
Modem
Modem
(to additional terminals)

Figure 4-4. Typical MTS/3000 network

Terminal Networks

Figure 4-4 is an illustration of a network of multipoint terminals. When terminals are connected by modems (remote access) communication is synchronous; hardwired terminals (local access) may be either synchronous or asynchronous. Up to 2000 feet (610 meters) of cable may separate individual hardwired terminals. However, the first terminal must be within 50 feet (15 meters) of the system unless using the Asynchronous Repeater to extend the cable distance. Multiple lines of terminals may be attached to an HP 3000 computer system, with a maximum addressing limit of 255 terminals multidropped on these lines.

Data Transmission

Terminals may transfer data at speeds up to 9600 bps. They may operate in either log-on (interactive) or data entry mode. In either mode, you enter data into the terminal's memory using the cursor positioning capabilities, TAB key, and RETURN key. This data can then be edited as much as desired until the ENTER key is pressed, transferring the data to the computer. Variable length blocks of data may be transferred in this manner.

Power Down Bypass Cable

A power down bypass cable is available for use with MTS/3000. The cable, ordered with the multipoint terminal, enables the system to bypass a terminal which has no power, so that the remaining terminals in the daisy chain group are not affected.

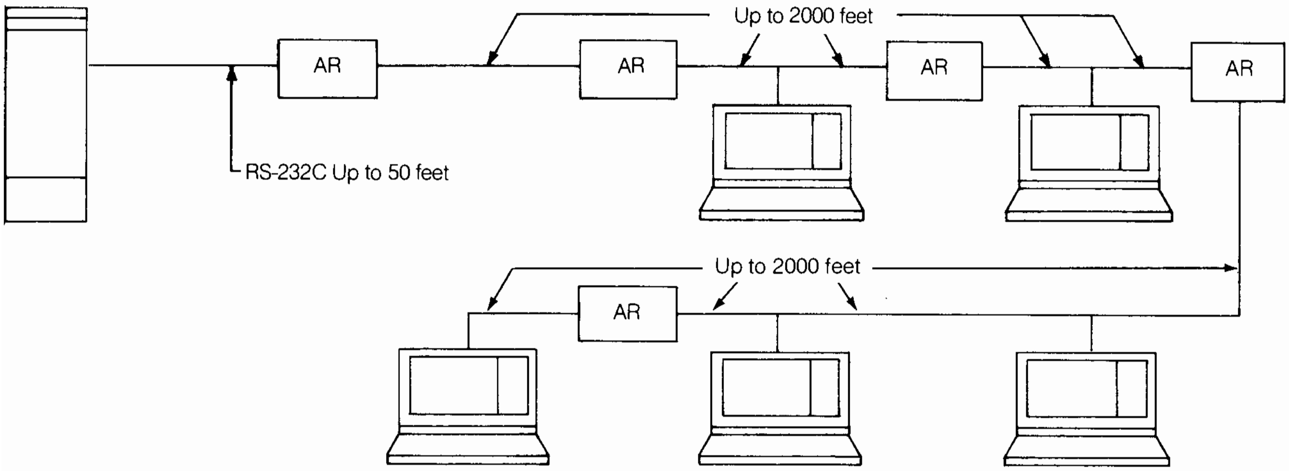


Figure 4-5. Use of the asynchronous repeater in multipoint terminal environment

Longer Distance Direct Connections

The Asynchronous Repeater (AR) is a stand-alone device which converts standard RS232C communication signals to levels compatible with the HP 2640 series of terminals. It is useful in either an asynchronous multipoint or point-to-point environment.

By adding an AR in a multipoint environment, the first directly connected terminal on a line can be located up to 2000 feet (610 meters) from the computer. This removes the 50 foot (15 meter) limitation imposed when an RS232C direct connect interface is used. In addition, each AR can extend the maximum cable distance between individual multipoint terminals (or a group drop of terminals) by an additional 2000 feet (610 meters). Figure 4-5 shows examples of the possible AR uses in a multipoint terminal environment. Note that additional ARs may be added to further extend the maximum distance allowed.

ARs may also be used in the point-to-point environment where terminals are attached to the system via the Asynchronous Terminal Controller.

Terminal Peripheral Devices

The multipoint terminal can be viewed as a processor with several peripheral devices:

- keyboard
- display
- two cartridge tape units (optional)
- printer (optional)

These peripheral devices can be controlled by transmitting the appropriate escape sequences to the terminal from your application program. The capabilities provided by the escape sequences include data transfers from the HP 3000 to a peripheral device, from a peripheral device to the HP 3000, and from one peripheral device to another within the same terminal. An escape sequence also exists to retrieve status information for a peripheral device.

These capabilities can be used with both point-to-point and multipoint communications. In multipoint transmission the peripherals share the terminal's multidrop line. In a point-to-point environment using the Asynchronous Terminal Controller (ATC), the peripheral shares the point-to-point line.

DISTRIBUTED SYSTEMS/3000

The Distributed Systems/3000 (DS/3000) software subsystem is used for intercommunication between HP computer systems. It provides a complete set of network communications services so that programs, files, peripheral devices, and processing can be shared in a network. No knowledge of communications programming is required to use DS/3000. In fact, most DS/3000 capabilities are available to you with no programming effort once the software and required hardware interfaces are installed.

DS/3000 allows an HP 3000 to communicate with both HP 3000 and HP 1000 systems. The HP 1000, with its Real-Time Executive (RTE) operating system, is a high-performance computer system designed for real-time computation and instrumentation applications.

Communication between systems occurs in a bidirectional interleaved fashion using hardwired coaxial cables for HP 3000 to HP 1000 communications, and both hardwired and modem connections for linking HP 3000 systems.

DS/3000: A Management Perspective

DS/3000 was designed to be particularly useful in commercial applications that involve transaction processing and are geographically dispersed. DS/3000 has four advantages of special interest to management:

- Easy user access among all HP 3000 computers in the network
- Simplified network management
- Ease of network implementation
- More efficient use of all systems resources

Easy User Access

In an HP Distributed Systems Network, any HP 3000 user has every other HP 3000 in the network at his fingertips. All system commands may be executed remotely simply by inserting the word REMOTE in front of the command. No knowledge of the communication link used —be it a direct connection (hardwired link) or telephone link—is required. Programmers can access remote data files using the normal input/output statements of each programming language, just as if the files were local. No special training for users or programmers is required.

Simplified Network Management

DS/3000 makes the system or network manager's job easier. Although computer power is distributed to the location where it is needed, control of the network may be maintained centrally. The remote command processing feature allows a single system manager to control all satellite HP 3000 systems from the central EDP facility, if desired. The manager may assign each user different capabilities on different computers in the network with each operating system automatically enforcing the capability assignments. This affords functional dedication of each computer in the network. For example, all program development may be done on one machine, all batch on another, and all on-line transaction processing on a third computer.

Changes in network topography can be made with no reprogramming. For example, when a company with two divisions using a single HP 3000 grows to the size that each division needs its own computer system, the two systems can be linked together and access common files with no application program changes. A network manager may define and re-adjust his network to best meet the needs and growth of his organization, without having to worry about the effects on existing programs and files.

Data security and accounting for resource usage are responsibilities of the network manager. DS/3000 automatically affords full data security for each computer in the network. Full control of which users can read or write specific data is provided. Given the proper security codes, either user or manager can obtain exclusive access to any specified link in the network. Resource usage is automatically logged for CPU time, disc space, and connect time for all users, whether remote or local.

Ease of Implementation

Using DS/3000, the network may be configured in any combination of rings, stars, or strings—whichever best reflects the structure of your organization. Among HP 3000's, you have a choice of switched, leased, or hardwired lines. These different line types may be mixed throughout the network, so you can choose the most convenient and/or economical type of line for each particular link.

Efficient Use of System Resources

With DS/3000 several systems can share the use of expensive peripherals via communications lines, thus keeping capital investment to a minimum without sacrificing capability. The peripherals appear to you just as if they were on your local system. Each physical communications line in an HP Distributed Systems Network can be used concurrently by many different batch and interactive applications originating from either end of the line. DS/3000 automatically processes all data being transferred over each line, so that each application is essentially unaware that anyone else is using the same line.

Time is a precious commodity in any organization. Within an HP Distributed Systems Network, a wide variety of processing can be in progress simultaneously, including local and remote batch jobs, problem solving, inter-system program-to-program communication, and remote job entry to your mainframe.

3000/3000 and 3000/1000 Communications Links

The HP 3000 to HP 3000 and HP 3000 to HP 1000 communications links give you the opportunity to select the appropriate blend of computing power for your specific requirements. After either of these communications links has been established, two commands enable you to make use of DS/3000 network services to other network nodes. The command DSLINE identifies the remote computer. Telephone number, identification list, maximum transmission buffer size, and the activation of data compression may also be specified. The other command, REMOTE, is used to direct locally entered commands to a remote computer identified in the DSLINE command. With these commands, the following capabilities are available:

- Remote Command Execution (RCE) allows users of a local system to employ exactly the same set of operating system commands as are available to a user at the remote system.
- Remote file access (RFA) gives you full access to the data files on a remote system.
- Remote peripheral access (RPA) works just like RFA, giving you full access to peripheral devices on the remote computer.
- Remote data base access (RDBA) lets you access a remote data base on a remote computer system on a transaction-by-transaction basis.
- Program-to-program communications allows your application programs being run on separate systems to interactively exchange data and control information.

The following paragraphs contain additional information about each of these capabilities.

Remote Command Execution

With remote command execution you can execute the entire set of MPE commands on a remote HP 3000 while connected to a local HP 3000 or a local HP 1000. You can also execute system level RTE operator commands on a remote HP 1000 while connected to a local HP 3000.

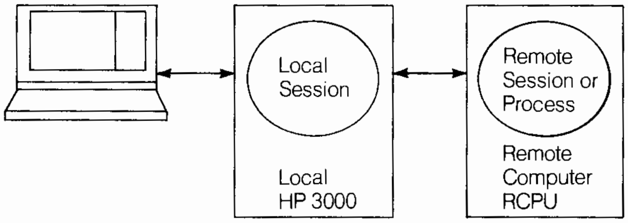


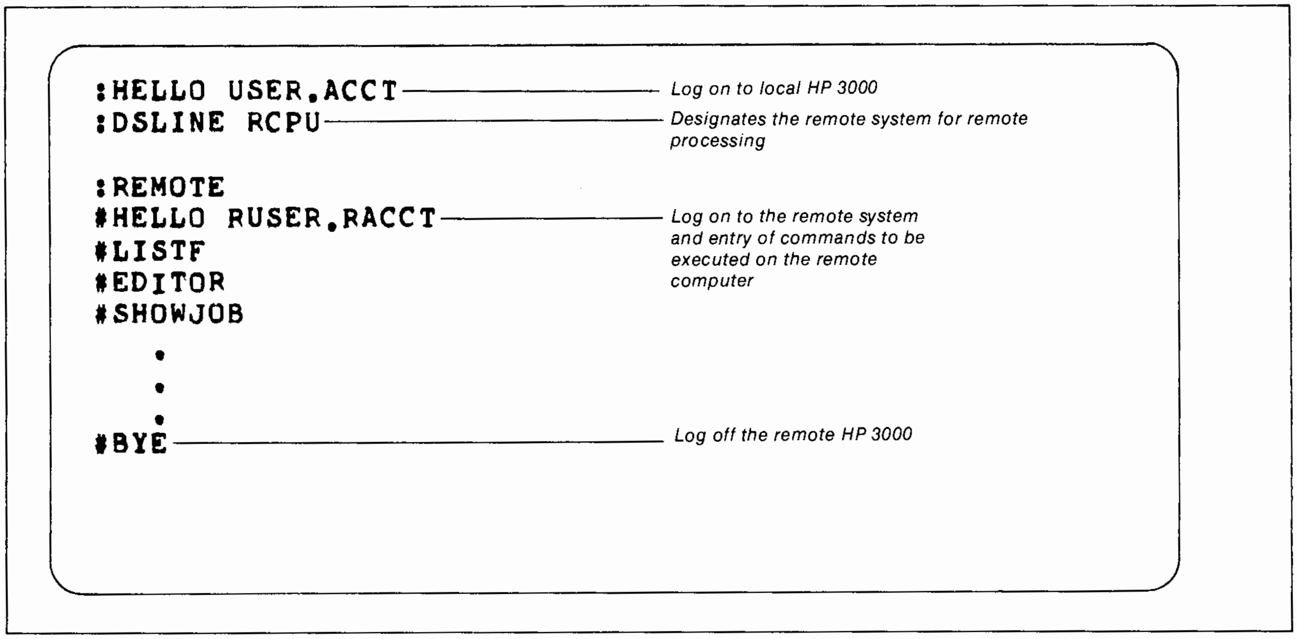
Figure 4-6. Remote command execution allows you to issue commands to a remote system as if the local terminal were connected directly to the remote system.

To execute a remote MPE or system level RTE operator command from a local HP 3000, you simply enter the REMOTE command at the local terminal as follows:

:REMOTE command

where command is the desired MPE or RTE operator command in its normal format.

The following is an example of HP 3000 to HP 3000 communications. Commands are issued from a local terminal to a remote computer for execution.



```
:HELLO USER.ACCT        Log on to local HP 3000
:DSLINE RCPU            Designates the remote system for remote
                        processing

:REMOTE
#HELLO RUSER.RACCT      Log on to the remote system
#LISTF                  and entry of commands to be
#EDITOR                 executed on the remote
#SHOWJOB                computer
    .
    .
    .
#BYE                    Log off the remote HP 3000
```

Issuable HP 3000 to HP 3000 commands include all user commands, system supervisor commands, account manager commands, and system manager commands.

You can use the same REMOTE command to issue standard HP 1000 system level commands to be executed on a remote HP 1000 computer system.

The procedure for issuing a remote MPE command from a local HP 1000 has a slightly different format. After the remote HP 3000 is accessed and the communications link is established, you need enter only the following at the local HP 1000:

```
#command
```

where command is the desired MPE command in its normal format.

Replies generated at the remote HP 3000 are returned to you at your terminal on the local HP 1000.

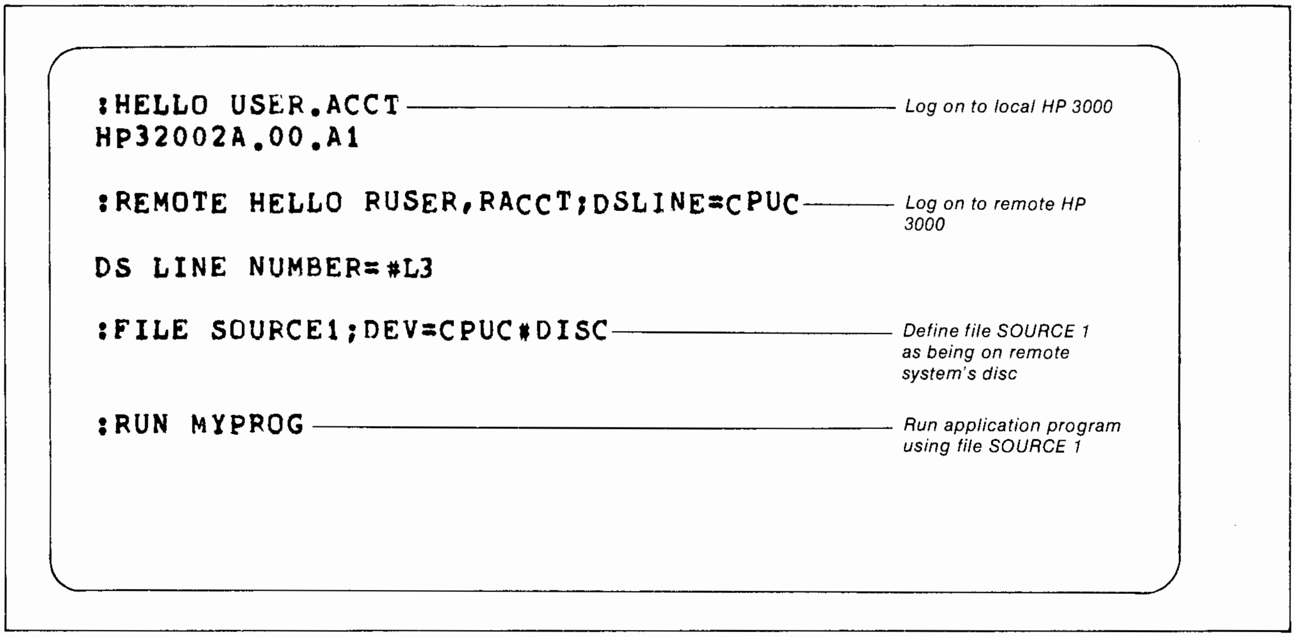
Remote File Access (RFA)

With DS/3000 linking an HP 3000 with either another HP 3000 or an HP 1000, you have access to the files of the remote system. There are three methods by which you can access remote files: utility programs (3000/3000 link only), standard language input/output statements (3000/3000 link only), and remote file access intrinsics (3000/1000 link).

HP 3000 to HP 3000 RFA: With the utility programs and editor, you merely issue a local MPE FILE command on your terminal prior to running the utility. The FILE command is used to define the desired remote file. Included in the FILE command must be a remote device specification denoting the location of the desired file. The utility is run as though it were accessing a local file.

Access using standard language input or output statements permits local programs written in any language to define files and manipulate complete files or file records on another remote HP 3000. All that is required is a FILE command which may be made external to the program or may be included in the program. This command specifies the location of the target file. Subsequently, the remote file may then be utilized on a record-by-record basis or as a complete file as though it resided on your local computer.

For example, you can run your local application program (MYPROG) which accesses the remote file SOURCE1 with the following command sequence:



```
:HELLO USER.ACCT
HP32002A.00.A1

:REMOTE HELLO RUSER,RACCT;DSLINE=CPUC

DS LINE NUMBER= #L3

:FILE SOURCE1;DEV=CPUC#DISC

:RUN MYPROG
```

Log on to local HP 3000

Log on to remote HP 3000

Define file SOURCE 1 as being on remote system's disc

Run application program using file SOURCE 1

Local HP 3000 application programs may utilize the remote file access intrinsics to access standard MPE files which reside on a remote HP 3000. In addition to accessing standard MPE files on a remote HP 3000, DS/3000 also allows you to access remote KSAM/3000 (Keyed Sequential Access Method) files using standard KSAM intrinsics from your local HP 3000.

Regardless of which method is used, the system file security available on a single HP 3000, is maintained across the DS/3000 link.

HP 3000 to HP 1000 RFA: To access files on an HP 1000's moving-head disc, flexible disc, or tape mini-cartridge from an HP 3000, you can write an HP 3000 program that makes use of a set of intrinsics, called remote FMP, that allow access to the remote HP 1000 file system. These intrinsics have a direct correspondence to the standard HP 1000 RTE file management package (FMP).

HP 1000 to HP 3000 RFA: Local HP 1000 application programs may use the remote file access intrinsics to define, control, and access disc files on a remote HP 3000.

To access HP 1000 files, you enter the following at the local terminal:

Remote Peripheral Access

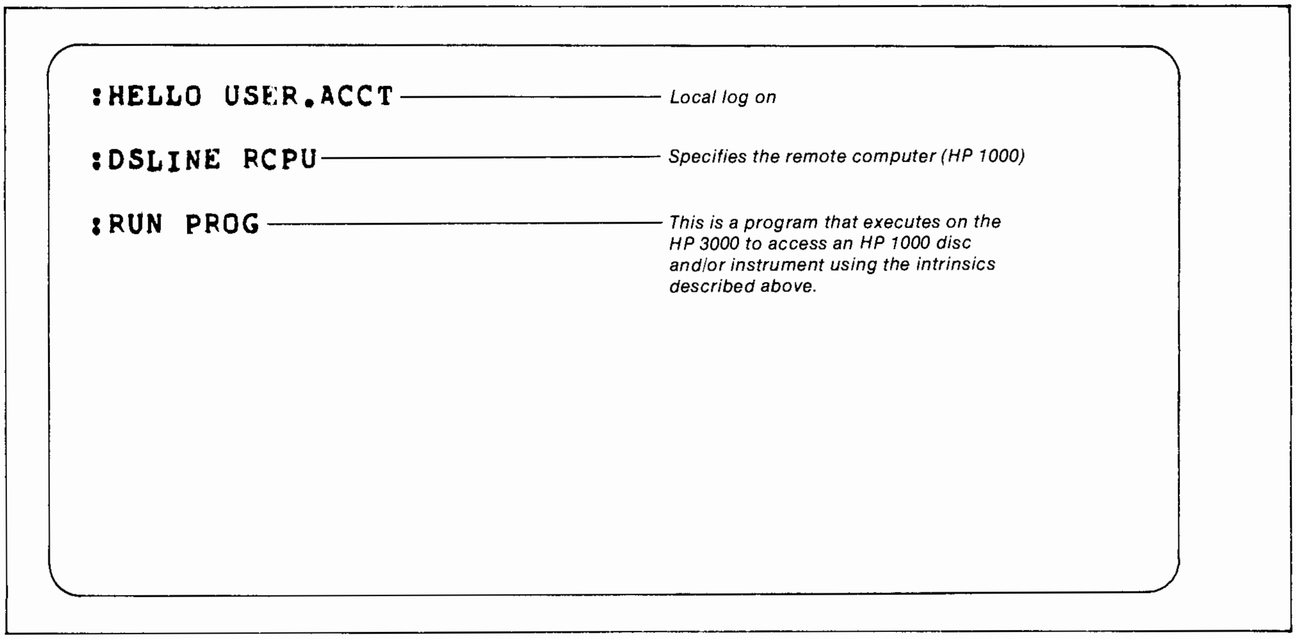
Accessing remote peripheral devices can be exactly the same as accessing remote files. The methods which exist for accessing remote files: HP 3000 utility programs and the standard language input/output statements, are available.

In addition, HP 1000 peripheral devices, such as line printers, magnetic tape units, and scientific instrumentation, can be accessed by an HP 3000 program using remote Distributed Executive (DEXEC) calls.

These DEXEC calls may also be used to schedule or terminate programs, request system time, or inquire about the status of a program or an I/O device. These intrinsics have a direct correspondence to standard RTE EXEC calls.

Remote Data Base Access

The remote data base access feature of DS/3000 gives you the capability of direct and indirect access of data bases on remote computer systems. Using the QUERY and IMAGE data base inquiry facilities of the remote system you can locate, report, and update data values directly in remote IMAGE/3000 data bases and indirectly in remote IMAGE/1000 data bases.



```
:HELLO USER.ACCT

:DSLINE RCPU

:RUN PROG
```

Local log on

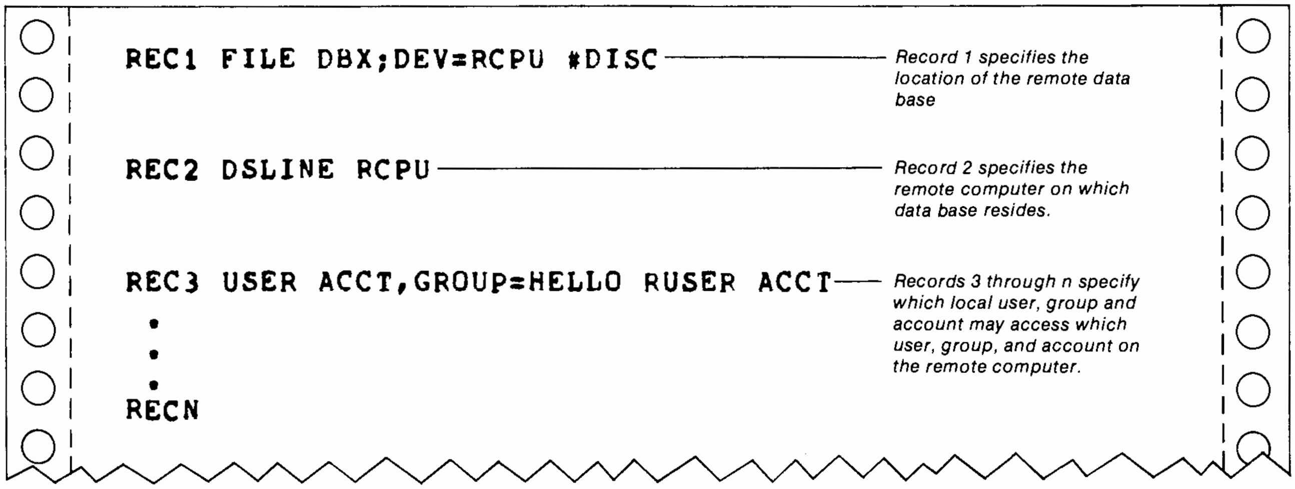
Specifies the remote computer (HP 1000)

This is a program that executes on the HP 3000 to access an HP 1000 disc and/or instrument using the intrinsics described above.

Remote Data Base Access (RDBA): The remote data base access feature of DS/3000 gives you the capability of direct and indirect access of data bases on remote computer systems. Using the IMAGE and QUERY data base inquiry facilities of the remote system you can locate, report, and update data values directly in remote IMAGE/3000 data bases, a remote program is not required.

Direct data base access is a feature of the HP 3000 to HP 3000 communications link only. Using this method, you can automatically interrogate data bases on remote HP 3000 computer systems with remote QUERY interactive commands. Your local application programs can also contain standard IMAGE calls which retrieve and manipulate information in the remote data base. Intervention from a remote program is not required.

There are three methods by which you can directly access a remote data base. The first method requires you to establish a communications line and a remote session and enter a FILE equation for each remote data base. The FILE equation specifies which data base is to be accessed on which remote system and device. A local IMAGE application program can then be run to access the remote data base. The second method allows you to embed MPE command calls within your program, removing the need for the user to know that the data base is remote. The third method requires that a file called the data base access file (DBA file) be created. This file provides IMAGE with the necessary information to establish a communications link and a remote session. It also specifies the remote data base file name so that the necessary IMAGE intrinsics can be executed on the remote computer. The following example illustrates this method. The DBA file is built by the HP 3000 EDITOR. It is named DBAX and contains:



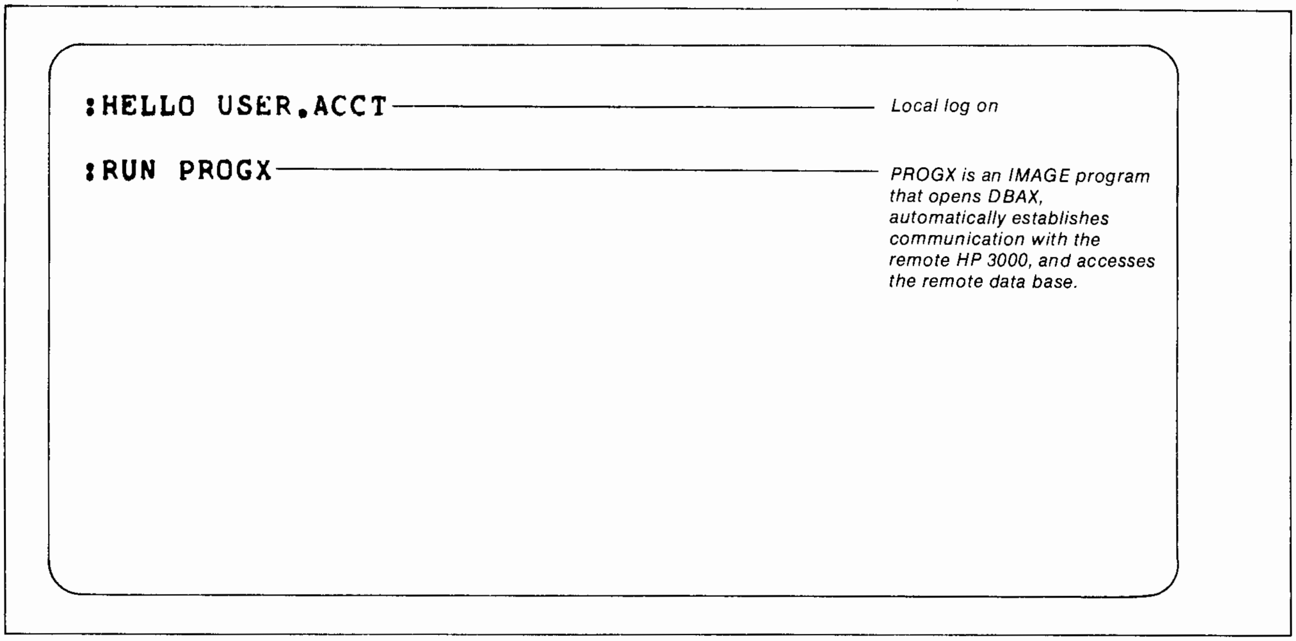
```
REC1 FILE DBX;DEV=RCPU #DISC

REC2 DSLINE RCPU

REC3 USER ACCT,GROUP=HELLO RUSER ACCT
 •
 •
 •
RECN
```

Record 1 specifies the location of the remote data base

Record 2 specifies the remote computer on which data base resides.

Records 3 through n specify which local user, group and account may access which user, group, and account on the remote computer.

As a local HP 3000 user, you simply type the following to access the remote data base DBX:



```
:HELLO USER.ACCT
:RUN PROGX
```

Local log on

PROGX is an IMAGE program that opens DBAX, automatically establishes communication with the remote HP 3000, and accesses the remote data base.

Indirect data base access is available on both the 3000/3000 and the 3000/1000 communications links, using the program-to-program capability. By first initiating a program on your local system, you can then run a remote program which accesses and/or updates the remote data base. The remote program then passes the requested information back to your local program.

Program-to-Program Communications (PTOPC): The DS/3000 program-to-program communications capability gives you the ability to write application programs using a set of nine intrinsics. The two programs can be in different languages. These intrinsics make it possible for two or more user programs residing on different computer systems to exchange data and control information directly and efficiently with one another. The intrinsics, called PTOPC intrinsics, are directly callable by SPL, FORTRAN, BASIC, and COBOL programs. The nature of any two programs communicating with one another in this manner is not symmetrical. One of them (referred to as the master program) is always in control and is the one that initiates all activity between the two programs. The other (referred to as the slave program) always responds to requests received from the master (see Figure 4-7). The master program opens the data link, initiates the slave program, and is always in control. The slave program merely responds to requests received from the master program, either to accept or reject the master program requests. Each computer may have master and slave programs active simultaneously, depending on the needs of your specific application.

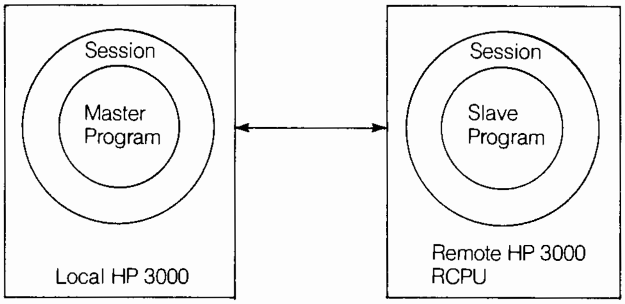


Figure 4-7. Program-to-program communication allows user programs on different systems to execute and exchange data in a coordinated manner. One of the programs is the master and is always in control. The slave program responds to the master's requests.

HP 3000 COMMUNICATIONS WITH IBM OR IBM-COMPATIBLE SYSTEMS

Distributed networks of HP computer systems or standalone HP computers can communicate with IBM mainframe ("host") computers or other computer systems that are communications-compatible with IBM mainframes. Two software products for the HP 3000 provide batch communication with these computers:

- Remote Job Entry/3000
  (2780/3780 Emulator)
- Multileaving Remote Job Entry/3000
  (HASP II/JES/ASP Workstation Emulator)

With Remote Job Entry/3000 (RJE/3000) software, the HP 3000 can submit and receive jobs from the mainframe as an IBM 2780 or 3780 batch terminal. RJE/3000 may also be used to communicate (pass and receive files) with many non-IBM systems that support or emulate IBM 2780 or 3780 devices. For IBM mainframes using HASP II, JES 2, JES 3, or ASP job entry systems, Multileaving Remote Job Entry/3000 (MRJE/3000) software may be used to to transmit and receive multiple jobs from the host at the same time.

Both software products run under the control of the HP 3000 MPE operating system, and may be used concurrently with other HP 3000 processing.

HP-IBM COMMUNICATIONS: A MANAGEMENT PERSPECTIVE

Broad Range of Applications

Hewlett-Packard's IBM-Communications software span a broad range of applications in a distributed processing environment. For applications that warrant high volume, low frequency (a few jobs daily or weekly) transmissions of data from the HP 3000 to the mainframe, RJE/3000 and MRJE/3000 are ideally suited. Typical examples are payroll processing and quarterly reporting. RJE/3000 may also be used for batch input or output to an interactive application on the mainframe, making it suitable for applications of intermediate frequency (up to several transmissions an hour) as well. Both products—RJE and MRJE—can be used concurrently on the same HP 3000.

Friendliness and Ease of Use

Hewlett-Packard's HP-IBM communications software is designed to minimize required user training and familiarity. MRJE commands, for example, are English words such as HOST, SUBMIT, and EXIT that are easy to remember.

Central Control—If Desired

With the HP 3000's remote console capability, HP-IBM communications can be managed from the mainframe site, if desired. A single HP 3000 system console at the mainframe site can dial-up different remote HP 3000's to transmit batch jobs through MRJE/3000 or RJE/3000, providing control from a central site. Both RJE/3000 and MRJE/3000 support auto-answer modems.

Efficient Use of System Resources

No special peripherals are required for input and output with HP-IBM communications products. RJE/3000 and MRJE/3000 allow jobs to be submitted and output to be received through any ordinary input or output device or file on the HP 3000. Any HP terminal—not just the system console—can be used by the MRJE manager to control workstation activity. This means better utilization of terminals, and in many cases alleviates time-consuming movement between different terminals.

Remote Job Entry/3000 (HP 2780/3780 Emulator)

RJE/3000 makes the HP 3000 appear to the host system as either an IBM 2780 or 3780 data transmission terminal, and thus allows batch jobs for the host system to be submitted and received from the HP 3000. RJE/3000 provides greater flexibility than the IBM data communication terminals it emulates, in that jobs may be submitted from the HP 3000 via any ordinary input device (disc file, magnetic tape, card reader, or terminal) and output received to the HP 3000 via any output device (printer, disc file, tape, card punch, or terminal). RJE/3000 jobs can be transmitted to the host, and job input and output devices on the HP 3000 specified, by either an interactive session from an HP terminal or a batch job running on the HP 3000.

RJE/3000—Where To Use It

RJE/3000 can be used for remote job entry to any host computer which supports the IBM 2780 or 3780 communication terminals. In addition to use with mainframe job entry subsystems, RJE/3000 may be used with mainframe communication monitors such as CICS for batch input/output to interactive applications. It can also be used to communicate (transmit and receive files) with the IBM 2780 or 3780 themselves, or other devices or systems that emulate them. RJE/3000 is often used as a means for communicating with other vendors' minicomputers.

Features

- Any input/output device on the HP 3000 may be used to submit or receive jobs (disc file, magnetic tape, card reader, card punch, printer, or terminal)
- Supports lines to multiple hosts and/or multiple lines to a single host
- Can be used with either switched (dial-up) or leased lines; Modem communications up to 19.2 Kilobits per second with the Intelligent Network Processor (INP) hardware interface for the HP 3000.
- Jobs may be submitted, and input/output devices specfied, by either an interactive session on an HP terminal, or by an HP 3000 batch job.

In addition to the capabilities of the IBM 2780 or 3780, RJE/3000 provides the following advantages:

- When emulating an IBM 2780, RJE/3000 performs short-record truncation (suppression of trailing blanks) without the user having to supply EM (End of Medium) control characters in the data.
- Unlike an actual IBM 2780 which cannot do character compression, RJE/3000 can compress blank fields when emulating an IBM 2780.
- When emulating an IBM 2780, RJE/3000 can block more than seven records.

As an interface between the HP 3000 and the communication line, RJE/3000 can use either the Synchronous Single Line Controller (SSLC) or the Intelligent Network Processor (INP). One SSLC or INP interface is required per concurrent user of RJE/3000. ASCII and EBCDIC character codes are supported by RJE/3000. IBM 2780 six-bit transcode is not supported.

Multileaving Remote Job Entry/3000

MRJE/3000 gives multiple users on the HP 3000 simultaneous batch access to any remotely connected host computer system using a HASP II, JES 2, JES 3, or ASP job entry system. Job data may be submitted from the HP 3000 via any ordinary input devices (disc files, magnetic tapes, card readers, or terminals) and output received to the HP 3000 via any output devices (printers, disc files, tapes, card punches, or terminals). MRJE/3000 jobs may be submitted, and job input and output devices on the HP 3000 specified, by either an interactive session at an HP 3000, or a batch job running on the HP 3000. In addition to providing for multiple MRJE/3000 users, MRJE/3000 provides for an MRJE Manager who can interactively monitor and control job activity. Host console commands can be entered by the MRJE manager from any HP terminal.

Features

- May be used with HASP II, JES 2, JES 3, or ASP job entry subsystems on the host computer
- Flexible, easy-to-use commands for job submission and status inquiry
- Any input/output devices on the HP 3000 may be used to submit or receive jobs (disc file, magnetic tape, card reader, card punch, printer, or terminal)
- Job submission capability available to multiple users simultaneously
- Supports for concurrent use: an operator console, up to seven logical print streams, seven logical card reader streams, and seven logical punch streams, all interleaved on the same communication line
- Supports multiple hosts and/or multiple lines to a single host
- May use either switched (dial-up) or leased lines; modem communication speeds up to 9600 bits per second
- Jobs may be submitted, and job input/output devices on the HP 3000 specified, by either an interactive session on an HP terminal, or by a batch job running on the HP 3000

Submit Jobs On-line or Off-line

MRJE users may submit MRJE jobs even if no connection exists between the HP 3000 and the mainframe. Jobs submitted off-line are spooled on the HP 3000 and automatically transmitted when the connection is made. Output from the host is then directed to the proper HP 3000 peripheral device or file without further intervention. If no output destination has been indicated, job output is routed to a default device designated by the MRJE manager. MRJE/3000 output may be sent directly to a printer without intermediate spooling, while simultaneously spooling local HP output for the printer, if desired.

MRJE/3000 is designed to be used in a full multiprogramming environment (i.e. it may be used concurrently with any other HP 3000 processing.) However, the HP 3000 with MRJE/3000 is not recommended as a dedicated batch workstation for high-volume remote job entry to the mainframe. Your local Hewlett-Packard Systems Engineer can assist you in evaluating your batch workstation application, in assessing its impact, if any, on local HP 3000 processing, and in planning MRJE activity to maximize overall system performance. A minimum memory size of 512 Kbytes for the HP 3000 is often recommended for MRJE/3000.

The hardware interface between the HP 3000 and the communication line for MRJE/3000 is the Synchronous Single Line Controller (SSLC).

MODEMS

Hewlett-Packard synchronous modems can be used with DS/3000, RJE/3000, MRJE/3000, and MTS/ 3000. They are fully compatible with the HP 3000 and can be used with the Synchronous Single Line Controller (SSLC) or Intelligent Network Processor (INP). A wide variety of options is available, including comprehensive self test facilities.

Two modems are available, one operating at 4800 bps and the other at 9600 bps. The 4800 bps unit will operate on switched or leased lines and can be used for multi-drop configurations. The 9600 bps unit is intended for point-to-point applications using leased lines. The units allow a wide variety of configurations and offer considerable flexibility when data networks are being constructed.

Chapter 5

THE OPERATING SYSTEM–MPE III

EFFICIENT VERSATILITY
A USER-ORIENTED OPERATING SYSTEM
USER INTERFACE
PROGRAM DEVELOPMENT
A DYNAMIC ENVIRONMENT
SYSTEM OPERATION

The functional heart of the HP 3000 is the Multiprogramming Executive, MPE III. This general purpose, disc-based operating system supervises all processing and maintains all user interface with the HP 3000. Two major attributes of MPE are its versatility and ease of use.

Designed to take full advantage of the computer's hardware features such as virtual memory and stack architecture, MPE demonstrates its versatility by enabling the HP 3000 to perform transaction processing, on-line program development, data communications, c.1d batch processing concurrently. In addition, MPE permits system resources to be accessed simultaneously by multiple users, each of whom interfaces with the system independently.

MPE demonstrates its ease of use with its many user assistance features such as a powerful, straight-forward command language and an on-line HELP facility which guides you in using MPE commands. In addition, MPE simplifies the programming task by monitoring and controlling program input, compilation, execution, and output. MPE regulates the order in which programs are executed, and dynamically allocates any hardware and software resources the programs require.

A complete account structure and automatic resource accounting are standard features of MPE. Easy to use MPE commands allow the system manager to set up a hierarchical accounting structure on the system in a style similar to a company organization chart. MPE then automatically keeps track of the system resources used by the various groups in the account structure. This resource usage information can then be used for billing, accounting, or any other application that requires such data.

MPE provides complete security, enabling you to operate in an environment protected from interference or illegal access by other users. This security is accomplished by means of multiple logon passwords, file lockwords, hierarchical access restrictions, and user capability sets.

MPE handles all input/output to peripheral devices, receiving the I/O requests, queueing them if necessary, and performing the actual data transfer. Because MPE treats I/O devices as files, you can write programs without concern for the physical source or destination of the data, and you can run them in either batch or interactive mode without changing the names of the files they reference.

MPE also handles asynchronous terminal communications. Synchronous data communication to terminals, other HP computers, or even non-HP computers is provided by optional data communications subsystems described in Chapter 4.

Figure 5-1 illustrates the major components of the Multiprogramming Executive III Operating System.

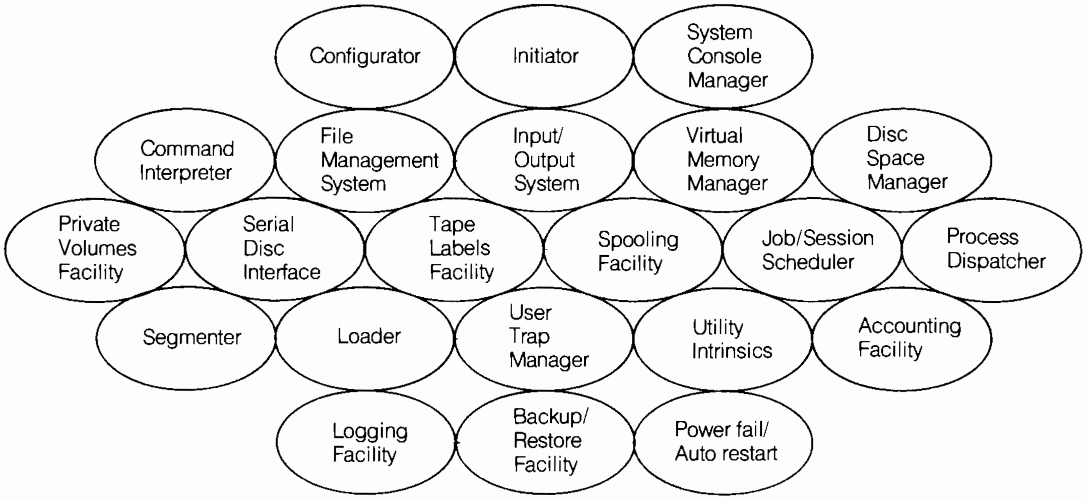


Figure 5-1. Components of the MPE operating system.

EFFICIENT VERSATILITY

All HP 3000 computer systems operate under a single operating system—MPE. This means that programs prepared under MPE III can be run without modification on any other HP 3000 operating under MPE III, provided that all devices required by the programs are connected on-line. It also means that you can move from one HP 3000 to another without undergoing additional training for a new environment.

Multiprogramming

One of the major ways in which operating efficiency is achieved in an HP 3000 is by multiprogramming—the concurrent execution of multiple programs. Multiprogramming allows system resources to be allocated among several competing programs. While one program is awaiting an I/O operation, for instance, control of the central processor is shifted to the next highest priority program waiting for the CPU. MPE is designed to allocate, schedule, and dispatch control of the central processor, storage, and input/output devices among the competing programs. This controlled competition for system resources reduces turnaround time, and significantly increases system throughput. Operating in conjunction with the architecture of the central processor, MPE provides complete protection against one program interfering with another.

The number of programs that can be processed concurrently depends upon such factors as hardware configuration, program operating modes, and the application programs involved. MPE is designed so that the maximum number of concurrently running programs can be increased or decreased by changing a single system configuration parameter.

MPE allows the concurrent execution of programs from two types of input media—traditional batch input devices and interactive terminals. Programs are independent of their input mode and the same system code is used to perform particular functions in either mode. This results in storage economy and reduced overhead.

Interactive Processing

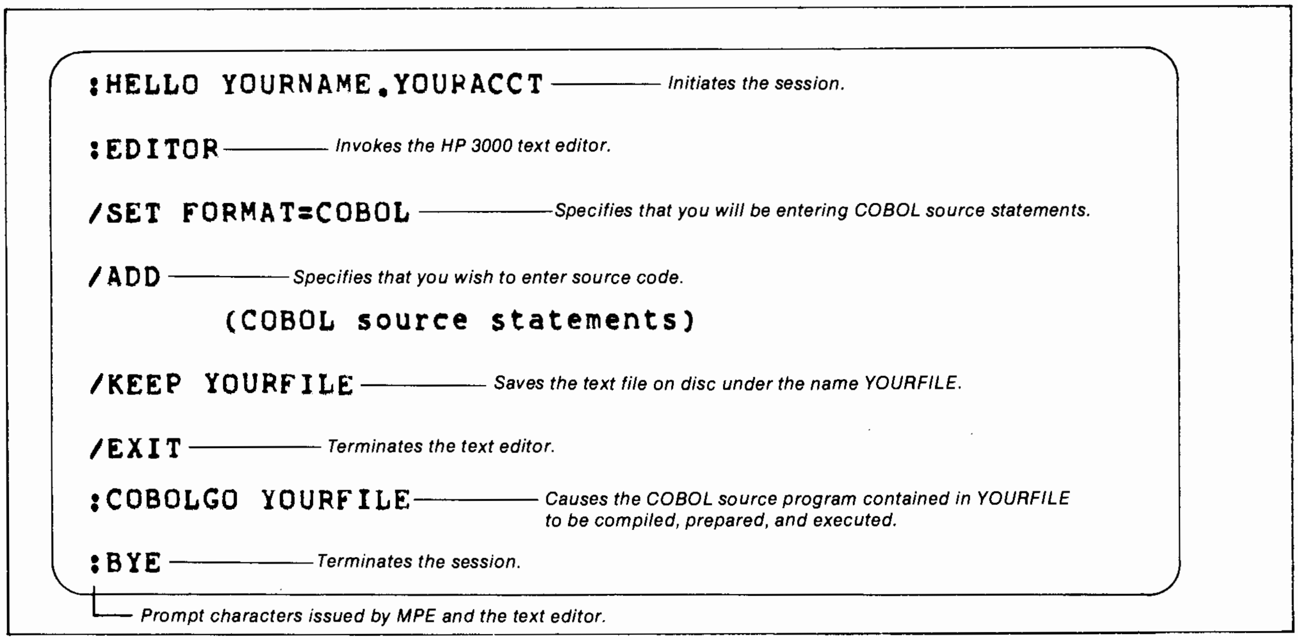
When using the interactive processing mode, you enter commands and data through a keyboard terminal and receive immediate responses to your input. This type of interaction is called a session and is especially useful for program development, text editing, data entry, information retrieval, computer-assisted instruction, and other applications where a direct dialogue with the system is preferred. Sessions can be used to access:

- Operating system commands and subsystems
- Language and utility programs
- Data base management programs
- Data communications programs
- Application programs

A session begins when you enter the :HELLO command from an on-line terminal and MPE connects you to the command interpreter. You may then enter commands to use language compilers or other subsystems such as the text editor, to run programs, or to modify your files. The session continues until you enter a :BYE command, a new :HELLO command, or the system operator forcibly aborts the session.

As an example, let's assume you want to create a COBOL program and then compile, prepare, and run it during an interactive session. Figure 5-2 shows the various commands entered during such a session. You initiate the session by pressing the RETURN key on a terminal that is connected on-line to the system. MPE responds by displaying a colon prompt character. You log on to MPE by entering a HELLO command containing your assigned user and account names, then call the HP 3000 text editor and enter two editor commands followed by the COBOL statements that constitute your program. When the entire source program has been entered, you save it on disc under a file named YOURFILE by entering a KEEP editor command, and then terminate the editor subsystem. Now the source program exists as a disc file in the system. To compile, prepare, and execute the program, you simply enter a COBOLGO command specifying the file YOURFILE. This one command first invokes the COBOL compiler which compiles the program, then invokes the MPE segmenter which prepares the compiled program into an executable form, and finally executes the prepared program. When the program has finished executing, MPE displays the message END OF PROGRAM followed by another colon prompt character. You terminate the session by entering the BYE command.

The example below is somewhat simplified since it does not include the various informational messages, compilation output, and program output generated by MPE, the text editor, the COBOL compiler, and the program itself. The fact remains, however, that if the source program (entered by way of the editor) contains no errors, the entire session can be performed by entering just eight MPE and text editor commands in addition to the COBOL statements which constitute the program.



```
:HELLO YOURNAME.YOURACCT
:EDITOR
/SET FORMAT=COBOL
/ADD
         (COBOL source statements)
/KEEP YOURFILE
/EXIT
:COBOLGO YOURFILE
:BYE
```

Initiates the session.

Invokes the HP 3000 text editor.

Specifies that you will be entering COBOL source statements.

Specifies that you wish to enter source code.

Saves the text file on disc under the name YOURFILE.

Terminates the text editor.

Causes the COBOL source program contained in YOURFILE to be compiled, prepared, and executed.

Terminates the session.

Prompt characters issued by MPE and the text editor.

Figure 5-2. Sample session

Batch Processing

Batch processing is a logical extension of the interactive functions available through MPE. Any capability, with the exception of BREAK, that is available in one mode, is available in the other and requires the same MPE commands. Languages, utilities, and applications development software can be run in either batch or interactive mode without changes. The standard input and output devices are automatically redefined.

The batch processing mode lets you submit to the computer, as a single unit, commands that request various MPE operations such as program compilation and execution, file manipulation, or utility functions. Such a unit is called a job. Jobs contain all necessary instructions to MPE and all references to programs and data required for their execution. Once a job is running, you need to supply no further information.

Jobs are often read through batch input devices such as card readers or tape units. A unique feature of MPE, however, allows you to enter batch job streams through the terminal during the course of an interactive session.

Several jobs can be submitted to the system from multiple devices concurrently. Batch job input is spooled on disc and MPE schedules each job for execution according to its job input priority specified in the JOB command. Additional commands are provided for monitoring job selection. The system operator specifies the maximum number of jobs that can be executed concurrently and can dynamically adjust the job selection criteria.

When a job enters execution, the commands within it are executed sequentially on a multiprogramming basis. MPE generates the job output on a local device such as a line printer, tape unit, or disc unit, or on a local or remote terminal. When one job is temporarily suspended, perhaps to await input of data, another job or session (if available) immediately enters execution. Spooled output on disc is selected for output processing according to the output priority specified in the JOB command.

MPE executes many sessions and batch jobs simultaneously. The only significant difference between a session and a batch job is that during a session you can interactively alter the course of processing, whereas in a job the the command stream is fixed and the job will be executed in its entirety, as pre-defined in the job control statements, without active intervention.

A USER-ORIENTED OPERATING SYSTEM

The many features and capabilities of the Multiprogramming Executive operating system are designed around the concept of the user. There are two distinct types of users:

- the end user, such as a data entry clerk, whose only concern is running a compiled application program, and
- system administrators, such as application programmers, who are responsible for the creation and maintenance of application programs as well as the day-to-day operation of the system.

Each type of user is associated with a particular set of capabilities and responsibilities, and each has access to MPE features which assist him with his specific tasks.

User Classification

Programmers are users who create application programs which run on the system. MPE provides two major areas of system interface for these users: an interactive interface which includes a command language, an on-line HELP facility, and job control facilities; and a programmatic interface which includes programming intrinsics and the MPE file system.

The end user, who can range from an order entry clerk to a functional manager, takes advantage of all the capabilities of the operating system through an application program which he can run without any knowledge of MPE itself.

System administration is performed by a hierarchy of users whose overall responsibility is the successful administration of the computer system. The various levels of MPE administration are defined in Table 5-1. In a small installation, a single user may perform the functions associated with all levels of administration. In a larger installation, the capabilities may be divided among several individuals at each administrative level. Thus it is more appropriate to think of "a user with system manager capabilities" rather than a formally titled "system manager."

TABLE 5.1 SYSTEM ADMINISTRATORS

System Manager:
Manages the overall system by creating accounts (basic structures for user access) and defining resource-use limits.

System Supervisor:
Manages the system on a day-to-day basis, controls scheduling queues, alters system configuration, and maintains the system library.

Account Managers:
Maintain accounts by defining the valid users and file groups for the accounts and specifying resource-use limits for them.

The system operator is the user who operates the system console and is responsible for responding to all system requests. MPE provides a range of operational capabilities which augment the performance of day-to-day operations such as system start-up, back-up, maintenance and recovery, as well as helping the system operator keep the system operating as smoothly and efficiently as possible.

User Capabilities

Capability sets are assigned to each system user based on the kinds of tasks each user needs to perform. These capability sets are divided into three categories:

- User attributes
- File access attributes
- Capability-class attributes

User attributes include system manager, system supervisor, and account manager capabilities. Additional capabilities such as account librarian with special file access capabilities for maintenance of account files, group librarian with special file access capabilities for maintenance of group files, and diagnostician with the ability to run diagnostic programs under MPE for on-line checkout of HP 3000 hardware components may also be assigned.

There are two file access attributes: *save,* which permits the user to save files by declaring them permanent, and *non-shareable devices,* which allows a user to use non-shareable devices, such as a magnetic tape unit.

Capability-class attributes refer to the ability to access special MPE facilities. Included here are interactive access, local batch access, process handling, extra data segment acquisition, multiple resource identification numbers, privileged mode, user logging, private volumes, and data communications.

The majority of users in a typical HP 3000 installation will simply have capabilities for interactive access and local batch access. MPE simplifies the assignment of user capabilities by establishing a set of default capabilities. If a user needs additional capabilities later, these can easily be added.

The presence of capability sets greatly simplifies the use of the system from the standpoint of each individual user by defining the extent to which he must understand MPE and permitting him to ignore those aspects of the system that do not apply to him.

USER INTERFACE

Three user-oriented software facilities provide a comprehensive interface between the system/application programmer and MPE. These tools are: the MPE command language, the on-line HELP facility, and job control facilities.

Command Language

The simplicity of the MPE command language greatly enhances the system's usability. MPE commands enable you to initiate a session and specify the various MPE operations you wish to have performed. When you specify a command, a portion of MPE called the command interpreter reads the command, checks its validity, and then causes the appropriate action to be taken. After the requested action is successfully completed, the interpreter processes your next command. You may enter commands interactively during a session or through a batch input device. Commands may also be issued programmatically from a running program.

MPE uses a colon (:) to prompt you for a command during an interactive session. When a batch job is submitted, MPE commands within the job are designated by a colon in column one.

The MPE command language is composed of many commands. Each command enables you to request a specific action of MPE. Collectively they provide a powerful system-usage tool. The full range of MPE commands is presented in Appendix B. The following is a list of common command uses:

- Initiating and terminating jobs and sessions.
- Running system programs or compilers.
- Running programs.
- Running system utilities.
- Creating, managing, or deleting files
- Displaying file information.
- Displaying job, session, or device status.
- Transmitting messages.
- Assisting in program debugging.
- Establishing communication between a local and a remote computer.

If the command interpreter detects an error during a session, MPE informs you with an error message which specifies the erroneous parameter. MPE then requests that a new command be entered. You can instantly correct command errors by retyping the command or using the REDO command, which allows you to edit the erroneous command, and the session can continue.

During a batch job, MPE lists the error on the listing device. Input from that point through the next EOJ, DATA, or JOB command are usually ignored. You can, however, use the CONTINUE command to request that the job be be continued despite the error.

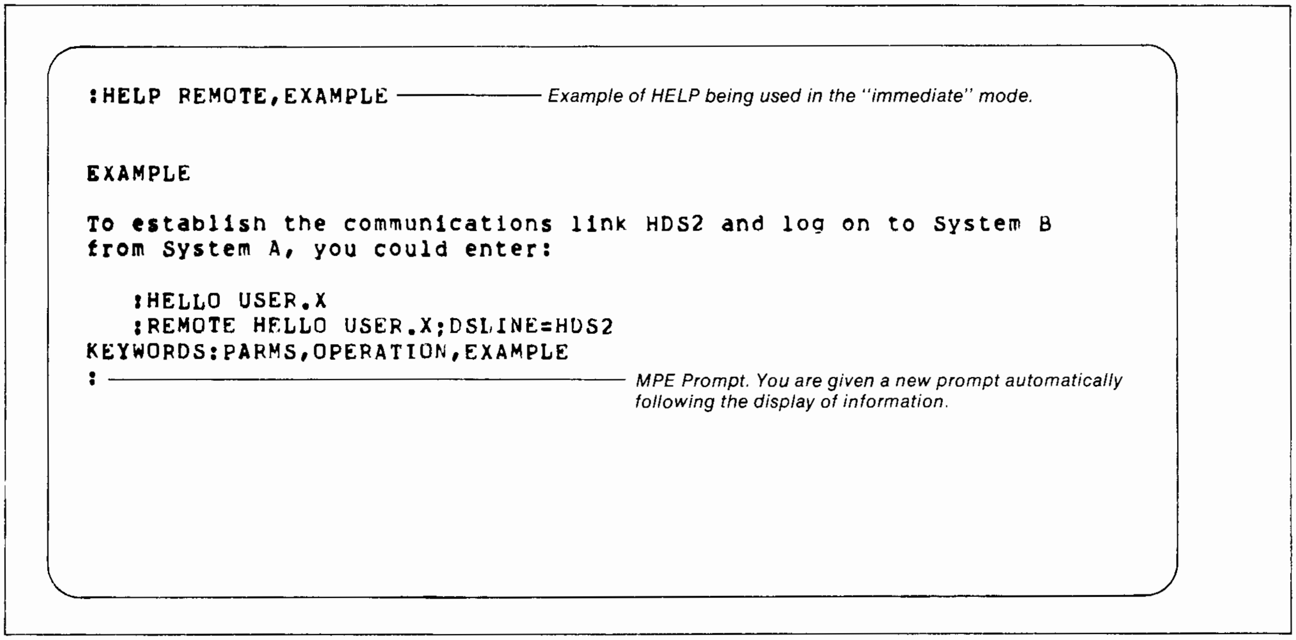
User-Defined Commands

MPE allows you to define your own commands by combining several MPE commands into a command procedure and assigning the procedure a name. The name can then be used as a command. Thus it is possible to enter a single command name which you have defined and cause several commands to be executed. These user-defined command sets can be created by each individual user as well as being made available to entire accounts and all accounts systemwide. It is also possible to redefine existing MPE commands and messages to suit your particular situation.

On-line HELP Facility

Whenever you need assistance with command language syntax or even the name of a particular command, you can invoke the on-line HELP facility. This facility provides graduated information on any MPE command or set of commands. The HELP messages displayed coincide with the information contained in *HP3000 Command Reference Manual.*

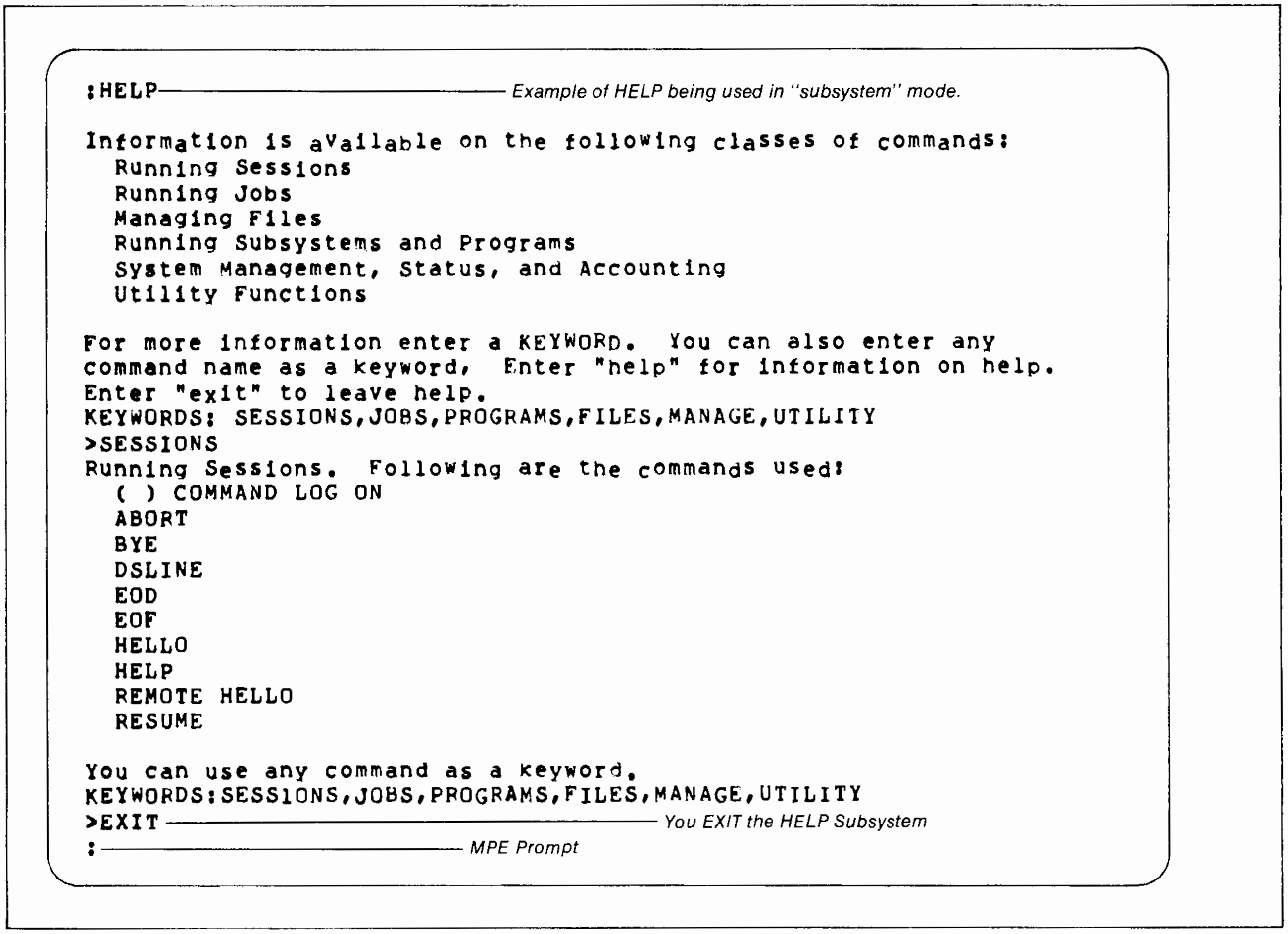
Figure 5-3 demonstrates the two ways in which the HELP facility can be used. In the "immediate" mode, you merely enter the HELP command followed by a parameter. Information detailing that parameter is displayed immediately. In the "subsystem" mode you enter the HELP command without any parameters. The system then displays a menu of valid parameters, and prompts you with a greater than sign (>) for the parameter you wish explained.



```
:HELP REMOTE,EXAMPLE


EXAMPLE

To establish the communications link HDS2 and log on to System B
from System A, you could enter:

   :HELLO USER.X
   :REMOTE HELLO USER.X;DSLINE=HDS2
KEYWORDS:PARMS,OPERATION,EXAMPLE
:
```

Example of HELP being used in the "immediate" mode.

MPE Prompt. You are given a new prompt automatically following the display of information.



```
:HELP

Information is available on the following classes of commands:
   Running Sessions
   Running Jobs
   Managing Files
   Running Subsystems and Programs
   System Management, Status, and Accounting
   Utility Functions

For more information enter a KEYWORD.  You can also enter any
command name as a keyword.  Enter "help" for information on help.
Enter "exit" to leave help.
KEYWORDS: SESSIONS,JOBS,PROGRAMS,FILES,MANAGE,UTILITY
>SESSIONS
Running Sessions.  Following are the commands used:
   ( ) COMMAND LOG ON
   ABORT
   BYE
   DSLINE
   EOD
   EOF
   HELLO
   HELP
   REMOTE HELLO
   RESUME

You can use any command as a keyword.
KEYWORDS:SESSIONS,JOBS,PROGRAMS,FILES,MANAGE,UTILITY
>EXIT
:
```

Example of HELP being used in "subsystem" mode.

You EXIT the HELP Subsystem

MPE Prompt

Figure 5-3. Using the on-line HELP facility.

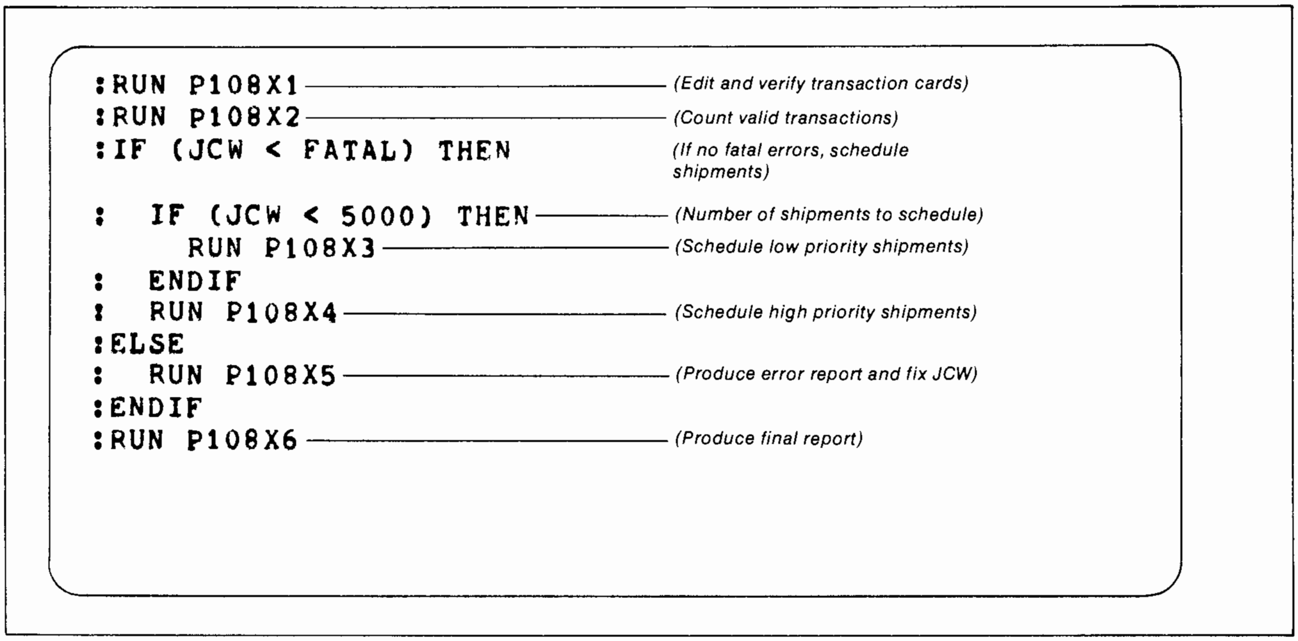
Job Control Facilities

MPE contains job control words (JCW) and conditional execution functions which permit the user to design job streams whose execution can be dynamically altered based on the results of previous job steps.

You can use both system defined and your own job control words to store job status information and to pass such information between programs and between a program and the MPE command interpreter. JCWs are defined and accessed by commands from the command interpreter and by intrinsics from your programs.

You can also use JCWs in conjunction with conditional execution function statements. These statements specify a logical expression (TRUE or FALSE), and are evaluated during program execution. If the value found is TRUE, the remaining statements related to that condition are executed. If the value is FALSE, any existing alternative statements are executed instead.

The following example illustrates the use of JCWs and a conditional execution function. The sample job runs a program which edits and verifies transaction cards and counts valid transactions. If no fatal errors are encountered, the job schedules shipments (either all shipments or only high priority shipments depending on the value of JCW) and produces a final report. If fatal errors do occur, the job does no shipment scheduling. Instead, it produces only an error report and a final report.



```
:RUN P108X1 ──────────────────── (Edit and verify transaction cards)
:RUN P108X2 ──────────────────── (Count valid transactions)
:IF (JCW < FATAL) THEN           (If no fatal errors, schedule
                                 shipments)
:   IF (JCW < 5000) THEN ─────── (Number of shipments to schedule)
      RUN P108X3 ─────────────── (Schedule low priority shipments)
:   ENDIF
!   RUN P108X4 ───────────────── (Schedule high priority shipments)
:ELSE
:   RUN P108X5 ───────────────── (Produce error report and fix JCW)
:ENDIF
:RUN P108X6 ──────────────────── (Produce final report)
```

PROGRAM DEVELOPMENT

The Multiprogramming Executive provides meaningful assistance with the task of generating application programs. MPE programming assistance includes:

- Consistent command language interface to all compilers.
- Program preparation performed by the MPE segmenter.
- Procedure libraries for external references.
- A device-independent file system.
- Flexible file security.
- Subroutines callable across languages.
- Access to all system intrinsics

The use of these software tools during program generation is described below.

Creating Programs

Three steps are required to take a program from source form to an executable state. The first step is to compile the source program into relocatable binary modules called RBMs. This is done by the various MPE language compilers which automatically store the RBMs in a specially formatted file called the user subprogram library or USL.

The second step is to take the USL file and prepare it into a program file. Program preparation resolves external references and results in loadable code segments. This step is done by the MPE segmenter.

The third and final step is to have the MPE loader allocate entries in the HP 3000 code segment table for all the segments in the program file, and to allocate an entry in the HP 3000 data segment table for this process data stack.

Often all of these steps are initiated by single MPE command. (This was illustrated in Figure 5-2.) When necessary, however, you can initiate each step individually, thereby controlling what happens along the way.

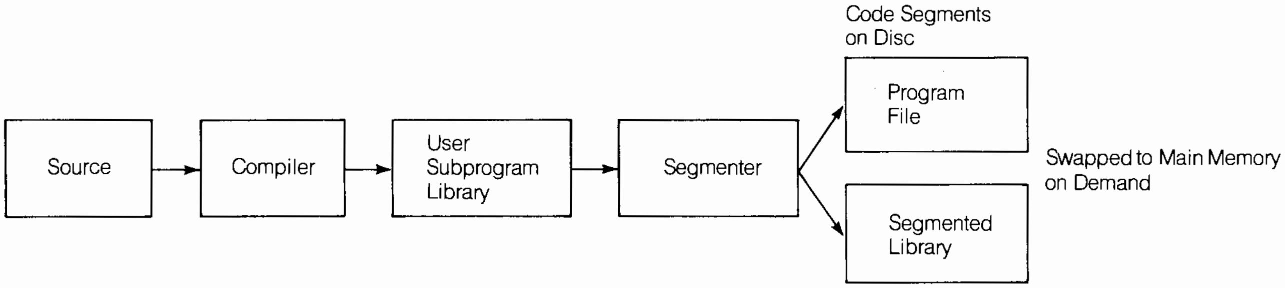


Figure 5-4. Code segment evolution

Accessing Compilers

Program compilation is the first step in converting a source program to an executable state. The format of the commands used to access a language compiler is consistent for all the MPE language compilers. Thus you do not have to learn a new method of program preparation for each programming language you employ.

Three commands are used in the process of program preparation. The first command compiles the program only and stores the resulting RBMs in a USL file for later use. The second command compiles and prepares the program, creating a program file for use in program execution. The third command compiles, prepares and executes the program.

The compiler name is used in the format of the commands. To illustrate, the commands for accessing the COBOL compiler are:

COBOL	which compiles the source program
COBOLPREP	which compiles and prepares the program
COBOLGO	which compiles, prepares, and executes the program

Access to the other compilers is identical, except that the name "SPL," "BASIC" "RPG" and "FORTRAN" replace "COBOL" in each of the three commands. The command BASIC invokes the BASIC interpreter, whereas the command BASICOMP invokes the BASIC compiler. APL invokes the incremental APL compiler

It is important to note that the data files created by these languages are all generated by the same MPE file system. Thus these data files are shareable among the various languages. For instance, a file created by a BASIC program can be read by a FORTRAN program. This file-sharing characteristic also carries down to many HP 3000 subsystems such as KSAM and IMAGE.

Segmenter

Program preparation is actually performed by the MPE segmenter in response to a PREP command or one of the combining forms which include PREP or GO. Segmenter commands may be used to manage USL files by adding, deleting, activating, or deactivating RBMs within them and to build and manage segmented libraries (SLs) which are used to resolve external references from user programs.

Occasionally, you may wish to alter the segmentation of a program to improve its run time efficiency. In many other systems the program would have to be recompiled. With MPE, however, the segmentation can be modified by using the segmenter to rearrange RBMs and then preparing the USL file into a new program file.

Another feature of the segmenter allows different versions of the same subprogram to be stored within a single USL file, and an optional index capability of the segmenter lets you activate and deactivate entry points within various versions of the subprograms.

Procedure Libraries

When a program is allocated and scheduled for execution, MPE searches the following libraries for unresolved external references:

1. The user's log-on group library
2. The public group library of the user's log-on account
3. The public group library of the system account.

Each library can possess two types of library files: segmented library files, and relocatable library files. Segmented library or SL files contain procedures in segmented form which may be shared between programs. Relocatable library or RL files contain procedures in RBM form which must be prepared before they can be loaded with your program.

Procedures contained in the SL file are in prepared form, that is, they are segmented. When a particular procedure is needed, the segment containing the procedure is loaded, as are all external references from that segment. Because the segmentation has been pre-defined in this manner, these procedures may be shared between programs, and only one copy will exist at any time in virtual memory even though several users may require a particular procedure concurrently.

The combination of segmented libraries and relocatable libraries gives great flexibility for storing often used subroutines and procedures. Procedures used system wide are normally stored in the segmented library at the system level. Procedures used by only a few users or a single group are stored in relocatable or segmented libraries at the group, account, or system level.

Special segmenter commands are provided by MPE which enable you to build an SL or RL within a particular group. In addition, commands are available to add or purge routines, and list the procedures contained in either library.

Code Segmentation

To fully appreciate the programming assistance provided by the segmenter, it is necessary to understand the logic behind program code segmentation. In the HP 3000 computer system program code is grouped into topical entities which consist of one or more procedures or subroutines. Each code segment may be up to 32 k bytes in length (where k=1024). Programs may be broken into multiple segments with procedures or subroutines fully contained within one segment.

A code segment consists entirely of information that is not subject to change during program execution. This includes program instructions and constants. No modifiable data may be interspersed with the instructions in a code segment, and it is only possible to change a code segment by recompiling the modified source. This feature ensures that all code is re-entrant, meaning that any sequence of instructions can be in simultaneous execution by multiple users. All HP 3000 computer system procedures are potentially recursive.

The fact that code segments are not modified during execution has specific advantages for the memory management system. Since all code segments are reentrant, all are potentially shared by more than one user when present in main memory. For example, operating system services are provided in part by segments contained in a system segmented library shared by all programs which request those services. Similarly, all users executing the same program file share the program's segments. Only one copy of a segment needs to be in main memory, no matter how many users may be executing it concurrently. Thus the system is able to handle more users in a given memory capacity.

Another advantage of code segment re-entrancy is that code segments are read-only. Thus they never need to be swapped from main memory back to disc, even when overlayed, because there is always an identical copy of the segment in the program file or library. Code is swapped only into main memory, never out. The resulting reduction in swap traffic leads to more efficient memory management.

File System

One of the main uses of a computer system is information management, i.e., the input, processing, and output of data. MPE manages information by means of its file system. With MPE a file is a body of information or data identified by a user-assigned name. A file may contain commands and/or programs, as well as information, and may be stored on disc, tape, or cards.

MPE also treats peripheral devices as files. Access to such files is device-independent, meaning that a program can read data from a card reader, terminal, magnetic tape, or disc by means of the same request. MPE automatically locates, buffers, transfers, and deblocks the data.

When you ask to read a named disc file, you are only implicitly specifying the disc address of the file; the MPE system determines the explicit address and performs the read. In the same manner, when you ask for a certain type of device by specifying a device class name (disc, line printer, etc.), the file system allocates the actual device for you.

The MPE file system permits sequential access to all files. Disc files with fixed- or user-defined length records may also be accessed randomly. Extensive disc file back-up facilities are provided for all types of disc files. The STORE command copies files to a serial storage device; the RESTORE command restores the files to disc.

Files can be accessed from any programming language by means of standard MPE file system intrinsics. A file can be accessed simultaneously by multiple programs; MPE automatically resolves any contention problems which might occur.

MPE file system commands enable programs to reference files without specifying their actual names, addresses, or characteristics. A file can be redefined without a major change to the program. For example, a program's input file named CARDIN designated as a card reader could be changed to a disc file through the use of a FILE command. File specifications can also be altered at run time by means of commands. To illustrate, a program could be coded to open a file with a record length of 128 characters. If at run time it is determined that the file has only 64 characters, you can override the file opening with a FILE command that designates the file to be 64 characters. Reprogramming is not required.

MPE file system commands can be used to build, purge, rename, and display file characteristics. You may specify how disc space is to be dynamically acquired, whether to deal with logical or physical records, whether to include special characters and so forth.

The user logging facility of MPE allows users and subsystems to record additions and modifications to files. In the event of a loss of a file, the user logging record can be used to recover the data in the file. In addition, the logging file can be used as a record of the activity in that file. The entire user logging facility is implemented through the use of MPE commands and intrinsics designed for this purpose.

The MPE file system is actually a collection of routines which reside in the system segmented library (SL). These routines enable you to open a file, obtain status information, read or write data, perform control functions, and close the file. When a program contains statements or constructions that input or output data, these procedures are brought into play automatically by MPE. The loading operations done by MPE to run your program search the library and establish linkages to allow these routines to be referenced during program execution. The code segments containing these file system procedures are shareable, as are all code segments under MPE, and may be used by several programs at the same time.

Subroutine Compatibility

The MPE file system allows programs written in one language to call subroutines written in another language. Once the subroutine is written in the chosen language, it is stored in the system SL (segmented library). Then, user programs written in FORTRAN, BASIC, or SPL can access this subroutine with a standard subroutine call. COBOL and RPG programs can also execute these SL subroutines via an SPL subroutine call.

This ability to intermingle different programming languages significantly expands programmer productivity and application efficiency. Your programmers are free to program in the languages that they are most familiar with. The different language blocks can then be linked by a master program for execution. This way, segments of programs can be written in the language that is most efficient for the operation being performed. This MPE ability allows you to expand the efficiency and capabilities of your application programs.

File Security

MPE provides two general methods of file security. The first is the use of passwords. The creator of a file can establish passwords (also referred to as lockwords) which must be correctly supplied when anyone makes reference to that file. The second method of file security is the use of file access mode and user type restrictions as outlined in Table 5-2.

The system manager specifies the file access modes allowed for an account and the types of users to whom they are available. The account manager specifies the access modes allowed for a group and the types of users to whom they are available. Finally, the creator of a file specifies the file access modes allowed for the file and the types of users to whom the file is available.

In this manner, access to files can be controlled at several levels which range from unrestricted access making the file available to anyone, to controlled access making the file available to its creator only. For example, you can make your data file available to any other user in a "read-only" mode, while only members of your account can append data to the file.

Often a need exists to save general purpose utility programs in public groups or accounts which may be accessed by all system or account users. MPE provides a special system account named "SYS" and a public group named "PUB" which exists under any account with less-restrictive default security provisions.

Intrinsics

A multitude of additional system functions are available in the form of special MPE procedures, called intrinsics, which may be invoked by calls from your program. Intrinsic calls are acted upon when the segmenter prepares the program containing the intrinsic calls for execution. The segmenter establishes a link between the executing program and the MPE procedure specified by the intrinsic call.

System intrinsics are written in the HP 3000 Systems Programming Language (SPL) and follow the rules and constraints of that language. They may be called from BASIC, FORTRAN, or SPL programs, and from COBOL and RPG programs by way of an SPL subroutine.

There are MPE intrinsics for:

- Opening and closing files
- Reading from, writing to, and managing files
- Controlling devices (such as rewinding magnetic tapes)
- Obtaining file information
- Obtaining user information
- Obtaining detailed error information
- Performing data translation
- Obtaining date and time
- Process handling
- Resource handling
- Data segment handling
- User logging

ACCESS MODES	
Reading	Allows the user to read files.
Appending	Allows the user to add information and disc extents to existing files.
Writing	Allows the user to delete or change information already present in existing files.
Executing	Allows the user to run programs stored in existing files.
Locking	Provides a logical lock which gives the user exclusive access to a file if desired.
Save Files	Allows the user to save permanent disk files within the user's group.

USER TYPES	
Any User	Makes the specified access modes available to any user of the system.
Account Member	Restricts the specified access modes only to users of the account in which the file resides.
Account Librarian	Restricts the specified access modes only to the account librarian of the account in which the file resides.
Group User	Restricts the specified access modes only to users of the group in which the file resides.
Group Librarian	Restricts the specified access modes only to the group librarian of the group in which the file resides.
Creator	Restricts the specified access modes only to the creator of the file.

TABLE 5-2. MPE FILE ACCESS MODES AND USER TYPES

Appendix C of this manual gives a complete list of all MPE intrinsics with a brief description of each.

When a system intrinsic is invoked to perform a system function, two types of error conditions may occur. MPE informs the calling program of a recoverable error by setting the condition code bits of the HP 3000 status register when the intrinsic is exited. The condition code indicates whether or not the request was granted and what conditions existed pertinent to the request. A request to an intrinsic which requires a special capability class not possessed by the calling program, or which passes illegal parameters to an intrinsic, is considered an irrecoverable error and causes the system to abort the program. In such a case, if you have not specified an appropriate "trap procedure," a batch job is usually removed from the system; an interactive session resumes with a message and a prompt for another command. The CONTINUE command can be used to continue execution of a batch job despite an irrecoverable intrinsic error. Also, by initially calling system trap intrinsics, you may specify special actions to be taken in the event of an irrecoverable error.

A DYNAMIC ENVIRONMENT

The Multiprogramming Executive environment is a dynamic one where programs are run on the basis of processes. A process is the basic executable entity of MPE. It is not a program, but the unique execution of a program by a particular user at a particular time. MPE automatically creates, manages, and deletes processes.

When you execute a program, a private hardware protected data segment called a stack is created for that particular execution. The stack and the program's code segments together constitute the process. To illustrate, when multiple users access the BASIC interpreter, a separate process is created for each of them. They all use the same code, there is only one BASIC interpreter; but each has his own data stack (environment) created by MPE.

The creation, maintenance and deletion of processes is accomplished by means of three MPE components: the virtual memory manager, the job/session scheduler, and the process dispatcher.

Virtual Memory

MPE's virtual memory manager uses both main memory and disc storage to greatly expand the total amount of memory space available. In fact, virtual memory allows programs up to 2 megabytes in length to be executed in minimal memory systems (256Kb). MPE logically divides programs into variable length segments of code and data. These segments reside in disc memory and are brought into main memory only when required for program execution as shown in Figure 5-5. When a code segment is no longer needed, it is overwritten by a new segment. If the code segment is needed again later, it is simply copied again from the disc on which it resides.

Data segments are dynamic and are handled somewhat differently. Since the content of a data segment may change during program execution, a data segment is copied automatically back to the system disc when it is no longer needed, thereby replacing the previous version of that segment.

This approach of segmenting code and data and transfering the segments back and forth between main memory and disc memory results in the allocation of local storage only as needed. In addition, since program code segments are not modified during execution, multiple users are able to share a single copy of the program code.

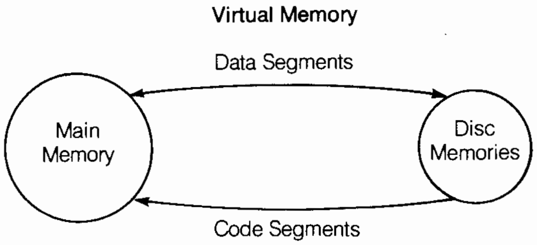


Figure 5-5. MPE virtual memory

A closer look at memory management identifies additional advantages. MPE divides main memory into two areas. The first, fixed memory, contains only those items required to be memory resident such as interrupt handlers, the memory manager, and the scheduler. The remainder of memory, linked memory, contains all other code and data. User and operating system segments are brought into this area by the memory manager as they are required. The architecture allows the operating system, including the file system, the command interpreter, the spooler, and even much of the I/O system, to be shared by all users even though they are brought into main memory only when needed.

The MPE memory manager is responsible for the allocation of main memory to the executing processes. Program and library segments which are needed for execution are automatically swapped into main memory from disc. An attempt by an executing process to access code or data not present in main memory causes the memory manager to allocate main memory space for the missing segment. When the absent segment is swapped from the disc memory to the main memory, the executing process continues. Frequently used segments remain in main memory, and may never be swapped, while rarely used segments are in disc memory most of the time. This results in higher efficiency and faster overall execution time. It also creates a dynamic situation in which segments are being swapped rapidly between main memory and disc memory, according to the demands of the executing programs.

Automatic Scheduling

The MPE job/session scheduler schedules jobs and sessions according to their assigned priorities. When the execution of one process is interrupted for any reason, such as I/O or an internal interrupt, control is passed to the process with the next highest priority which is awaiting CPU resources. When two or more programs have the same priority, the oldest process is selected first.

Jobs and sessions are scheduled by means of a master queue which is ordered by priority as shown in Figure 5-6. This master queue is divided into areas called priority classes. Each area is bounded by two priority numbers established by the system manager.

MPE automatically assigns priority classes to each process executing on the system. The user may, however, specify priority classes by selecting a general category of process dispatching priority for the program. This is done by including the PRI = parameter in your JOB or HELLO command. The five process dispatching priority types (queues) available are:

AS —system processing only

BS —very high priority

CS—interactive

DS—batch

ES —very low priority (background)

MPE actually translates priority types into numerical ranges which are ordered in a master queue (see Figure 5-6). The numerical range of each priority type can be changed at any time to ensure that an optimal balance of services is maintained among the processes on the system.

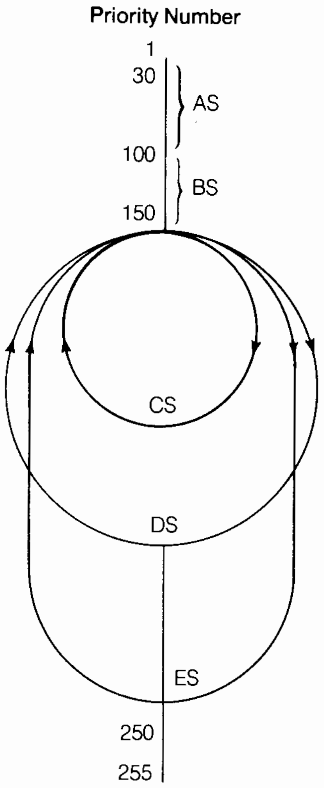


Figure 5-6. MPE master queue structure

Process Execution

The MPE process dispatcher schedules processes for execution. Each process has a dynamically changing priority number, and the dispatcher keeps a list of active processes (those requesting execution) ordered by priority. The dispatcher attempts to execute the highest priority process first. If that process is not in memory, the dispatcher instructs the memory manager to make enough of the process's segments present in memory to allow it to run.

As a process runs it may require another code or data segment. If the segment is not present in main memory, the memory manager is instructed to retrieve the segment before the process is allowed to continue executing. While the process waits for the needed segment to be transferred, MPE transfers control to the next process ready to be executed.

The memory manager determines the working set for each program. The working set is the particular set of segments required in memory for a process to run efficiently. The working set is determined by means of MPE's "segment trap frequency" algorithm, and the size of the working set is expanded or contracted as necessary by MPE. If a working set is too large, the process will control more main memory than it actually needs and thus degrade the performance of other processes. If a working set is too small, the process itself will run inefficiently.

When extra segments are transferred to main memory for a process, the dispatcher determines whether or not to add them to the working set of that process. When segments are removed from the working set, they simply reside on disc.

If there is more main memory available on the system than is required for the working sets of all active processes, code and data segments are left in main memory. If there is insufficient space in main memory for a new segment request from an active process, the memory manager creates space as follows:

- First, available free areas are used.
- Next, segments available for overlay are located and used. Overlaying a data segment involves ensuring that the segment has been copied back to virtual memory.
- The space created by the removal of overlayed segments is coallesced with any other free space available.
- If additional space is needed after all overlayable segments have been used, part of the working set of an active process is temporarily removed by the dispatcher and its space is used for the requested segment. The process with the lowest priority is always selected.

The objective of the process dispatcher and the memory manager is to provide for optimum efficiency in the use of system resources while satisfying the requirements of executing processes. This is done automatically by MPE without assistance from the system users.

Privileged Mode

Privileged mode operation lets you access all MPE resources directly, including privileged intrinsics, system tables, and privileged CPU instructions. In essence, the privileged mode option lets you function as the operating system.

Normally your programs are not able to affect the operation of other users or of MPE itself. Privileged mode operation, however, bypasses normal operating checks and limitations. It is possible for a privileged mode program to destroy file integrity, including that of the MPE operating system itself. Hewlett-Packard can not assume responsibility for system integrity when your programs operate in privileged mode.

SYSTEM OPERATION

This section deals with the day-to-day operational aspects of running an HP 3000 installation. For all of its power and features, MPE is surprisingly easy to maintain. General operation of the system is primarily the responsibility of three users: the system manager, system supervisor, and the console operator. The console operator is responsible for routine day-to-day system operations. It is his task to respond to system messages and to keep the system and the peripheral devices functioning smoothly and efficiently. The system supervisor, who is appointed by the system manager, has day-to-day responsibilities in several areas. Included in his duties are the maintenance of the system logging and resource accounting facilities. He also has the capability to retrieve information and change parameters relating to the master scheduling queue. The system manager implements MPE's unique account/group/user organization and appoints account managers to monitor account usage.

System configuration, the first operation undertaken with a new HP 3000, establishes the boundaries under which MPE will function in terms of memory and peripheral devices. An interactive process that takes place with the operator at the system console, system configuration requires intervention only when options other than the system defaults provided are specified, and can be accomplished in a matter of minutes. Once configured, the system is ready for use.

The next operation involves the building of various accounts and the identification of system users. Usually, the first user identified is the system manager who will have overall control of the system. Among his responsibilities is the task of allocating various accounts within the system and identifying an account manager for each. The account managers in turn identify the users who may access the system through their respective accounts.

Once the system is configured and the account and user structure has been defined, the operation of program development begins. Programming may be done interactively on the HP 3000 by means of terminals and the many programming aids provided by MPE for the development of source programs; or programs can be developed in batch processing mode. Once developed, programs are executed as frequently as required.

Occasionally the account manager may need to add new users to their accounts, or it may be necessary to reconfigure the system as new peripherals or more memory is added. But the primary continuing operations are the development of programs and their execution. Figure 5-7 illustrates the interrelation of these general HP 3000 system operations.

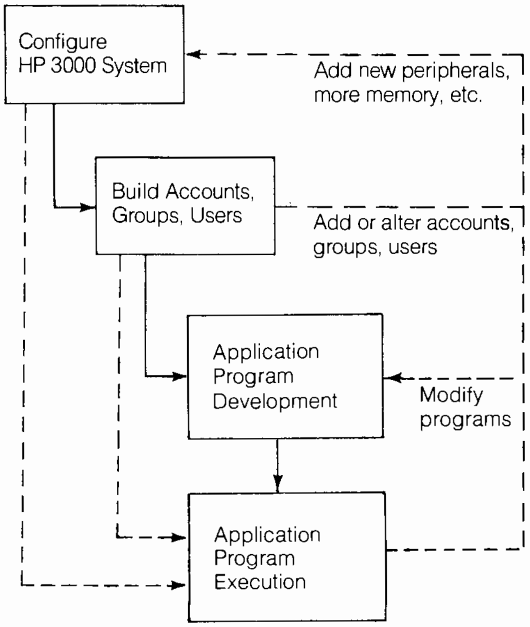

Configure HP 3000 System
Add new peripherals, more memory, etc.
Build Accounts, Groups, Users
Add or alter accounts, groups, users
Application Program Development
Modify programs
Application Program Execution

Figure 5-7. HP 3000 operations

Console Operator Function

The console operator function is available to any one user logged onto a non-REMOTE interactive terminal on the system. Operator commands are entered in SESSION mode just as are MPE commands. The console operator has the ability to move the console to other terminal devices, and may selectively allow users to execute specific operator commands. The operator can also allow individual users to have operator control over specific devices. Operator commands deal primarily with peripheral and other hardware system requirements, as well as with system start-up, back-up and recovery procedures.

Start-up and Modification

MPE is initially brought up on an HP 3000 by the operator at the system console, through one of several restart operations:

- WARMSTART—The system is restarted from the disc. Spoolfiles are recovered and incompletely processed spooled jobs are automatically restarted.
- COOLSTART—The system is restarted from the disc. The spoolfiles in existence when the system was shut down are not saved, but all resident user files are saved.
- COLDSTART—The system is read from a serial storage device (magnetic tape or serial disc). The I/O configuration and system configuration from the serial storage device used to define the system are merged with the directory and files present on the disc.
- UPDATE—Similar to cold start, except that the I/O configuration on the disc is used to define the system. This start-up mode is used when starting the system from an updated version of MPE supplied by Hewlett-Packard.
- RELOAD—The entire system, directory, files, and configuration information are read from a serial storage device.

The restart operation can include an interactive dialogue between the console operator and the MPE initiator program. This optional dialogue permits the operator to change the system configuration. Upon completion, MPE is operational.

System Backup and Recovery

Periodically, the MPE system and user files are copied on a serial storage device for back-up purposes. The serial storage device is then available for reloading the system in the event of a hardware or software failure, or to transfer the system to another hardware installation.

In the latter case, a new I/O and system table configuration may be specified during an interactive dialogue. Areas which may be changed include:

- I/O devices
- System table sizes
- System disc allocation
- Logging facility
- Scheduling changes
- Segment size limits
- System modules to be allocated or made memory resident

The SYSDUMP command (which requires System Supervisor capability) may be used to create media for reloading on a different or changed HP 3000. A new I/O configuration and system table configuration may be specified during the interactive dialogue which occurs. A back-up date may also be specified which enables the operator to back-up only those disc files which change each day.

Power Fail/Auto Restart

When an AC power failure is detected by the HP 3000, a power fail/auto restart routine is automatically invoked. This routine preserves the operating environment prior to complete loss of power. Normal system operation resumes as soon as power is restored. Jobs and sessions in progress on the system continue where they were interrupted, with programs unaware of the interruption (except for magnetic tape units and dial-up terminals in use when the failure occurred, or if the power outage lasts longer than the built-in system memory battery backup).

In addition to the system configuration aids mentioned above, MPE provides a range of software aids specifically designed for use by the system operator. Among these are:

- Spooling facility
- Tape label facility
- Disc options.

Spooling Facility

This MPE facility permits the concurrent usage of devices which would otherwise be non-shareable, such as card readers, magnetic tape drives, or line printers.

This is accomplished by copying the input or output to disc where it is processed later. This procedure is called spooling. (SPOOL is an acronym for Simultaneous Peripheral Operations On-line.)

To illustrate, if six users need to produce output on a line printer at approximately the same time, their output is directed to spoolfiles on disc from which the output is printed on a priority basis as the line printer becomes free. In this way each user can immediately proceed with other processing activities without having to wait for the line printer. Similarly, if there are ten jobs to be read from a card reader or magnetic tape unit, they are all read immediately and are directed to spoolfiles on disc where they wait to be executed. Thus, the card reader or magnetic tape unit is not tied up by one job which must be executed before the others can be read.

The spooling of batch job input can be initiated not only from a card reader or magnetic tape unit, but also from within an interactive session as described earlier.

Tape Labeling Facility

MPE provides a tape labeling facility for use in reading and writing labels on magnetic tapes. The facility can be used to:

- Identify magnetic tape volumes (reels).
- Protect tape volumes from being inadvertently written over.
- Protect private information.
- Facilitate information interchange between computer systems.

The facility can be used to read, but not write, IBM-standard tape labels; read and write ANSI-standard labels; and read and write user-defined labels on previously labeled magnetic tapes.

Disc Options

The MPE operating system includes a serial disc interface which allows non-system domain drives to be used as non-shareable serial devices. To MPE, the discs appear to be magnetic tape drives. They provide a fast system backup and recovery capability when used as an alternative to magnetic tape in backing up the MPE system and storing and restoring files and/or data bases.

MPE also provides a private disc volume facility which allows you to create and access files on removable disc volumes. Private volumes consist of removable disc packs which, when mounted on a disc drive, can be accessed by MPE through the file system. Under private volumes, the disc packs mounted on the drives during a cold load are dynamically allocated to the system domain for normal use or to the non-system domain for private use. Non-system domain packs can be both physically and logically mounted and dismounted during normal system operation.

System Account Structure

The primary responsibility of the system manager is to create and maintain the organizational structure of accounts, groups, and users under which access to MPE occurs. The account structure provides maximum security and control by permitting the system manager to assign specific access and system usage capabilities to each user.

"Accounts" are collections of users and groups. Each account has a unique name and an optional password assigned to it when the system manager creates the account. Each account also has its own file domain or unique set of files. The system manager may define resource-use limits for an account. MPE maintains a running count of each resource that the account uses. MPE also stores a list of user name and group names recognized by the account, the maximum job priority at which jobs in the account may be scheduled, and limits established on the account's usage of disc file space, CPU time, and connect time.

"Groups" are used to partition the file domain of an account. Files must be assigned to a group, and each group has a unique name (within the account) and optional password. Limits may be established on the permanent disc space, CPU time, and connect time used by a group. MPE maintains running counts of resource usage for each group and the sum of these group counts always equals that of the account in total.

"Users" are individuals who access the HP 3000. Each user is assigned a unique name and optional password, and is assigned to a specific account. Each user may have a specified home group of files, and may access any other file groups in the account. A maximum job priority may be assigned to each user.

Each account "owns" a unique set of files separate and distinct from every other account. This ownership is indirect in that only groups may own files directly. Thus, every file belongs to a group, every group belongs to an account, and every account belongs to the system.

To illustrate how accounts, groups, and users interrelate, consider the following example. Figure 5-8 represents a system which includes interactive terminals dispersed throughout a company. The system manager has assigned three accounts: Marketing, Engineering, and Finance. The marketing account manager has defined two users who can access the system: Bill and Dave. Each user has his private group (assigned as his home group) where he stores his private programs and files. Bill and Dave can also access programs and files stored in the other groups in the account. A group named PROJ1 was created to contain programs and data files related to current projects.

An administrative group was also created to contain administrative work such as schedules and budgets. The public group, to which no password was assigned, contains general purpose utility programs for use by all.

Bill can log on to the HP 3000 from a terminal with the command:

```
:HELLO BILL.MKTG
       ↗    ↗
    user  account
    name   name
```

By default, Bill now has access to all programs and data files in his home group: BILLSGRP. Bill can gain access to a file in the PROJ1 group by using the fully qualified file name which specifies both the account and group under which the file was created. The file Bill wants is data file 1, so he enters:

```
FORECAST.PROJ1.MKTG
  ↗       ↗      ↗
 file   group   account
```

Alternatively, Bill could have logged on to the HP 3000 and requested access to all programs and files in the PROJ1 group by appending the group name to his log on request, as follows:

```
:HELLO BILL.MKTG,PROJ1
   ↗       ↗       ↗
 user   account   group
```

Bill now has access to all files in the PROJ1 group.

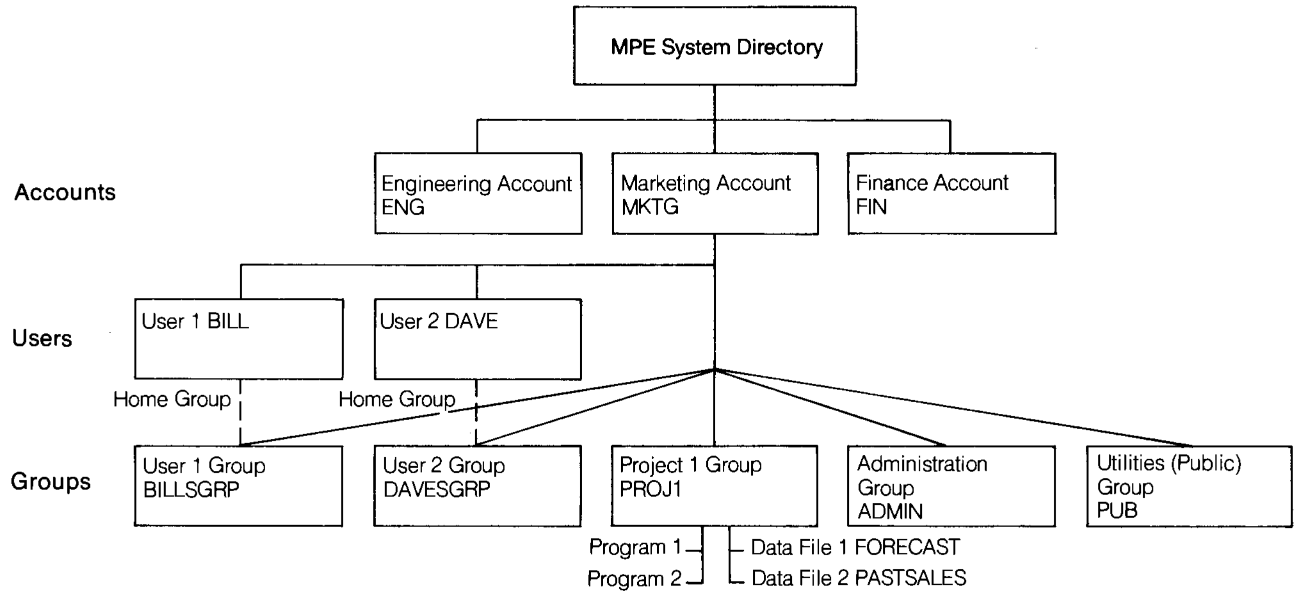
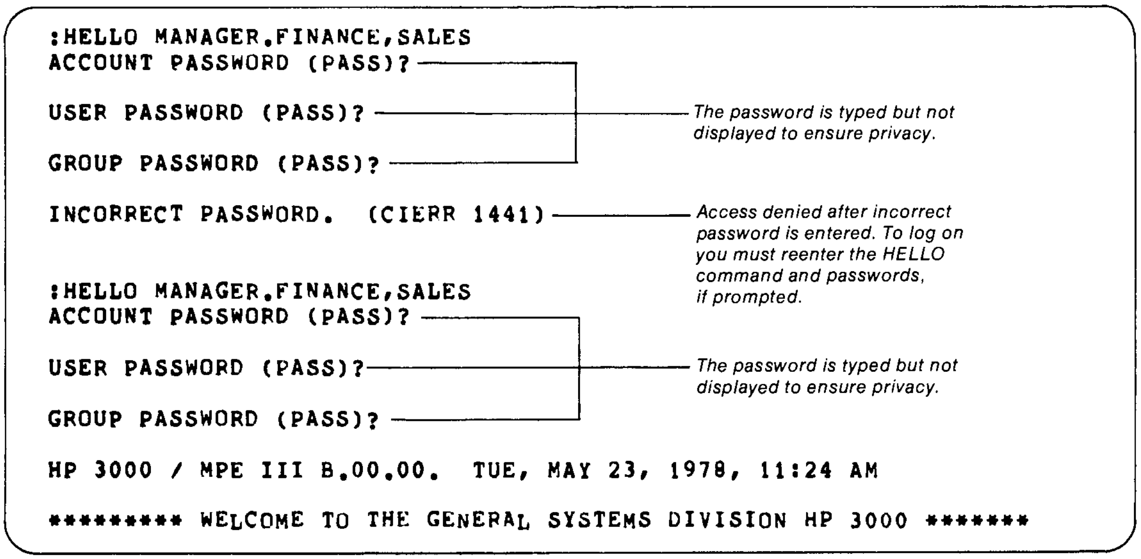


Figure 5-8. MPE account structure

To summarize, you can log on to the system using only your assigned user and account names, in which case you are automatically given access to your home group. Or, you can log on specifying your user name, account name, and a group name which gives you access to the group you specify whether or not it is your home group.

To access a file in a group other than the one specified when you logged on, you use the fully qualified name of the file which can consist of the file name, the group name, the account name, and the appropriate passwords when required.

As you can see, the account structure provides both control and security over file use. Access to the system is granted only to individuals with a valid log-on identification consisting of account, group, and user names, each of which may require a password. Figure 5-9 illustrates both an unsuccessful and successful log-on procedure where passwords are required.



```
:HELLO MANAGER.FINANCE,SALES
ACCOUNT PASSWORD (PASS)?

USER PASSWORD (PASS)?

GROUP PASSWORD (PASS)?

INCORRECT PASSWORD.  (CIERR 1441)


:HELLO MANAGER.FINANCE,SALES
ACCOUNT PASSWORD (PASS)?

USER PASSWORD (PASS)?

GROUP PASSWORD (PASS)?

HP 3000 / MPE III B.00.00.  TUE, MAY 23, 1978, 11:24 AM

********* WELCOME TO THE GENERAL SYSTEMS DIVISION HP 3000 *******
```

The password is typed but not displayed to ensure privacy.

Access denied after incorrect password is entered. To log on you must reenter the HELLO command and passwords, if prompted.

The password is typed but not displayed to ensure privacy.

The user now has access to all HP 3000resources as defined by his user capability set.

Figure 5-9. MPE system access security

System Supervisor

The user with system supervisor capabilities can use commands which enable him to control the master scheduling queue, permanently allocate or deallocate a procedure or program to virtual memory, and manage the logging facility.

He can control the master scheduling queue in two ways: first by entering the SHOWQ command which displays information about the scheduling of processes and the contents of the master queue and various subqueues; and second, by entering the QUANTUM command which allows changes to the time quantum or limits defining the bounds of any subqueue.

Two important commands, ALLOCATE and DEALLOCATE, allow him to tune the system by permanently allocating a procedure or program in the HP 3000 to virtual memory. This significantly decreases the time needed for an operation with large, frequently used routines.

System Logging Facility

The MPE logging facility records details of system resource requests in a series of log files on disc and can be used to monitor system resource usage. The system supervisor selects those system and user events that are to be recorded. Log records are provided for job and session initiation or termination, program termination, file closing, file spooling completion, and system shutdown. Log files can be used in the generation of precise billings based on accurate system usage records.

I/O device failures on any device are also recorded in log files which can then be used to detect device problems before they begin to interfere significantly with overall system operation.

Accounting Facility

The MPE accounting facility provides a flexible and powerful means of coordinating access to the system and disc file usage. To coordinate system access, system administrators can devise a structure of accounts and users which reflects the functional organization of the people who use the system. The accounting facility maintains running totals on the amounts of system resources that each account consumes, including disc space used, cumulative CPU time consumed, and cumulative terminal connect time for sessions. The current totals can be displayed at any time and can be used for billing purposes.

File usage can be coordinated also because the overall permanent disc file domain of the HP 3000 is partitioned among the various accounts. Each account's file domain is further partitioned into groups. If a request to save a file would result in exceeding the permanent file space limit at either the account or group level, the request is denied.

Users may create files only in the account and group under which they are currently running. By using fully-qualified file names users also have access to any file present in the system, provided that existing security provisions allow them access.

Chapter 6

SYSTEM ARCHITECTURE

STACK ARCHITECTURE
SEPARATION OF CODE AND DATA
PROCESSES
VARIABLE-LENGTH SEGMENTATION
CODE SEGMENTATION
DATA STACK
VIRTUAL MEMORY
MICROPROCESSOR
REGISTERS
INSTRUCTIONS
MICROCODE

Stack Architecture

The HP 3000 with its hardware stack implementation is referred to as a stack machine. A stack is a linear storage area for data. It is so named because data items are placed on the "top," "pushing down" the data items already present. Data items are removed from the top, "popping up" those data items remaining. If you have ever worked with a Hewlett-Packard calculator that has an "ENTER" key, then you have worked with a stack. Consider the following calculation:

$$\frac{6 * 10}{2 * (3+12)}$$

Using a stack, no temporary intermediate values need to be named or stored in registers until required later in the computation. An example of how this problem would be evaluated in a stack is presented in Figure 6-1. Note that the top of stack (TOS) moves downward in keeping with the HP 3000 conventional representation. All operations can be performed without naming specific operands, as the top two stack data values are implicitly used.

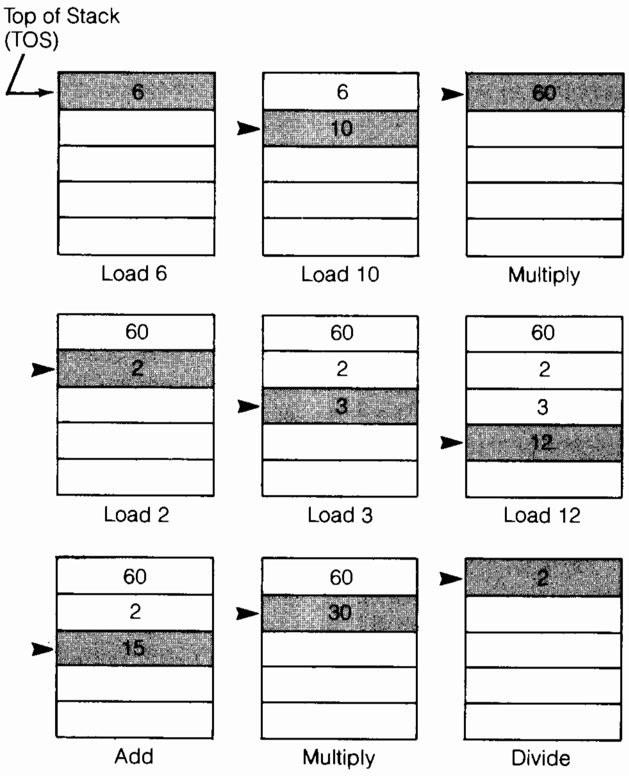


Figure 6-1. Stack Evaluation of (6*10)÷[2*(3+12)]

The benefits of a stack architecture are numerous. First, as seen above, the storage allocation is dynamic. Local storage is allocated only upon entry of a procedure and is automatically freed upon exit so that areas of memory are not tied-up and can be re-used by other parts of the program. Temporary storage of intermediate values is automatically provided. Thus compilers don't have to be concerned with saving and restoring registers for intermediate results.

Code compression is made possible by the omission of operands in many of the instructions on a stack machine. No extra registers are required for subroutine parameters and temporary variables. A register machine requires 50 to 100 percent more bits for code than a stack machine.

The subroutine is one of the most important concepts in software. Modular, structured programming has as its principle idea the partitioning of a large program into many small, understandable modules which can be called as subroutines. The best mechanism for subroutine calls and execution is the stack, because all subroutine return addresses, I/O parameters, and local variables can be pushed onto the stack. This leads to easy parameter passing and provides highly efficient subroutine linkage.

Fast execution on the HP 3000 is a benefit of having several CPU registers to hold the top part of the stack, rather than leaving it all in main memory. The HP 3000 provides each user with hardware protection of his data stack. Rapid interruption and restoration of user environments is made possible by storing the operating environment in a special block as an extension to the user's stack.

Separation of code and data

In most computer systems, programs consist of an inter-mixing of instructions and data. For example, within a subroutine there are program locations reserved by the compiler for return addresses of other subroutines and space set aside for the storage of local variables.

The HP 3000 approach separates a program into those elements that do not need to be altered and those that do. The result is that an HP 3000 program consists of a separate code area and data area (data stack), as illustrated in Figure 6-2.

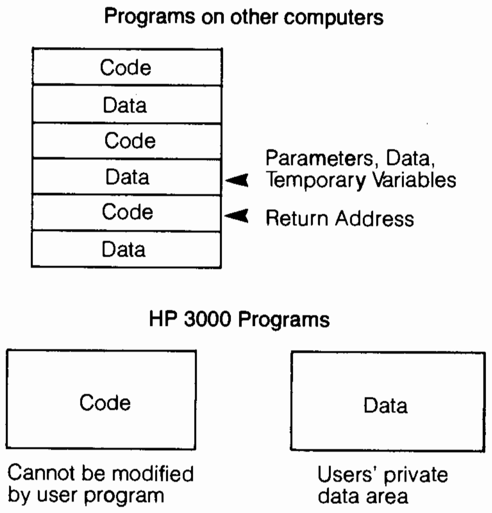

Programs on other computers
Code
Data
Code
Data
Code
Data
Parameters, Data, Temporary Variables
Return Address
HP 3000 Programs
Code
Data
Cannot be modified by user program
Users' private data area

Figure 6-2. Separation of Code and Data

Code consists of the executable instructions that make up a program or subprogram. The values and arrays used by the program or subprogram are referred to as data. Code cannot be altered by your program as there is no instruction available in the HP 3000 instruction set to allow such a modification. However, since you, and you alone, must be able to manipulate your data, the system provides you with a unique, private, modifiable data area (data stack).

In the HP 3000, code and data are maintained in strictly separate domains and cannot be intermixed (with the exception, however, that program constants may be present in code segments). This fact, plus the fact that code is nonmodifiable while active in the system, permits code to be shareable among several users. HP 3000 code is also reentrant. Re-entrant means that when a program is interrupted during execution of a code segment and another user's execution needs the same segment, that segment can be used, is completely protected against modification, and will be returned intact to the previous user's execution. Since re-entrancy allows only one copy of heavily used programs to be shared by many users concurrently, main memory can be used with optimum efficiency. Re-entrancy and stack-structured data together make possible subprogram recursion, a subprogram calling itself, which is essential for efficient compilers and system software. Also, since code is non-modifiable, exact copies of all active code can be retained on disc, thus allowing code to be overlayed without having to first write it back out to the disc.

Processes

Programs are run on the basis of processes created and handled by the operating system. The process is not a program itself, but the unique execution of a program by a particular user at a particular time. If the same program is run by several users, or more than once by the same user, it forms part of several distinct processes.

The process is the basic executable entity. It consists of a process control block (PCB) that defines and monitors the state of the process, a dynamically-changing set of code segments, and a data area (stack) upon which these segments operate. Thus, while a program consists of data in a file and instructions not yet executable, a process is an executing program with the data stack assigned. The code segments used by a process can be shared with other processes, but its data stack is private.

For example, each user working on-line through the BASIC language is running his program under a separate process; all use the same code (the only copy of the BASIC interpreter in the system) but each has his own stack.

Processes are invisible to the programmer. In normal operation you have no control over processes or their structure. However, optional capabilities are available to permit you to create and manipulate processes directly.

Variable-length segmentation

Variable-length segmentation of code and data is used to facilitate multiprogramming. This method, in comparison with paging schemes, minimizes "checkerboard" waste of memory resources due to internal fragmentation. It also makes it possible for the operating system to deal with logical instead of physical entities. This means, for example, that a particular subprogram can always be contained within one segment rather than arbitrarily divided between two physical pages, thus minimizing the amount of swapping that needs to be done while executing that subprogram. The location and size of all executing code segments is maintained by MPE in a code segment table while the location and size of all associated data segments is maintained by MPE in a data segment table. These tables are known to both hardware and software. Software uses them for dynamic memory management by the operating system. Hardware uses them to perform references and transfers between segments and to make sure that all segments required for current execution are present in main memory. Code segments may be up to 32,760 bytes in length. Data segments may be up to 65,528 bytes in length.

Segments are stored on disc and are brought into main memory only when needed. This design results in a virtual memory environment which appears to be many times larger than the maximum size of the physical main memory. It should be noted that virtual memory in MPE does not contain code segments—only the data segments which must be swapped. The code segments stay in their original positions, anywhere on the discs.

Code segmentation

Code segmentation allows you to divide your program into several segments using control statements to the compiler at compilation time or commands to the MPE operating system. Then the operating system takes over the management of these segments. That is, MPE determines whether or not a code segment should be in main memory or in disc memory. Thus you can write programs much larger than the available main memory of your HP 3000. In the example in Figure 6-3, a program has been divided into five code segments. Code segments 1, 4 and 5 are in main memory while code segments 2 and 3 are in disc memory. The code segment table (CST), which is resident in main memory, points to the active code segments. If the code segment is in main memory, the code segment table points to the beginning of that code segment. If the code segment is on disc, the code segment table points to its disc address. The management of code segments and the code segment table is completely transparent to your program. When a subroutine call is made from one code segment to the next, the instruction that makes the call examines the code segment table to determine whether or not the new code segment is in main memory. If it is, control is transferred immediately to that code segment. If the code segment table indicates that the required segment is on disc, then the instruction interrupts the operating system and informs it that a required code segment is not in memory. It is then up to the operating system to make space for that code segment in memory for execution, and transfer it to main memory so execution of the process can continue.

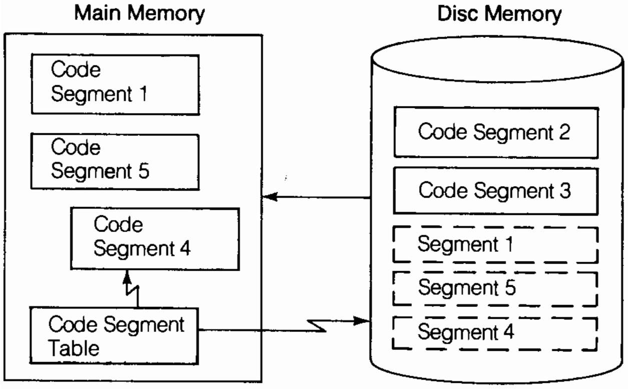


Figure 6-3. Code Segmentation

You have control over segmentation; that is, you determine where the program is divided. The result is that segmentation can be tailored to your program's logic. The size of segments can be optimized to memory size and you can avoid thrashing (unnecessary swapping of code segments due to poor segmentation—across a DO loop, for example).

Data stack

The data area of an HP 3000 program consists of a data stack, an area of main memory that expands and contracts as the program requires. In all programs there is a certain amount of global or common data that is required throughout the life of the program. This global area represents the absolute minimum size of the data stack. Beyond this, the data area grows to meet the needs of subroutines as they allocate storage for their own working areas. When a subroutine has finished its job, this area can be cut back to make use of the same memory space for the next subroutine. The end result is that less memory is required for program execution.

In general, a stack is a storage area in which the last item entered is usually the first item removed. In actual use, however, programs have direct access to all elements in the stack by specifying addresses relative to several CPU registers (the DB, S, and Q registers). The stack structure provides an efficient mechanism for parameter passing, dynamic allocation of temporary storage, efficient evaluation of arithmetic expressions, and recursive subprogram calls. In addition, it enables rapid context switching to establish a new environment on subprogram calls and interrupts. In the HP 3000, all features of the stack (including the automatic transferring of data to and from the CPU registers and checking for stack overflow and stack underflow) are implemented in the hardware.

When programming in a high-level language such as COBOL or RPG, all manipulation of the stack is automatically done for you by the language processor. You can, however, manipulate the stack directly by writing subprograms in SPL (Systems Programming Language for the HP 3000).

Figure 6-4 illustrates the general structure of a data stack as viewed from a subprogram. The white area represents filled locations, all containing valid data, while the shaded area represents available unfilled locations.

You can see that the contents of the data stack fall into four general areas. They are the global data required by all subroutines during the execution of a program, parameters that are passed to subroutines, the local data area required by the currently active subroutines, and the temporary storage required for the evaluation of expressions and intermediate results. The remaining area of the stack is unused and represents the amount of expansion possible in the stack without operating system intervention.

The stack area is delimited by the locations defined as DB (data base) and S (stack pointer). The addresses DB and S are retained in dedicated CPU registers. The Q-minus relative addressing area contains the parameters passed by the calling program. The area between Q and S contains the subprogram's local and temporary variables and intermediate results.

The data in the DB location is the oldest element on the stack. The data in the S location is the most current element. The location S is also referred to as the top of stack or TOS. Conventionally, the top is shown in diagrams downward from DB; this corresponds to the normal progression of writing software programs where the most recently written statement is farther down the page than previous (older) ones.

The area from S+1 to Z (the shaded area) is available for adding more elements to the stack. When a data word is added to the stack, it is stored in the next available location and the S pointer is automatically incremented by one to reflect the new TOS. This process is said to push a word onto the stack. To delete a word from the stack, the S pointer is simply decremented by one, thus putting the word in the undefined area.

To refer to recently stacked elements of data, S-minus relative addressing is used. Under this convention, S–1 is the second element on the stack, S–2 is the third, and so on. S-minus relative addressing is one of the standard addressing conventions. The others are DB-plus relative addressing and Q-minus and Q-plus relative addressing (the Q-register separates the data of a calling program or subprogram from the data of a called subprogram).

Since the top elements of the stack are the most frequently used, there are several CPU registers which may at various times contain the topmost stack elements. The use of CPU registers in this way increases the execution speed of stack operations by reducing the number of memory references needed when manipulating data at or near the top of the stack. These registers are implicitly accessed by many of the machine instructions and whenever the top stack locations are specifically referenced. Data stacks are automatically expanded by the operating system during execution up to a maximum size of 64k bytes (k = 1024).

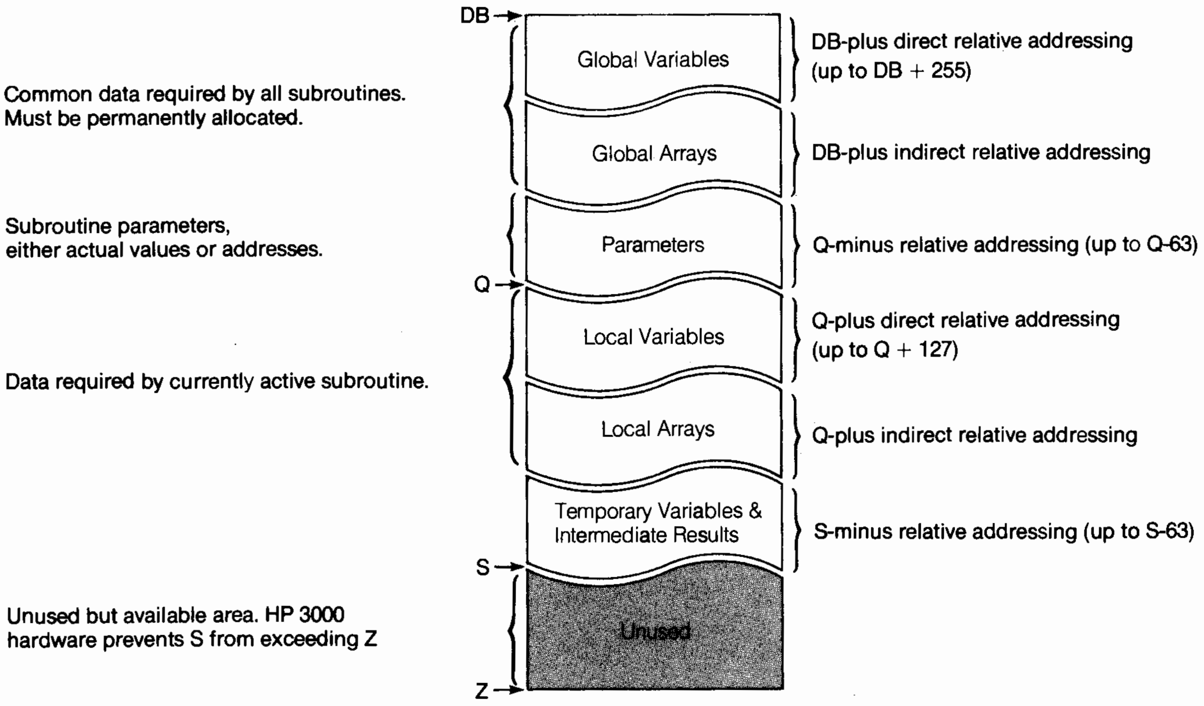

DB
Global Variables
DB-plus direct relative addressing (up to DB + 255)
Common data required by all subroutines. Must be permanently allocated.
Global Arrays
DB-plus indirect relative addressing
Subroutine parameters, either actual values or addresses.
Parameters
Q-minus relative addressing (up to Q-63)
Q
Local Variables
Q-plus direct relative addressing (up to Q + 127)
Data required by currently active subroutine.
Local Arrays
Q-plus indirect relative addressing
Temporary Variables & Intermediate Results
S-minus relative addressing (up to S-63)
S
Unused but available area. HP 3000 hardware prevents S from exceeding Z
Unused
Z

Figure 6-4. Data Stack Contents

Registers

The architecture of the HP 3000 employs a set of specific. purpose registers rather than a set of general purpose registers. Each register is included in the system to efficiently perform a single specific function.

All addressing of code and data is done relative to hardware address registers. Thus by simply changing the base addresses in these registers, segments are dynamically relocatable in memory. The few instances where absolute addresses are required are privileged operations handled by the operating system.

Approximately one-half of the HP 3000 registers are accessible to user programs and/or the operating system and its related software. The remaining registers are used, for example, by the interrupt system and for microprogram processing. The registers for the Series 30 and 33 are summarized in Appendix D, and registers for the Series III are summarized in Appendix F.

Virtual memory

Virtual memory is a very efficient memory management scheme which, in addition to main (semiconductor) memory, uses disc storage as secondary memory. Users' program code and data are divided into variable-length segments which reside in secondary memory. As a program is being executed, only those segments of code and data which are required at a particular time actually reside in main memory; the other related segments remain on disc until they in turn are required. When a particular code segment is no longer needed, it is simply overlayed by another segment.

This can be done because in the HP 3000, code segments are nonmodifiable and re-entrant. When the segment is needed again, it is simply copied from the disc on which the program resides. Since programs are copied into main memory directly from disc memory, they need not be copied prior to execution to a special "swapping disc." Data segments, however, are dynamic, and their contents can change during execution. Therefore, when a particular segment is no longer needed, it is automatically copied back to the system disc (replacing the previous version of that segment on the disc), and the main memory space of that segment is then available for other segments. The process of transferring segments between secondary memory and main memory is referred to as swapping. Whenever a segment is referenced, the hardware checks to see whether it is in main memory; if it is not, the operating system is invoked to bring it in. Thus the management of the virtual memory is totally automatic and transparent, and the system can reference a virtual memory space far larger than the real memory available.

Microprocessor

The heart of the HP 3000 hardware is the microprocessor. Each HP 3000 machine instruction is mapped into a unique microprogram that resides in control memory and consists of a series of 32-bit instructions that are executed by the microprocessor. One of the first operations in each microprogram is the execution of a NEXT instruction which begins fetching the next instruction from the HP 3000 program. Thus, when the first microinstruction is finished, the next instruction is ready for execution. This parallel execution of one instruction while the next one is being fetched, or pipelining, speeds up the overall execution time.

The primary benefit of a microprocessor is that each instruction does not require its own unique hardware logic. Instead, the instructions share the logic of a common processor called the microprocessor. This means that there is much less hardware involved in order to achieve a very sophisticated and powerful instruction set. Another advantage is that the instruction set is tailored to the compilers and the operating system. Also, the operating system's work is reduced by putting some of the more complicated operations in micro code. For instance, the operation of determining whether or not a code segment is resident in core or on disc is taken care of for the operating system by an instruction. Finally, with a microprocessor, rather than an instruction set implemented in hardware, it is easy to add new instructions. This allows Hewlett-Packard to develop new CPU hardware technology and incorporate it under MPE without changing user software.

Microcode

The entire instruction set of the HP 3000 is in the form of microcoded operations in read-only memory. This microcode is executed by a microprocessor in response to machine instructions fetched into the CPU. By allowing microprogrammed hardware to execute certain repetitive functions, such as subprogram linkage, string processing, and buffer transfers (traditionally software-implemented), the amount of code and execution times are greatly reduced.

In addition to the instruction set, many system operations that in the past were programmed in software have been microcoded. These operations are requested by machine instructions that each, in turn, execute multiple microinstructions built into the central processor hardware. Some of the standard system functions which have been microprogrammed include the interrupt handler, a cold-start loader, the saving of critical environment information on power failure, automatic restart upon restoration of power, and a set of microdiagnostics that can be invoked from the front panel of the system.

The microprogrammed instructions routinely check for addressing bounds violations during execution and automatically interrupt to error handling routines if violations occur. These memory protection checks are usually overlapped with the operand fetch and therefore do not slow execution.

Instructions

There are over 200 unique HP 3000 instructions which are microcoded in a read-only memory (ROM) under separate microprocessor control. Many of these instructions have multiple actions, several addressing modes, indirect addressing, and/or indexing which give a high complexity-to-instruction ratio. Code compression is achieved through the use of no-address (stack) instructions which implicitly use the contents of the stack registers as operands. All instructions except the stack operations are in 16-bit format; the stack operations may be packed two per 16-bit word to further enhance the code density.

A complete set of arithmetic instructions provides integer (16-bit two's complement), double integer (32-bit two's complement), logical (16-bit positive integer), 28-digit packed decimal (BCD coded digits packed two per byte), floating-point (32 bits including a 23-bit precision mantissa), and extended precision floating-point (64 bits including a 55-bit precision mantissa) arithmetic.

Other instructions are designed to facilitate string processing, subprogram linkage, and loop control. Certain special instructions are designated as privileged, meaning that they were designed specifically for use by the operating system. They may, however, be used by programs which the installation permits to run in privileged mode. Some of these special instructions, such as the DISP instruction for entry to the MPE dispatcher, instructions for enabling/disabling process switching, and instructions for data transfers between data segments, contribute greatly to the efficiency of the operating system. Complete machine instructions are provided in Appendix E for the Series 30 and 33 and Appendix G for the Series III.

Section II

HP 3000 COMPUTER SYSTEM REFERENCE SHEETS

The reference sheets which follow contain the specifications of the software and hardware products which comprise the HP 3000 Series 30, Series 33 and Series III computer systems.

As availability is subject to change, you should refer to the HP 3000 Price/Configuration Guide for the most current product availability information.

HP 3000 Business Systems

HEWLETT PACKARD

Models 32412B, 32413B, 32430B, 32431B, 32435B

HP 3000 systems are virtual memory computers with true multiprogramming and multilingual capabilities. They can simultaneously handle transaction processing, data communications, on-line program development, and batch operations in any of five high level programming languages (COBOL, RPG, BASIC, FORTRAN, and SPL, the HP 3000 Systems Programming Language). Additionally, the Series III offers programming in APL\3000.

Operating system

A powerful disc-based operating system, Multiprogramming Executive III (MPE III), optimizes the processing of multiple users who are concurrently communicating with the system. MPE is designed to dynamically allocate such system resources as main memory, the central processor, and peripheral devices to each program as needed. At the same time, each user operates in an environment of complete security without interference or illegal access from unauthorized users.

Data management

A wide choice of data management facilities is available to HP 3000 users. The award winning IMAGE/3000 data base management system allows information to be related logically between data sets (files), minimizing data redundancy and facilitating information retrieval. IMAGE/3000 handles multiple files and makes it easy to define and create a data base tailored to your specific needs. QUERY/3000 allows both programmers and non-programmers to access an IMAGE data base with simple English-like commands. The Keyed Sequential Access Method subsystem (KSAM/3000) also extends the file system by providing files which may have one primary and up to 15 alternate keys, with retrieval based upon the value of the data. To simplify data entry procedures, the HP VIEW/3000 subsystem facilitates the design and maintenance of CRT terminal data entry screens and provides edit checking of entered data. HP VIEW/3000 can be used for data entry without separate programming or may be the front-end of a transaction processing system.

Data communications

Data communications subsystems extend the basic asynchronous terminal communications under MPE to include synchronous multipoint terminal communications (MTS/3000); IBM 2780/3780 emulation (RJE/3000); IBM multileaving workstation emulation (MRJE/3000), providing batch communications with HASP II, JES 2, JES 3 and ASP job entry subsystems on the mainframe; and Distributed Systems software (DS/3000). DS/3000 provides the capability to establish interactive communications links between different types of Hewlett-Packard computer systems in geographically dispersed locations. Refer to the HP 3000 Price/Configuration Guide for data communications availability with specific HP 3000 systems.

Manufacturing applications software

Manufacturing applications software (MFG/3000) is currently offered for sale in North America, Europe and Australia. Designed for the discrete manufacturer who assembles standard, multi-piece products in lots, MFG provides an integrated on-line system for managing the materials planning and control function of the manufacturing operation.

Compatibility

The HP 3000 family of compatible business systems is comprised of the Series 30, Series 33, and Series III systems, each of which offers a full range of peripheral options, access to HP 3000 system software, and expandable hardware configurations. All HP 3000 systems feature totally compatible system software and application programs. Applications developed on one HP 3000 system can be executed on any other HP 3000 system without modification or recompilation. Each HP 3000 is supplied with the facility to execute compiled programs without the source language compiler on the system (except for programs written in APL\3000). The systems differ only in performance and expandability.

Features

- Patented stack architecture
- Virtual memory
- Automatic memory fault detection and fault correction
- Microprogrammed CPU
- Multiprogramming operating system
- Complete system security and automatic accounting of resources
- Concurrent CPU and I/O operations
- Integrated terminal access with modem support
- Rechargeable battery packs to maintain memory data during power failure.
- Automatic restart after power failure
- Remote diagnostic capability (Series 30, Series 33)
- Fundamental Operating Software
- Software compatibility between HP 3000 Series 30, Series 33, Series II, and Series III systems

Availability of software and hardware products is subject to change. Refer to the HP 3000 Price/Configuration Guide for the most current product availability information.

The HP 3000 Fundamental Operating Software consists of the following:

- MPE III (Multiprogramming Executive) Operating System
- IMAGE/3000 Database Management System
- QUERY/3000 Database Inquiry Program
- KSAM/3000 Keyed Sequential Access Method
- HP VIEW/3000 Forms Management and Data Entry Subsystem
- Utilities Set (EDIT/3000, FCOPY/3000, SORT-MERGE/3000, and utilities for system administration)
- Facility to execute compiled programs without the source language compiler on the system (except for programs written in APL\3000)

Multiprogramming Executive Operating System

The Multiprogramming Executive, MPE III, is the general purpose, disc-based operating system which supervises the processing of user programs on HP 3000 computer systems. Designed to take advantage of HP 3000 features such as virtual memory and a stack architecture implementing separation of code and data, MPE allows multiple users to concurrently access all of the computer system resources.

Features

- Multiprogramming: Concurrent transaction processing, data communications, on-line program development and batch processing
- Virtual memory
- Stack architecture: Separation of code and data, variable length segmentation, and data stacks
- Concurrent multilingual capability: COBOL, RPG, FORTRAN, BASIC, APL, and SPL
- File system with file backup, user logging, and security
- System security and complete accounting of resources
- Friendly, powerful command language, including user-defined commands, conditional job control, on-line HELP facility, and meaningful error messages
- Device and file independence
- Input/output conveniences: Spooling of input and output, private disc volumes, and tape labels
- Complete, automatic terminal management, local and remote
- Power fail/auto restart

The HP 3000 operating under MPE is extremely versatile, performing transaction processing, data communications, on-line program development and batch processing concurrently. Programs which support multiple terminals are both easy to create and easy to manage using MPE's full set of terminal management functions. Since the system controls device status monitoring, padding characters, buffer management, and error handling, the application program simply makes regular file read/write requests. MPE also offers you two ways to construct a transaction-oriented application: central control, in which an application program manages multiple terminals; and individual control, giving each terminal user the ability to run separate programs.

MPE relieves you of many program control, input/output, and other housekeeping responsibilities by monitoring and controlling program input, compilation, run preparation, loading, execution, and output. MPE also regulates the order in which programs are executed and allocates the hardware and software resources they require. It does this by providing an account/user/group structure, a command interpreter, a file system, a scheduling mechanism for the CPU, and a memory allocation manager.

Total system security allows you to operate in an environment protected from interference or illegal access from other users. Program protection (in memory) is provided by the hardware, with access to the system controlled through a set of passwords on account, user and group names at log-on. File security is based on a series of file passwords (called lockwords in MPE) and hierarchical access restrictions that allow you to specify the degree of security.

HP 3000 system resources such as main memory, the central processor, and peripheral channels and devices are dynamically allocated to each program as needed. Typically, each user is independent of all others on the system. For those cases where you need to interact and cooperate with others on joint efforts, however, MPE provides capabilities for sharing information.

All input/output (I/O) to physical devices (peripherals) is handled by the MPE I/O system. It receives I/O requests from other system software, queues them if necessary, and performs the transfer of data to or from the device. Because all I/O devices are treated by MPE as files, you can write and compile programs without immediate concern for the physical source of input or destination of output.

Asynchronous terminal communications are automatically handled by MPE. Synchronous data communications to terminals, other Hewlett-Packard computer systems (including additional HP 3000s) and non-HP computers is achieved by optional software subsystems which extend the MPE data communications capabilities.

MPE also includes many tools for over-all management and control of the system including a power fail/automatic restart routine. In addition, system managers and system supervisors can access an accounting facility, logging facility, and a SYSDUMP program which is used to backup system software and user files.

SPECIFICATIONS:

The HP 3000 Multiprogramming Executive consists of these major components:

- Configurator
- Initiator
- System console manager
- Command interpreter
- File management system
- Input/Output system
- Virtual memory manager
- Disc space manager
- Private volumes facility
- Serial disc interface
- Tape labels facility
- Spooling facility
- Job/session scheduler
- Process dispatcher
- Segmenter
- Loader
- User trap manager
- Utility intrinsics
- Accounting facility
- System Logging facility
- User Logging facility
- Backup/restore facility
- Power fail/auto restart

Real memory supported: 256kb to 2Mb

Virtual memory supported: up to 16Mb

Maximum length of spoolfiles: 536Mb

Maximum number of I/O controllers (DRT entries): 125

Maximum number of processes: 255

Maximum data segment length: 64kb

Maximum code segment length: 32kb

Maximum number of code segments per program: 63

Maximum number of files open per program: 253

Maximum file size: 536Mb

Maximum number of extents per file: 32

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

MPE Commands Reference Manual
(30000-90009)

MPE Intrinsics Reference Manual
(30000-90010)

MPE Segmenter Reference Manual
(30000-90011)

MPE Debug/Stack Dump Reference Manual
(30000-90012)

MPE System Utilities Manual
(30000-90044)

Error Messages and Recovery Manual
(30000-90015)

Index to MPE Reference Documents
(30000-90045)

System Manager/System Supervisor Manual
(30000-90014)

Console Operator's Guide
(30000-90013 for Series II/III; 30070-90025 for Series 30 and Series 33)

HP 3000 Software Pocket Guide
(30000-90049)

Training

Hewlett-Packard offers the following training courses to give you a working knowledge of the MPE operating system:

HP 3000—A Programmer's Introduction, 5-days, #22801B at HP Technical Center, #22801X on-site.

HP 3000—System Management and Operation, 5-days, #22802B at HP Technical Center, #22802X on-site.

HP 3000—Special Capabilities, 5-days, #22805A at HP Technical Center.

HP 3000—SPL File System Introduction, 5-days, #22804A at HP Technical Center

HP 3000CX or Series I-to-MPE-III Conversion, 1-day, #22818A on-site.

Also available is HP 3000 Software Consulting, #22825A, an on-site consultation service to help in applying Hewlett-Packard software to your specific applications problems.

IMAGE/3000 Data Base Management System

IMAGE is a general purpose data base management system that allows information to be related logically between data sets (files), minimizing data redundancy and facilitating information retrieval. It provides facilities to describe data base structures, create a data base, and access, maintain, restructure and back-up data. IMAGE operates concurrently in both terminal and batch environments within the constraints of an external (MPE) and internal security scheme. Application programs for use with IMAGE may be written in COBOL, RPG, FORTRAN, compiled BASIC, and SPL (Systems Programming Language). When used in conjunction with a computer network featuring Distributed Systems/3000 software (DS/3000), IMAGE allows direct access and modification of remote data bases.

Features

- Network structure allowing for complex data relationships
- Password security at data base, data set, and data item levels
- Serial, direct, calculated, and chained access methods
- Concurrent terminal and/or batch access to one or more data bases on local or remote HP 3000
- Library routines callable from COBOL, FORTRAN, BASIC, and SPL; RPG language interface

IMAGE components

IMAGE consists of three components:

Data base definition language. The data base designer describes the data items (fields), security, data sets (files), data set relationships, and storage requirements using **data base definition language;** this description is called a **schema.** Schema statements are then processed by a schema processor (DBSCHEMA), to create a stored data structure known as the **root file.** The location and relationships of information are known to IMAGE through this root file.

Data base management intrinsics. To access and maintain data on both local and remote HP 3000 computer systems, a set of IMAGE library routines is provided. These intrinsics are callable from user written programs, and include the following:

DBOPEN Initiates access to a data base.

DBINFO Returns information about the data base currently being accessed.

DBGET Retrieves items from data entries (records).

DBPUT Adds new data entries to the data base.

DBLOCK Provides temporary exclusive control of a data base.

DBCLOSE Terminates access to a data base.

DBFIND Prepares for chained access to data entries

DBUPDATE Modifies existing data entries.

DBDELETE Deletes data entries.

DBUNLOCK Relinquishes temporary exclusive control of a data base.

DBBEGIN When logging, designates the beginning of a transaction and, optionally, writes a user's information record to the log file.

DBEND When logging, designates the end of a transaction and, optionally, writes a user's information record to the log file.

DBMEMO When logging, writes a user information record to the log file.

DBCONTROL Allows a program operating in exclusive mode to enable or disable the "deferred update" option.

DBEXPLAIN Examines status information and prints an English-language message on the standard listing device

DBERROR Examines status information and supplies an English-language message in a buffer.

Data base utilities. Stand alone utility programs aid in the creation and the maintenance of the data base.

DBUTIL

(1) Allocates and initializes disc space for a data base

(2) Re-initializes the data sets of a data base back to their empty condition

(3) Purges the root file and all data sets of a data base

(4) Activates or deactivates access to a remote data base.

DBSTORE Produces a physical copy of a data base on a serial storage device.

DBRESTOR Copies a data base from a serial storage device.

DBUNLOAD Produces a logical copy (data only) of a data base on a serial storage device.

DBLOAD Loads data saved by

DBUNLOAD into an existing data base on disc.

Data types

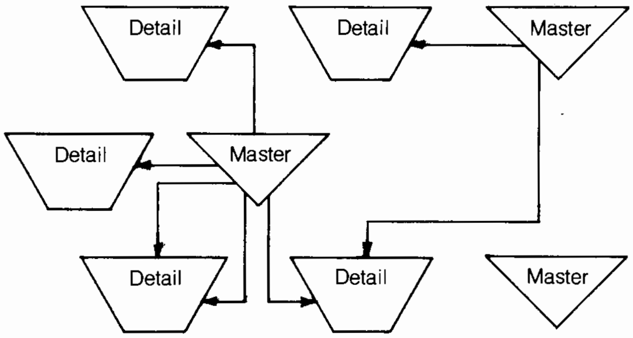
Signed binary integer, logical binary, real, ASCII character string, packed decimal, and zoned decimal.

Data set types

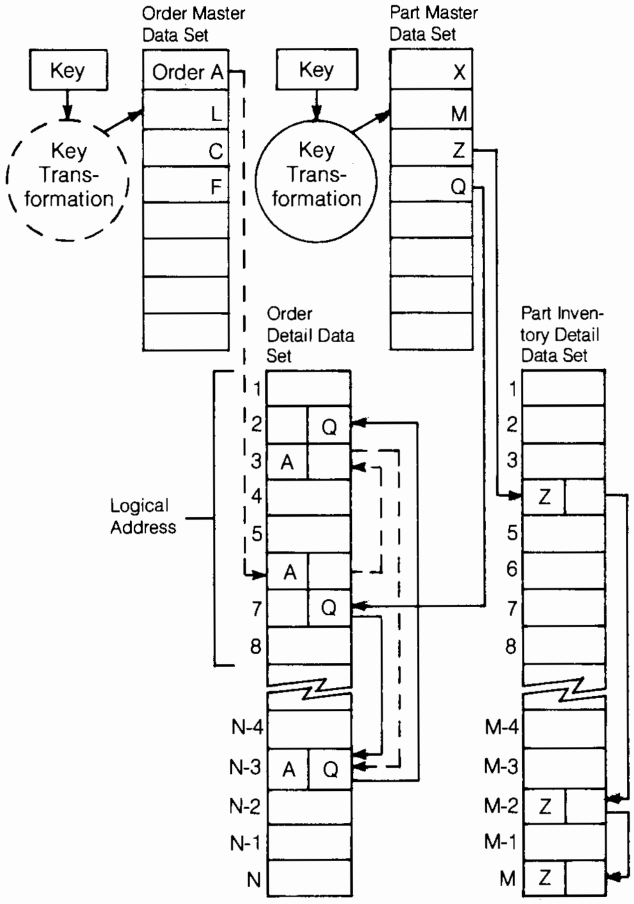
IMAGE supports two types of data sets, master and detail. Access to data entries in a master data set may be calculated, based on the key value of the data entry. Access to a data entry in a detail data set is usually via a particular master entry and a particular relationship between the master and detail data sets.

The logical relationships which may exist are exemplified in the following diagram:

Image Network Structure



An example of data set relationships is shown in the following diagram:



Master detail chain paths:

Order Master entry A links to Order Detail entries 6, 3, and N-3 Parts Master entry Q links to Order Detail entries 7, N-3, and 2. Parts Master Z links to part inventory entries 4, M-2 and M. This data structure allows a master entry to be related to many detail entries, and a detail entry related to many master entries.

Security

IMAGE maintains privacy and security down to the item level. Up to sixty-three classes of users can be defined. A password is associated with each class. Sets of user classes can then be permitted 'read' or 'read-and-write' access to any or all data items and data sets, independent of the elements accessible to other user classes.

Restructuring

DBLOAD, DBUTIL, and DBUNLOAD utilities provide for restructuring data bases in a number of ways, including changing data item names or data set names, increasing or decreasing data set capacities, adding or removing data items at the end of a data entry, and changing data set relationships.

Accessing data bases

A data base is accessed through a call to IMAGE intrinsics from COBOL, FORTRAN, SPL, and BASIC, through the chain and read statements of RPG programs after the data has been declared in the RPG file specifications section, or from QUERY as part of the language function.

Data entries may be accessed serially, directly, or by key value.

Each of these data base access methods can be applied to both local and remote IMAGE data bases. To access a data base on a remote HP 3000, both the local and remote computer systems must include DS/3000.

Transaction logging and recovery

IMAGE/3000 offers a logging and recovery system which is based on MPE III user logging and is designed to restore data bases to a consistent state, both logically and structurally.

The IMAGE/3000 logging facility provides the means to log data base transactions to a log file on tape or serial disc. The log file is a record of all modifications to data base items, including information about previous entries as well as the current state of the data base. User text can be logged in order to facilitate future access and interpretation of the log files; it may also be a useful tool for auditing. The recovery system reads the log file to re-execute transactions against a data base back-up copy in the event of a system failure.

The data base administrator is responsible for enabling or disabling the logging and recovery processes, making logging a global function controlled at the data base level rather than at the individual user level.

Additional features

- Multiple master-detail data set relationships
- Shared and exclusive data base access
- Storage and retrieval of related entries in sorted sequence
- Access to multiple data bases
- Automatic linkage management when data is added, modified, or deleted
- Efficient disc utilization (no index or overflow areas and automatic reuse of deleted record space)

SPECIFICATIONS

- Data item names per data base: 255
- Data items per data entry: 127
- Data sets per data base: 99
- Detail data sets per master data set: 16
- Master data sets per detail data set: 16
- Search items (keys) per detail data set: 16
- Maximum entry size: 2047 words (4094 bytes)
- Entries per data set: 2^{23}—1 (8,388,607)
- Entries per chain: 65,535
- Characters per data base name: 6
- Characters per password: 8
- Characters per data set name: 16
- Characters per data item name: 16

Documentation

For further technical information, consult the following Hewlett-Packard manual:

IMAGE Reference Manual
(32215-90003)

Training

Hewlett-Packard offers the following training courses for this product:

22956A IMAGE Data Base Management, 5 days (HP Technical Center)

22956X IMAGE Data Base Management, 5 days (On-site)

QUERY/3000

QUERY is a data base inquiry facility designed to easily locate, report and update data values within an IMAGE data base. QUERY can be executed from either a terminal or batch device, and reports may be directed to either a terminal or a line printer. You communicate with QUERY through 23 English-like commands, all of which may be used with both local and remote data bases. To use QUERY with a remote data base, both the local and the remote HP 3000 must include Distributed Systems/ 3000 (DS/3000).

Features

- Interactive or batch data base interrogation
- English-like commands for simplified use by nonprogrammers
- Data base updating through addition, deletion, and modification of data records
- Formatted reporting of retrieved data including page titles, column headings, group subtotals, totals, and averages
- Command files to store complex or frequently-used commands for repeated execution

Security considerations

QUERY adheres to all the security provisions of the IMAGE data base. The security password determines which data elements (i.e., data items and data sets) you are allowed to access. QUERY returns an error whenever your security class does not match the security class of the data element you are attempting to access.

You may display the data base structure and determine which data elements are accessible by entering the FORM command. QUERY responds by lising all the data elements available, based upon the password that was entered.

Locating data

QUERY is capable of retrieving all occurrences of data within a data set which meet user-specified conditions. To accomplish this, you enter a FIND command which includes logical terms that are similar to phrases spoken in English. For example, suppose a production control manager wishes to know which part numbers are in short supply or over supply when compared with outstanding customer orders. Assuming the data base contains a part-number data set, the manager could locate all such parts with one FIND command (note that LT means "is less than" and GT means "is greater than"); FIND QUANTY LT 100 AND CUST-ORD GT 50 OR QUANTY GT 10000 AND CUST-ORD LT 1000.

QUERY responds to the FIND command by locating all the data entries (records) which contain the requested data item values.

Reporting data

After the data entries have been located through the FIND command, you may enter a REPORT command to specify which items within those records QUERY is to display. The REPORT command may also specify:

- Top-of-page titles including data and time
- Addition, counting and averaging of selected data items
- Arithmetic operations on data items using registers
- Column headings, group subtotals and totals
- Line spacing
- Up to 5 levels of sorting to produce grouped items
- Edit masks to suppress leading zeroes, insert punctuation characters, etc.
- Page skipping

For quick information, the REPORT command can simply specify that all data item names and their values are to be displayed without formatting. The LIST command will display all or specified data items, automatically formatted and with headings.

Updating a data base

Maintaining the data base can also be performed using QUERY. The ADD, DELETE, and REPLACE commands, designed for this purpose, allow insertion and deletion of data entries and replacement of data item values. When ADD is entered, QUERY prompts you at the terminal for data item values. You are not required to enter values for all items.

Frequently used commands

Repetitive or complex operations are easily performed through QUERY's ability to execute FIND, REPORT and UPDATE commands from a command file stored on disc. A command stored within a file is referred to as a procedure; a procedure may consist of one or more lines. QUERY provides commands for creating, deleting and listing procedures within a command file. Also, the lines within a procedure may be added, deleted or replaced. In addition, a sequence of commands can be stored in a file and executed at any time by entering a single command, XEQ.

Data types

The following data types are converted and error-checked during QUERY I/O operations:

- One word integer numbers
- Two word integer numbers
- Two word real numbers
- Extended precision real numbers
- One word logical real numbers
- ASCII character strings containing no lower-case alphabetics
- General ASCII character strings
- Zoned decimal numbers
- Packed decimal numbers

Documentation

For further technical information, consult the following Hewlett-Packard manual:

QUERY Reference Manual
(30000-90041)

Training

Hewlett-Packard offers the following training courses for this product:

22956A IMAGE Data Base Management, 5 days (HP Technical Center)

22956X IMAGE Data Base Management, 5 days (On-site)

KSAM/3000 Keyed Sequential Access Method

KSAM (Keyed Sequential Access Method) allows you to create and maintain disc files whose records are accessed by the value of the key fields within the data records. Each data record contains one primary key field and may include up to 15 alternate key fields. Data records are written to a KSAM file in any order without regard to a key sequence, although they may be presorted if desired. Records are accessed sequentially or randomly by primary or alternate key value, by logical record number, or in chronological (physically sequential) order. Duplicate key values are allowed, and records can be accessed by generic keys (partial key values) or by approximate keys.

Features

- Multiple keys: one primary and up to 15 alternate keys
- Duplicate key values allowed
- Retrieval by generic key value or by approximate match
- Access from COBOL, RPG, FORTRAN, BASIC or SPL
- Fixed or variable length data records

KSAM Components

KSAM consists of the following two logical components:

KSAM procedures and intrinsics. To assist in accessing KSAM data records, special sets of COBOL procedures, BASIC procedures, and MPE file system intrinsics (for FORTRAN and SPL) have been provided.

Note: Most of the standard MPE file system intrinsics can be used with little or no parameter changes to access and manipulate KSAM files.

KSAM utility program (KSAMUTIL). This utility program provides a set of commands designed specifically for creating and manipulating KSAM files.

BUILD Creates a KSAM file consisting of a data file and a key file.

ERASE Clears the contents of a KSAM data file and resets the pointers in the associated key file.

PURGE Deletes a KSAM file from the system.

RENAME Changes the name of a KSAM data file or key file.

SAVE Saves a session/job temporary KSAM file as a permanent file.

VERIFY Displays the current status of a KSAM file.

HELP Displays a description of all the KSAMUTIL commands.

EXIT Terminates KSAMUTIL and passes control back to the MPE command interpreter.

KSAM file structure

A KSAM file consists of two physical disc files: a key file and a data file. A KSAM data file contains all the data records. The associated key file contains one or more sets of entries that maintain the primary and alternate logical sequences of the data records. When a data record is added to the data file, a key entry is added to the associated key field in the new data record. These key entries are dynamically added in ascending order to maintain the sequential nature of the key file. The diagram below illustrates the functional relation between a KSAM key file and data file.

Accessing KSAM files

Data records in a KSAM file can be retrieved, updated, or deleted sequentially or in random order by key value through the key file. They can also be accessed as they were written, in chronological order, or by record number without using the key file.

KSAM files can be accessed:

RPG/3000: Through file specifications and chaining operations.

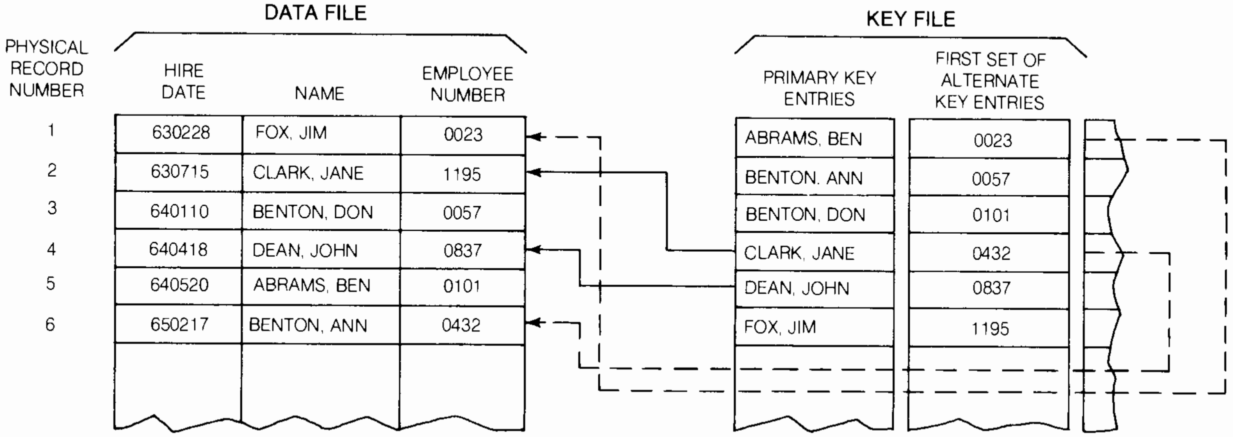
COBOL/3000: Through call statements to a special set of KSAM procedures designed specifically for use with KSAM files.

FORTRAN/3000: Through call statements to MPE file system intrinsics or to the special set of KSAM procedures.

BASIC/3000: Through call statements to a special set of KSAM procedures designed specifically for use with KSAM files.

SPL/3000: Through MPE file system intrinsic calls.

KSAM files can be opened for shared access by more than one user, or access to a KSAM file can be restricted to a single user.



DATA FILE

PHYSICAL RECORD NUMBER	HIRE DATE	NAME	EMPLOYEE NUMBER
1	630228	FOX, JIM	0023
2	630715	CLARK, JANE	1195
3	640110	BENTON, DON	0057
4	640418	DEAN, JOHN	0837
5	640520	ABRAMS, BEN	0101
6	650217	BENTON, ANN	0432

KEY FILE

PRIMARY KEY ENTRIES	FIRST SET OF ALTERNATE KEY ENTRIES
ABRAMS, BEN	0023
BENTON. ANN	0057
BENTON, DON	0101
CLARK, JANE	0432
DEAN, JOHN	0837
FOX, JIM	1195

CHRONOLOGICAL ORDER: 1, 2, 3, 4, 5, 6

PRIMARY KEY (Last Name) ORDER: 5, 6, 3, 2, 4, 1

FIRST ALTERNATE KEY (Employee Number) ORDER: 1, 3, 5, 6, 4, 2

Key file structuring

KSAM automatically performs any key file restructuring required to accommodate newly added records to the associated data file. This restructuring is invisible to you.

Data types

Data fields, including key fields, may contain the following types of data: integer, double integer, real, four-word long real, byte string, packed decimal, and numerical display.

Security

All MPE file system security provisions also apply to KSAM file.

KSAM File Recovery

If the system fails when a KSAM file is open for any type of access except read-only, the file can only be reopened after it is recovered. The file is easily recovered in most cases by running the KEYINFO command of the KSAMUTIL program.

Compatibility

FCOPY/3000 and the MPE STORE/RESTORE commands can be used with KSAM files. FCOPY can produce copies of files that are sequenced in primary key order, alternate key order, or chronological order. In addition, the NOKSAM option allows you to retrieve records that have been flagged as "deleted." The SUBSET option allows you to select only certain records based on key value or record number.

Documentation

For further technical information, consult the following Hewlett-Packard manual:

KSAM/3000 Reference Manual
(30000-90079)

Training

Hewlett-Packard offers the following training course for this product:

22828X KSAM/3000, 2-day, on-site training course.

HP VIEW/3000

HP VIEW/3000 is a data entry software product for use on an HP 3000 computer system.

HP VIEW/3000 can help users implement straightforward interactive data entry tasks easily and efficiently, and can facilitate the development of more complex terminal-oriented applications through the use of a high-level program interface. It is designed both as a stand-alone source data entry facility that can be implemented without programming effort and as a "front-end" to transaction processing applications.

Features

- **A FORMS DESIGN FACILITY** utilizing most HP 264X terminals, allows the creation of interactive screens from "fill-in-the-blanks" menus and the use of function keys. Simple edits are accepted by standard defaults and comprehensive data editing and validation can be specified by the use of a free-form field definition language.
- **A SOURCE DATA ENTRY FACILITY** allows immediate on-line entry and modification of data through forms created with the Forms Design Facility.

  This facility controls the flow of forms to a terminal, edits and validates the input data, and records that data in a special file. The same facility allows the terminal operator to "browse" through this file and to modify already entered data as desired. This entire process is completed without the need to write a single program, and it makes data collection easier to implement on an HP 3000 computer system.
- **A DATA REFORMATTING FACILITY** to change the format of entered data to meet the input requirements of existing application programs.
- **A PROGRAM INTERFACE** which aids efficient and easy implementation of source data entry to transaction processing applications. This library of high-level procedures is available to provide a simple programmatic interface between an application program on the HP 3000 computer system, the terminal, the forms and edits created by the forms design facility, the entered data, and the data file.

Data entry applications have traditionally been designed using a high-level programming language such as COBOL. Data editing was accomplished through the use of user-specified edit routines. With HP VIEW/3000 screen formats are created by simply drawing them in an interactive fashion on the CRT and selecting edit routines from a standard set. Implementation does not require programming effort or extensive training.

Forms Design Facility

The HP VIEW/3000 program FORMSPEC is an interactive forms design facility that reduces the complex problem of formatting CRT terminal screens to a simple step-by-step process. The FORMSPEC program enables the creation of screen formats or forms by drawing them on a terminal screen. Each form contains fields whose characteristics are defined from a set of standard descriptions such as type of field (Required, Optional, or Display-Only) and data type, (Character, Numerica, or Date). Default values are provided for each field to accelerate and simplify the screen development process.

The FORMSPEC facility provides comprehensive DATA EDITING, DATA FORMATTING, DATA MOVEMENT AND CONDITIONAL CONTROL functions without having to generate such routines independently:

- Comprehensive data editing capabilities are offered. Among other edits, the following are available:

  Length Check
  Range Check
  Table Check
  Equality Checks
  Pattern Match
  Check Digit Verification (Modulo 10 or 11)
- Data may be formatted as it is being collected. HP VIEW/3000 offers standard routines which justify, fill, strip, and upshift the data in the fields specified.
- Data movement may be specified to move values between fields in a single form or field values between forms. This capability, for example, allows the sum of several fields in a form to be moved to another field in the same form reserved for the total amount.
- Arithmetic and conditional processing, dependent on the value entered in a field, may be defined and specified as needed by utilizing the standard advanced edit processing features of HP VIEW/3000.
- Custom error messages can be specified with each edit characterization, to be displayed at run time.

Finally, multiple screens may be linked together for one application, and the sequence in which the forms are presented for data entry may be altered as data is collected.

All unprotected fields on a form have unique identifiers independent of their physical location on the form which allows rearranging of fields or form modification without changes to existing specifications and application programs.

Once the forms are designed, they are stored in a forms file for use whenever needed. Any form or field stored in the forms file is easy to modify either during or after initial creation.

Source data entry facility

For situations which require a stand-alone source data entry capability, HP VIEW/3000 provides a data entry program called ENTRY. The ENTRY program allows forms to be called from the forms file created by HP VIEW/3000 FORMSPEC program and to be displayed on the terminal screen. As data is entered, the ENTRY program performs the editing and validation routines specified by the designer for each field. If an error is detected during data entry, ENTRY highlights the field containing the error and displays a diagnostic message for the operator. Data can be immediately corrected and re-entered at the source.

The ENTRY program stores the corrected, entered data in a batch data file. ENTRY also allows operators to review the data in the file and, if desired, to change the entered data.

Data reformatting facility

Occasionally, it is necessary to reformat the entered data to meet the specific input requirements of a customer's application program. For this HP VIEW/3000 provides the following reformatting capabilities:

- Combining data from several forms into a single record in the output file.
- Splitting data from a single form into two or more records in the output file.
- Rearranging the data within a record, inserting constants, and generating check digits before writing it to the output file.
- Adjusting data within fields (for example, justifying the data or performing a zero fill).

The program REFSPEC allows specification of how the data in the batch file is to be reformatted and written to an output file. Specifications are entered using standard menus much like those used for forms design. The specifications are stored in a "reformat specification file."

The program REFORMAT performs the reformatting of the data. REFORMAT is a non-interactive program that requires only the names of the batch data file, reformat file, and output file to execute. It can be run at any time after data entry is complete, and the output file can then be used as input to existing application programs. A formatted listing of the output records can be requested.

Program interface

HP VIEW/3000 provides a library of high-level procedures which offer a simple programmatic interface between an application program on the HP 3000 computer, the terminal, the forms with edits defined, the entered data, and the batch data file. These procedures provide control from the user's RPG, COBOL, BASIC, FORTRAN, or SPL application program.

The following table lists a few of the functions these procedures perform for forms management, terminal input/output, data editing, and data access.

VGETNEXTFORM—Retrieves the screen image and all editing characteristics in a single access.

VSHOWFORM—Displays the current form, any data in the data buffer and diagnostic error messages on the terminal.

VREADFIELDS—Reads input from the terminal.

VFIELDEDITS—Edits all fields according to forms file specifications.

VGETFIELD—Returns the value of a single field to the program.

VPUTBUFFER—Writes data from the application program to the data buffer.

Most of the procedures require only one parameter and all are easy to use.

Performance

The terminal response time and performance of programs using HP VIEW/3000 depend upon the customer's specific application and need to be examined in detail by a Hewlett-Packard Systems Engineer. Contact your local HP Sales Representative for more information regarding the applicability and performance of HP VIEW/3000 in specific application situations.

Hardware Environment

The minimum system required for HP VIEW/3000 is an HP 3000 Series 30 with 256KB of memory. The amount of memory actually needed above 256KB will depend on a customer's expectations regarding terminal response time and transaction volume.

The interactive terminals must be HP 2640B, 2641A, 2644A, 2645A, 2647A, or 2648A CRT units. Terminals may operate in either point-to-point or "multi-point" mode. All of the video and editing features of these terminals can be used, including display enhancements, the line-drawing and other alternate character sets.

Software environment

HP VIEW/3000 requires the MPE III operating system and KSAM/3000 (Keyed Sequential Access Method).

Documentation

For further information, consult the following HP manuals:

HP VIEW/3000 Reference Manual
(32209-90001)

Using HP VIEW/3000
(32209-90004)

ENTRY Operator's Guide
(32209-90003)

HP VIEW/3000 Programmers Pocket Guide
(32209-90002)

Terminal Templates

Two templates are available for the terminal function keys. One shows the ENTRY programs function keys (part no. 7120-1189); the other describes the FORMSPEC and REFSPEC keys (part no. 7120-1190).

Training

Hewlett-Packard offers the following training courses for this product:

22830A HP VIEW/3000 four-day course given at an HP Technical Center

22830X HP VIEW/3000 four-day course on-site.

Utilities

EDIT/3000

The EDIT/3000 text editor permits you to create and manipulate files of any ASCII characters. Lines, strings and characters can be inserted, deleted, replaced, searched for, etc. The files to be edited can be source language programs, such as COBOL, FORTRAN, RPG, or SPL, or text material, such as reports.

The command language is designed to include those commands that normally exist in all editors (e.g., DELETE, REPLACE, INSERT), as well as commands to write complex command sequences, where editing is based on conditions found within the text itself. For example, you can:

- Change occurrences of a character string
- Call user-written procedures for modifying or processing text
- Execute pre-stored EDIT/3000 commands
- Use a nested, interactive loop facility for repetitive editing
- Perform multiple-line deletions, insertions, moves and replacements
- Use Boolean logic for conditional editing
- Display before editing, display after editing, do not display
- Set and reset margins during operation
- Store data in a hold file to be duplicated into another portion of the work file
- Selectively concatenate portions of files
- Easily modify complex text using a line by line template display (MODIFY command)

Documentation

For further technical information, consult the EDIT/3000 Reference Manual (03000-90012).

FCOPY/3000

FCOPY/3000 is a program used for general file copying operations. In addition to this basic capability, it can translate character code, dump files in a user readable form, verify a copy operation, select a subset of a copied file, and ignore a specified number of read errors from a source file. These functions can be performed as a single operation or as multiple operations within a single access to FCOPY.

Character code translation gives you the ability to convert EBCDIC and BCD source files to ASCII and vice versa.

Dump formatting allows for the formatting of octal, hexadecimal, and character dumps. When you specify the dump formats and title, the utility automatically establishes the dump format according to the output device type.

The copy verification capability allows you to compare two files. When a compare error is found, you are given both the record and the word or byte number where it occured.

Through the subset option, you can select a portion of a file based on field content, or number of records starting with a given record, or all records contained between two record numbers.

FCOPY can copy files from any supported input device to any supported output device. When using this utility to copy files from a tape cassette on another terminal, you must first copy the files to an intermediate I/O device (such as a disc).

Documentation

For further technical information, consult the FCOPY/3000 Reference Manual (03000-90064).

SORT-MERGE/3000

SORT-MERGE/3000 consists of two programs: Sort and Merge. Sort allows the user to read individual data records from one or more files, rearrange them in the user's choice of collating sequence (ASCII, EBCDIC, or user-defined), and write them into another file in the new sequence. Merge combines several sorted files to form one sequential file.

The user can use SORT-MERGE/3000 as a standalone program or call it programmatically from COBOL, FORTRAN, or SPL programs.

The major features of SORT-MERGE/3000 include:

- Alternating collating sequences can be defined by the user
- Single or multiple input files may be used for Sort
- Display of collating sequence and translation table on the terminal, line printer, disc or tape
- Fixed, variable or undefined length records may be used
- Signed numeric keys can be displayed
- Ascending and descending sort by keys can both be performed
- Multiple keys can be used
- Keys can be contiguous, separated, or overlapping
- Keys may be of multiple data types
- Record size is unrestricted
- Input and output media may be of various types (e.g., disc files, magnetic tapes, cards, printer output, etc.)
- The sorted output can be chosen from sequenced records, key fields, record numbers or record numbers plus key fields
- User specified routines may be used for key compare, pre-processing, and post processing.
- Any sorted files can be merged

Documentation

For further technical information, consult the SORT-MERGE/3000 Reference Manual (32214-90001).

Facility to Execute Compiled Programs Without the Source Language Compiler

The facility to execute compiled programs without the source language compiler on the system (except for those written in APL\3000) allows programs compiled on one HP 3000 to be run on other HP 3000 computer systems, without requiring the compilers for the languages in which the programs were written. This facility of the Fundamental Operating Software allows a single compiler on one system to supply the program development needs of an entire network for that language.

Scientific Library

HEWLETT PACKARD

Model 32205B

The Scientific Library is a collection of procedures that perform the functions required most often in scientific applications. These procedures may be called directly by user programs written in FORTRAN or SPL; most of these functions may also be accessed from BASIC.

Features

- Error function of a single-precision or extended-precision number
- Complementary error function of a single-precision or extended-precision number
- Gamma function of a single-precision or extended-precision number
- Natural logarithm of the gamma function of a single-precision or extended-precision number
- Single-precision or extended-precision elliptic integral function of the first or second kind
- Single-precision or extended-precision complete elliptic integral function of the first or second part
- Special integral functions (exponential, sine-cosine, and Fresnel) of a single-precision number
- Jacobian elliptic functions sn, cn, and dn using single-precision numbers
- Bessel function of the first or second kind of a single-precision number
- Computation of mean, standard deviation, standard error of the mean, variance, kurtosis, skewness, minimum, maximum, and range
- Calculation of product-moment correlation coefficients, means, and standard deviations
- Extraction of a subset of correlation coefficients and a vector of a dependent variable
- Computation of multiple linear regression coefficients and related analyses
- Inversion of a symmetric matrix stored in triangular form

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the Scientific Library Reference Manual (30000-90027)

COBOL/3000 Compiler

HEWLETT PACKARD

Model 32213C

COBOL provides you with language resembling English as a programming tool. It is self-documenting, easy to learn, and permits fast program development. The language has efficient statements to simplify file descriptions, I/O, table handling, sorting, mass storage manipulation and report generation.

Features

- Packed decimal, binary and display (zoned) data types
- Sequential and random files
- File lock capability
- Interface to KSAM
  (Keyed Sequential Access Method) files
- Multiple entry points for subprograms
- Subprogram code segmentation allowed
- Data segmentation through dynamic-type subroutines
- Object code segmentation controlled by programmer
- Compile time editing
- Selective compilation
- Table handling up to 3 dimensions
- Optimal bounds checking for tables at program execution time
- Communication with COBOL or non-COBOL subroutines
- Direct communication with SORT-MERGE/3000 via SORT verb

Implementation level

The major standard describing COBOL compilers is the ANSI (American National Standards Institute) standard. Hewlett-Packard COBOL has fully implemented the ANSI 1968 standard in all categories, except report writer. The following table shows the COBOL/3000 rating.

MODULE	ANSI RATING*
Nucleus	High
Table Handling	High
Sequential Access	High
Random Access	High
SORT	High
Report Writer	Null
Segmentation	High
Library	High

*(HIGH=ANSI Level 2, LOW=ANSI Level 1, Null=ANSI Level 0)

COBOL modules

COBOL/3000 is a set of functional processing modules that have the following capabilities:

Nucleus: Provides a basic language capability for the internal processing of data within the basic structure of the four divisions of a COBOL program.

Table Handling: Used to define tables of contiguous data items and access an item relative to its position in the table. Tables may be variable length and may have up to three dimensions.

Sequential Access: Used to access records of a file in an established sequence. Sharing memory area among files is also provided.

Random Access: Used to access records of a mass storage file according to a programmer-supplied key. Sharing memory area among files is also provided.

Sort: Used to order a file of records according to a set of user-specified keys within each record. Special processing of addition, deletion, creating, altering, editing, etc., is provided.

Segmentation: Used to specify object program segmentation requirements.

Library: Used to specify text that is to be copied from a library. Library text is available to a source program at compile time and need not be actually written as part of the source program.

Interprogram Communication: Provides the capability to call (or be called by) a program written in COBOL or other HP 3000 languages.

(continued)

Language extensions

In addition to the ANS standard, Hewlett-Packard has implemented a number of extensions which include:

- Interprogram communication
- Packed decimal (Computational-3)
- Note lines (defined by an asterisk in column 7)
- Current-date (MM/DD/YY)
- Time-of-day (HHMMSS)
- THEN optional
- Multiple REDEFINEs of a given location
- Unary +
- GO TO MORE-LABELS EXIT
- Synchronized for index data items
- Forms message for special forms

Data types

COBOL/3000 allows the following data types:

- Binary (Computational)
- Packed decimal (Computational-3)
- Display (Zoned)

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:
COBOL/3000 Reference Manual (32213-90001)
Using COBOL (32213-90003)

RPG/3000 Compiler

HEWLETT PACKARD

Model 32104A

RPG, the Report Program Generator, is a machine-independent, problem-oriented report generating language that is easy to learn, use, and code. It allows you to specify many important operations with a minimum of effort by making simple entries on specially formatted coding sheets. Because RPG is a standard language available on many different machines, programs can be submitted coded in another manufacturer's RPG or RPG II, directly to the Hewlett-Packard RPG compiler with little or no re-coding for conversion. In addition, the RPG compiler helps detect errors at the source language level with extensive diagnostic messages.

Features

- Automatic program segmentation
- Edit codes
- Calculation control of I/O
- Closed subroutines
- Single dimension arrays
- Automatic EBCDIC/ASCII file translation and alternate collating sequence
- Cross reference
- Formatted dump
- Preselected or dynamic run time error options
- Source level debugging with DEBUG
- Spread cards
- File error option

RPG can handle jobs ranging from simple tasks such as printing address labels to very complex ones such as an entire payroll process (printing paychecks, payroll registers, and various allied reports). RPG is ideal for producing such documents as inventory lists, billings, invoices, insurance benefit notices, or summaries of sales, profits and losses, and customer transactions. It is also excellent for updating the master files used in producing these documents.

Extensions to RPG II

Parameters for external subroutine calls: Parameters may be specified after an EXIT (external subroutine call) operation, simplifying interfacing with COBOL, SPL, BASIC, or FORTRAN subroutines.

Interface to data base management: Data bases can be accessed through regular I/O reads and writes by specifying the file as being an IMAGE data base in the file specification section. Keyed sequential files with up to 16 keys may be accessed in the same program.

Run-time error options: Three methods are provided for handling run-time errors.

1. Specifying on the Control Record at compile time whether the run-time error should be ignored or the program terminated.
2. Allowing the operator to determine the mode of operation at run-time.
3. Testing an error code in RPG calculations and determining the mode of operation programmatically.

Cross reference option: A cross reference may be requested showing all references to file names, indicators and field names.

Automatic program segmentation: RPG will automatically segment code generated for an RPG program in programmer-selectable 2k, 4k, 5k or 8k-byte segments, resulting in a large virtual workspace.

EBCDIC/ASCII automatic translation: You can request RPG to automatically generate file translation tables for EBCDIC to ASCII or ASCII to EBCDIC conversions, or to use an EBCDIC alternate collating sequence.

Combined terminal file: You may define an Input/Output terminal file.

Calculation indicator repetition: Duplicate conditioning indicators need not be repeated line-to-line in calculation.

(continued)

Data types

RPG allows data to be input or output in the following formats:

- Binary—one or two word binary data
- Packed decimal
- Alphanumeric
- Unpacked decimal
- Unpacked decimal with leading or trailing sign

Specification types

The statements that describe the input, processing, and output to the compiler must be written according to the rules of RPG. The seven specification types are:

- Control record
- File description
- File extension
- Line counter
- Input
- Calculation
- Output

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the RPG Compiler Reference and Application Manual (32104-90001).

FORTRAN/3000 Compiler

HEWLETT PACKARD

Model 32102B

FORTRAN/3000 is a full implementation of the ANSI STANDARD FORTRAN (X3.9-1966). It also has many extensions which expand the capabilities and increase the power of the language.

Features

- Seven data types—integer, double integer, logical, real, double precision, complex, and character
- Character variables and character arrays
- Bit extract and deposit capability with partial-word designators
- Arrays with up to 255 dimensions
- Named common blocks initialized by block data subprograms
- Multiple entry points for subprograms
- Support of user-written error handling routines which are called under trap conditions
- Parameter statement for specifying constants with symbolic names
- Dynamic array declaration and allocation in subprograms
- Up to 99 files available during execution of a FORTRAN program
- Functions and subroutines called recursively
- One logical IF statement as the dependent statement of another logical IF
- Parameters to non-FORTRAN subprograms passed by value rather than reference
- ACCEPT and DISPLAY statements for free field input/output
- Compilation time editing
- Symbolic names with up to 15 characters
- Action labels specified in READ/WRITE statements to indicate point of transfer in case of end-of-file or I/O error
- Label used as an argument in subprogram call statements to allow alternate return points
- Mixed mode arithmetic
- Generic functions
- Built-in optional cross reference listing
- Undefined variable detection

Source program format

FORTRAN/3000 was designed with several powerful convenience features for interactive users. The nature of terminal devices makes the historical position-dependent fixed-format program representation inconvenient; however, FORTRAN overcomes these drawbacks by offering both fixed format and free format representation for source language input.

File facility

Uniform access to disc files and standard input/output devices is accomplished through the MPE file system. You access your files using normal READ/WRITE statements. The structure of a file and method of access can be defined via a file statement or left to default values. This provides device independence and easy access to all types of files, including KSAM files and IMAGE data bases.

Device type can be defined at execution time; consequently, the devices used by a program can be readily changed.

Sequential and random access of disc files is supported by FORTRAN/3000.

If you have highly specialized requirements you may communicate directly with the MPE file system. Data file privacy is achieved through the normal MPE protection mechanisms.

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

30000-90040 FORTRAN Reference Manual

32102-90002 FORTRAN Pocket Guide

BASIC/3000 Interpreter and Compiler

HEWLETT PACKARD

Model 32111A

BASIC is an easy-to-learn language designed especially for interactive terminal use. Applied extensively in business, educational, and scientific environments, BASIC/3000 provides all the features commonly found in a BASIC system, as well as numerous extended capabilities. Consisting of both an interpreter and a compiler, BASIC/3000 allows you to create and debug your BASIC programs interactively and then compile them for faster production execution.

Features

- Programs and data files accessed from either time-share or batch mode
- Conversational program generation with extensive messages
- Four numeric data types: real, integer, real extended precision, and complex
- Mixed mode arithmetic
- All standard functions (SIN, COS, LOG, etc.), plus matrices, strings and files
- Program segmentation with common storage
- User definable file security including password
- Can be used alone or in conjunction with BASIC Compiler
- Can call SPL subprograms
- MPE Systems Intrinsic are callable

Environment

Programs and data files can be accessed from either interactive or batch mode.

Interactive mode

Implementation of BASIC with the HP 3000 operating system (MPE) results in a very powerful language which encourages the use of extensive conversational capabilities.

Batch mode

HP BASIC itself is a flexible language that may also be used in batch mode. In batch mode, all input (program statements, commands and data) is read from the batch input device; all output is directed to the batch output device.

User tailored modes

BASIC permits full use of MPE device independence. You can link each type of input (program statements, commands and data) and output (program output, messages and listings) with any available peripheral device. This flexibility within BASIC can be employed to construct end-user packages in which BASIC is invisible to the user. The resultant simplicity of execution is especially important to instructional/educational applications.

Character string manipulation

The BASIC Interpreter incorporates the ability to define and manipulate ASCII character strings and string arrays. All digits, upper and lower case alphabetic characters, and all other printing and non-printing ASCII characters can be stored in string variables. They can be input and output at the terminal and stored and retrieved from data files. Substrings as small as zero characters and as large as 255 characters in length can be printed, concatenated and compared to other strings. These may be used for branching or sorting.

Data files

BASIC maintains three distinct file types:

- FORMATTED files provide advanced, easy-to-use capabilities that are intended for (but not restricted to) BASIC language use. These enable run-time checking of file data type.
- ASCII and BINARY files are available for communicating data to and from programs written in languages other than BASIC.

BASIC FORMATTED files may have a record size between 4 and 319 (16 bit) words. Data can be accessed either serially or on a record basis with random access to any record in the file. The ADVANCE and UPDATE statements provide the capability to access individual items within a record. Files may be created and purged either by commands or under program control.

(continued)

Subroutines

BASIC provides two types of functions and two types of subroutines:

- Built-in functions include SIN, TAN, TNH (hyperbolic tangent). Approximately 40 such functions are provided.
- User defined functions are established in the user's program and can be called from within the program. They may consist of multiple statements and local variables and arrays whose scope extends only within the declared function.
- A simple subroutine consists of a set of BASIC statements followed by a return statement. There is no explicit indication in a program as to which statements comprise a subroutine.
- External subroutines which are written in another language, i.e., FORTRAN or SPL can be called by BASIC. BASIC programs may also call external subroutines from the libraries accessible to the user.

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

03000-90025 BASIC for Beginners

30000-90026 BASIC Interpreter Reference Manual

03000-90050 BASIC/3000 Pocket Guide

32103-90001 BASIC Compiler Reference Manual

(continued)

Compiler

The BASIC Compiler provides the means for converting BASIC programs (including those which have been written, debugged and saved via the BASIC Interpreter) into machine code. Compiled BASIC programs exist in the system as actual code segments and can be run directly, rather than through line-by-line interpreting.

Features

- Supports all HP 3000 BASIC interpreter language extensions
- Faster average execution speed than interpreter
- Shareable machine code
- Can be combined with SPL procedures and FORTRAN subroutines in the same program file
- Load-and-go capability

Since the programs to be compiled are written using the BASIC Interpreter, all of the language features described for the interpreter apply to the BASIC/3000 Compiler as well.

Environment

There are three general phases in the development of a BASIC compiled program—program development, compilation and preparation, and execution.

Program development

In the first phase, a BASIC program is usually written and debugged interactively using the BASIC Interpreter commands and statements. The interpreter constructs the interpretive versiop of a program. When you are satisfied that the BASIC program runs properly in its interpretive form, you save it (SAVE, FAST) in a file. This fast save file is the "source" input to the BASIC Compiler.

Compile and prepare

The BASIC Compiler is used to compile the fast save file. The program is then prepared in a form that results in an efficient machine code version of the original program.

Execution

The third phase is to execute the program directly under the operating system using the RUN command. BASIC programs may be compiled in batch or interactive mode, and programs may be run in either mode.

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the following manuals:

03000-90025 BASIC for Beginners

30000-90026 BASIC Interpreter Reference Manual

03000-90050 BASIC Pocket Guide

32103-90001 BASIC Compiler Reference Manual

APL\3000 Language Subsystem

HEWLETT PACKARD

Model 32105A

APL\3000 is a language subsystem for the Hewlett-Packard 3000 computers consisting of an advanced implementation of APL (A Programming Language). APL\3000 is a language which in many applications contributes to reducing program development time due to its conciseness and power in manipulating arrays of character or numeric data.

Features

- Patterned after APLSV—shared variables, same standard notation and many extensions.
- Virtual workspaces—size limited only by on-line storage and the maximum APL file size.
- Large array processing—up to 500 million possible array elements.
- APLGOL, a structured language extension to APL.
- Friendly, powerful editor.
- Access to the MPEIII file system via shared variables.
- Component file system—for direct storage and retrieval of APL data.
- Commercial formatter, for quick, detailed report formats.
- External procedure calls—access to SPL/3000 procedures
- Programmatic access to APL system commands.
- Secure applications environment—error handling, function locking and environment locking.
- Dynamic incremental compiler.
- Distributed systems extensions.
- Extended control functions.
- Interactive or batch operation.
- Access through terminals with the standard ASCII interface.

APL operates in an environment known as a workspace in which the user can place variables (data) and user-defined functions (programs). Workspaces can be saved, loaded, modified, and erased. Users enter APL in a calculator mode where calculations and functions can be executed directly.

Commercial APL with extensions

APL\3000 contains the IBM APLSV extensions which typically are part of the most recent implementations of the language. Other extensions include: format (⍕), execute (⍎), scan (\ ), and matrix divide (⌹); system variables; shared variable capability for accessing MPE commands and intrinsics; system functions such as canonical representation (□CR), name list (□NL), and DEBUG and TRACE functions. In addition, APL\3000 includes several other powerful extensions.

Virtual workspace and array size

A firmware-assisted virtual memory scheme is employed in APL\3000 with the result that very large workspaces are available, constrained only by the amount of free on-line storage up to 500 million bytes. Similarly, the number of elements of an array is limited only by the workspace size.

APLGOL

Users have the option of defining functions in either APL or APLGOL, a unique Hewlett-Packard extension of APL. APLGOL facilitates structured programming by using ALGOL-like key words in conjunction with APL expressions to describe the control flow within a given function. APL and APLGOL functions can call each other.

Friendly, powerful editor

The APL\3000 editor is more powerful and friendly than standard APL del (∇) editors. It features a full text editor, used in edit mode, and a calculator mode editor. Information entered in edit mode can be stored as a character matrix, vector or user defined APL or APLGOL functions.

File access

Shared variables allow access to the MPE III file system. Thus, users can transfer data to and from peripheral devices and disc files. APL provides a straightforward mechanism for data type conversion to and from APL and the file system. The Component File System understands APL data. Users can place APL data in the files and reference it by component number. This capability allows easy data file creation and manipulation without regard to record size.

Formatting

The Commercial Formatter is a versatile system function which gives the user the flexibility and convenience of arranging output data in a large variety of report formats.

External procedures

External procedure calls permit the user to call procedures written in SPL as if these procedures were APL functions. This gives the sophisticated APL user the ability to extend the language capabilities by communicating with other subsystems. (Note: Users must be responsible for maintaining the integrity of the workspace when using external procedure calls.)

Programmatic access to system commands give APL functions the ability to execute many system commands as APL functions. These systems commands include: □EX, □LOAD, □COPY, □LIB, □ LISTF, □PCOPY, and □RESUME.

(continued)

Secure applications environment contains several extensions which allow the locking of user defined functions and environments, and the programmatic handling of errors.

Dynamic incremental compiler

APL\3000 compiles, runs and saves the compiled code statement by statement. A signature containing characteristics of the statements (such as data types, ranks, and dimensions) is saved in the compiled code for each statement. Subsequent program executions can often interpret this saved code, allowing them to rerun much faster. This compilation of code allows several performance enhancements to be implemented.

Distributed systems extensions

APL\3000 users can extend operations to distributed networks with the ability to LOAD, SAVE, COPY, and DROP remote workspaces and to CREATE, TIE, and RENAME remote files.

Extended control functions

Extended Control Functions allow access to variables and execution in the environment of the function rather than the current execution environment.

Availability

- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

32105-90002 APL\3000 Reference Manual

32105-90003 APL\3000 Pocket Guide

SPL/3000 Compiler

HEWLETT hp PACKARD

Model 32100A

SPL is the Systems Programming Language for the HP 3000 computer systems. Because it combines the efficiency of a machine-dependent language with the simple structure of a high level language, SPL was used to write the HP 3000 operating system, language compilers, system utilities, and data base management and data communications subsystems.

Features

- Self-documenting for ease of readability
- Permits access to all hardware features and data types
- Dynamic allocation of local storage for working space and local variables in procedures. Memory deallocated on exit from procedures
- Program segmentation feature
- Assemble statement permits machine level coding
- Six data types—logical, byte, integer, double integer, real, and long real

If you are used to developing your application software using assembly language, you will find that SPL provides the same efficient and powerful results while greatly reducing the development time. It allows you to write software quickly, easily, and efficiently, while producing object programs with good code compression and efficient execution times.

To simplify systems programming, SPL offers a high level language similar to (but not equivalent to) ALGOL, together with features which enable you to easily exert control over machine-dependent functions of the computer system. You can operate directly on hardware registers, perform branches based on hardware status, extract/deposit/shift bit fields, or generate any sequence of hardware machine instructions. Also, commonly executed routines can be written in SPL and called from COBOL, BASIC, RPG, FORTRAN, or APL programs.

The language provides many features normally found only in applications languages such as ALGOL and PL/1, and includes:

- Free-form structure
- Arithmetic and logical expressions
- High level statements with unlimited nesting (IF, FOR, SWITCH, CASE, DO-UNTIL, WHILE-DO, MOVE, SCAN, assignment and compound statements)
- Recursive procedures
- Variables and arrays of many data types

Environment

Programs may be compiled in batch or interactive mode.

Variables

Variables may be either "global" or "local." Global variables are those declared in the main program and are accessible from any part of the program including procedures. Local variables, however, are declared within a procedure and are only accessible from within the procedure.

Programming features

Each SPL statement is either a high-level or machine-dependent feature.

High-level features

In all programming efforts, a need frequently arises for standard program constructs, such as loops, and evaluation of arithmetic expressions. Rather than hand-coding these often-used structures each time, SPL/3000 allows you to write them at a high level. The compiler than provides an efficient, error-free code sequence in each case.

Machine-dependent features

SPL allows the use of machine-level constructs to ensure complete control of the HP 3000 computer systems. These constructs permit the following:

- Direct register references
- Branching based on actual hardware conditions
- Bit extraction, deposit, and shift
- Generation of any sequence of hardware machine instructions (in the midst of high level constructs)

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:
SPL Pocket Guide (32100-90001)
SPL Reference Manual (30000-90024)
SPL Textbook (30000-90025)

Training

Hewlett-Packard offers the following training course for this product:

22804A HP 3000 SPL File System Introduction, 5 days, HP Technical Center.

MFG/3000 Manufacturing Systems

HEWLETT hp PACKARD

Models 32380A EDC/3000, 32392A SPC/3000, 32384A IOS/3000, 32388A MRP/3000

MFG/3000 is an integrated on-line system developed by Hewlett-Packard for managing the materials planning and control function of a manufacturing operation. The four MFG/3000 software products are: Engineering Data Control (EDC/3000), Standard Product Costing (SPC/3000), Inventory and Order Status (IOS/3000), and Material Requirements Planning (MRP/3000).

A discrete manufacturer who assembles standard, multipiece products in lots represents the ideal candidate for MFG/3000; however, the software is applicable to most manufacturing operations. Hewlett-Packard Sales Representatives and Manufacturing Industry Specialists are available for consultation regarding the suitability of the MFG/3000 products for a specific application.

Features of MFG/3000

- Easy-to-use, on-line transaction menus
- On-line assistance via "help" screens
- Forms-like data entry
- Terminal-based editing and error correction
- Pre-defined data base
- On-line data base updating
- Use of proven materials management techniques

MFG/3000 overview

By maintaining bills of material and current information about items with EDC/3000 and current inventory and open order information through IOS/3000, customers can take advantage of MRP/3000's material requirements planning to meet production plans.

An important feature of the MFG/3000 software products is the ease and speed with which the user can enter, retrieve, and modify data via an interactive terminal. The user has the facility to call up a variety of "menu-like" screens on the terminal to perform many tasks associated with materials planning and control.

With the EDC/3000 and IOS/3000 products, all transaction processing employs a technique by which edits are stored in tables. These edit tables are used to enforce update security restrictions, to add or change screen format definitions, and to modify the editing rules which apply to any particular data field. Changing the edit tables requires no programming modifications and can be done by the system administrator through an on-line program.

Engineering Data Control (EDC/3000)

As the first building block in the Hewlett-Packard materials planning and control system, EDC/3000 maintains an IMAGE/3000 data base that contains basic information about every item or part in a manufacturer's materials system. This information includes descriptive and planning data, bills of material, bills of labor (standard routings), cost data, engineering changes, and work center information.

EDC/3000 features

- Terminal data entry with on-line field editing
- On-line retrievals of bills of material, where used, routing, and item data information
- Multiple engineering changes by data or order number
- Phantom bills of material for subassemblies consumed in production
- Standard batch reports for single level bills, indented bills, summarized bills, single level where-used, indented where-used, workcenters, and standard routings.

EDC/3000 on-line retrievals

There are eight on-line retrievals available. The first four are structure or "bill" oriented while the remaining retrievals are designed to provide functional data about a part. Each of the retrievals can be used in lieu of referencing paper files.

- Single level bill of material
- Single level bill of material with remarks
- Single level where used
- Routing for a part
- Inventory planning and control information
- Production control information
- Accounting information
- Descriptive information

(continued)

EDC/3000 standard batch reports

For those situations where hardcopy documentation is required, EDC/3000 provides seven standard reports. Each report is requested by indicating on a report screen which reports are desired.

For reports not specifically provided in EDC/3000, the user may use QUERY/3000, Hewlett-Packard's inquiry language. With QUERY/3000 it's easy to formulate and produce a report in minutes even without programmer training.

The standard batch reports provided in EDC/3000 are:

- Single level bill of material
- Single level where used
- Indented (multi-level) bill of material
- Indented (multi-level) where used
- Summarized bill of material
- Standard routing
- Work center listing

Standard Product Costing (SPC/3000)

SPC/3000 performs the calculations necessary to determine the current total cost for each subassembly, assembly, and end-item in a manufacturer's material planning and control system. It can also set standard costs for each product and its components.

SPC/3000 features

- Systematic cost editing and Cost Roll-up Edit Report
- Current Cost Roll-up without affecting established standard costs
- Part Selection for Roll-up by any one of four methods:

  part number
  product line
  account number
  workcenter
- Product Cost Sheet for each part costed
- Current to standard Cost Roll-over
- Costed Bill of Material

SPC/3000 input/output

SPC/3000 inputs are user-supplied current cost information maintained in the EDC/3000 data base and control parameters and part selection criteria input through on-line SPC/3000 data entry screens. The output of SPC/3000 is a series of reports which correspond to the costing features selected. These reports are:

- Cost Edit Report
- Product Cost Sheets
- Costed Bills of Material

Inventory and Order Status (IOS/3000)

IOS/3000 maintains an IMAGE/3000 data base containing current status information for every inventoried item. Information such as quantity on-hand, scheduled receipts from work and purchase orders, and scheduled issues (both allocations and backorders) for each item is recorded in the data base. An on-line audit trail of all inventory transactions is automatically maintained.

IOS/3000 features

- On-line order entry and update with table-driven editing
- On-line inventory transaction processing
- Transaction logging and batch recovery
- On-line retrievals for all data:

  Purchase and workorder review
  Inventory status and history review
  Review of allocations by part or workorder
  Review of back orders by part or workorder
- Allocation of open workorders with pre-shortage reports
- Generation of picking lists
- Filling of backorders in priority sequence at time of receipt
- On-line audit trail of all inventory activity
- Cycle counting by ABC code

(continued)

On-line input/retrieval

The major feature of IOS/3000 is its on-line, interactive nature. All transactions that create and modify orders or update inventory are done on-line. In addition, information can be retrieved on-line on specific items including:

- On-hand stock status
- Scheduled issues and receipts by part-number
- Workorder, purchase order, backorder review
- Allocations by part-number and by workorder
- Recent inventory activity for each part
- Vendor review
- Purchase order review by line-item
- Backorders for a part

IOS/3000 standard batch reports

In addition to the capability of reviewing individual orders, item status, allocations, and vendor description on-line, a number of batch reports are available. The batch reports are:

- Material requisition
- Picking list
- Inventory count list
- Inventory physical count analysis
- Vendor listing
- Purchase commitment report
- Shortage/preshortage report
- Backorder aging report
- Inventory value report
- Extra usage report
- Order list by vendor, controller, or part-number
- Allocations by order-number or part-number
- Released allocations
- Backorder list by order-number or part-number.

Material Requirements Planning (MRP/3000)

MRP is a priority planning technique used in a manufacturing operation to help optimize materials control. Utilizing the EDC/3000 and IOS/3000 data bases, MRP/3000 determines the timing and quantities of parts to be ordered and identifies the current orders to be pulled up, pushed out, or cancelled.

The customer's master schedule and other independent demand information may be stored on the IOS/3000 data base or entered directly into the MRP/3000 run. MRP/3000 is a regenerative system. It recalculates the entire materials plan on each cycle. The timing of this MRP/3000 replanning can be adapted to any desired cycle.

MRP/3000 features

- Six standard order planning policies available simultaneously on a part-by-part basis:
  Fixed order quantity
  Lot for lot
  Part period balancing
  Days of supply
  Order point
  Gross requirements
- Rescheduling of workorders and purchase orders, if desired
- Five-year bucketless planning horizon
- Firm planned order
- Single-level pegging of requirements
- Automatic recalculation of ABC codes
- Standard batch reports for order planning, exceptions, and suggested action

(continued)

MRP/3000 input/output

MRP/3000 is a batch-oriented system designed to run on a periodic basis (usually weekly). Its primary inputs are the bill of materials, item description and item planning information from EDC/3000, and order and inventory status from IOS/3000. The output of MRP/3000 is a series of reports which constitute the recommended materials plan. These reports are:

- Exception report
- Action report
- No activity report
- Controller summary
- MRP controller list
- MRP parts list

Standard application products

All MFG/3000 products are supplied with predefined screen formats, transaction edit criteria, and retrieval and report formats. They can be run "as is," or, if required, many external features of the software can be modified by non-programmers. The items that affect terminal users and are easily modified include:

- Default values for certain transactions
- Screen sequencing
- Certain field editing characteristics
- Most transaction/screen formats
- Originator numbers and capabilities
- Documentation screens
- Maximum number of records in each data base.

In addition to implementation consulting and comprehensive customer training, Hewlett-Packard offers a full range of services and documentation for each product. A set of manuals is supplied with the software to provide reference material for both the inexperienced terminal user and the system administrator.

Documentation

For further reference information, consult the following Hewlett-Packard publications:

5953-0540(47) MFG/3000 Manufacturing Systems Data Sheet for EDC/3000, IOS/3000, and MRP/3000

5953-0573(47) MFG/3000 Manufacturing Systems Data Sheet for SPC/3000

5953-0576(47) MFG/3000 General Information Manual

For further technical information, consult the following Hewlett-Packard manuals:

32379A which includes the following individual manuals.

32380-90001 EDC/3000 User's Reference Manual

32380-90002 EDC/3000 Administrator's Reference Manual

32392-90001 SPC/3000 User's Reference Manual

32384-90001 IOS/3000 User's Reference Manual

32384-90002 IOS/3000 Administrator's Reference Manual

32388-90001 MRP/3000 User-Administrator's Reference Manual

Training

Hewlett-Packard offers the following training courses for these products:

32378A MFG/3000 System Administrator and User Instructor Training, 5 days (H-P Technical Center)

32383B EDC/3000 System Administrator and User Instructor Training, 2 days (On-site)

32387B IOS/3000 System Administrator and User Instructor Training, 2 days (On-site)

32391B MRP/3000 System Administrator and User Instructor Training, 1 day (On-site)

Availability

MFG/3000 is available in North America, Europe, and Australia.

- Series 30
- Series 33
- Series II
- Series III

DS/3000 Distributed Systems Software

HEWLETT PACKARD

Model 32190A

Distributed Systems (DS/3000) software provides the capability to establish interactive communication links between different types of Hewlett-Packard computer systems. HP 3000 systems can communicate with another HP 3000 or HP 1000 system. The connected systems may be in the same location or geographically dispersed. The linking can be done by direct connection with the Hardwired Serial Interface (HSI) or the Intelligent Network Processor (INP), or via modems using either the Intelligent Network Processor or Synchronous Single Line Controller (SSLC). The ability to interconnect different systems allows flexibility in matching the right system with each of your specific data processing tasks.

HP 3000-HP 3000 communication

Features

- Remote command execution
- Remote file access
- Remote peripheral access
- Remote data base access
- Program-to-program communication

Remote command execution

From a terminal on the local HP 3000, you can execute the entire set of MPE commands on a remote HP 3000. To execute a remote MPE command, you simply enter from your local terminal the word REMOTE, followed by the desired MPE command in its normal format. No special command language is needed. To the user on the local HP 3000, it is as if his terminal is physically connected to the remote HP 3000.

Remote file access

With DS/3000 you have access to the files of the remote system, under control of the usual system file security. The remote system files can be accessed with the standard language input/output statements or using the HP 3000 utility programs. Using standard language input/output statements in application programs, local programs written in COBOL, FORTRAN, BASIC, or Systems Programming Language (SPL) can define files on a remote HP 3000 as if the data were on the local system. All that is required is a FILE command which may be made external to the program or may be included in the program. This command specifies the location of the target file. RPG programs may also use this capability by accessing an SPL subroutine, with the appropriate commands. The remote file may be utilized on a record-by-record basis or as a complete file.

With the utility programs and editor you merely issue a local MPE FILE command on your terminal prior to running the utility, defining the desired file residing at the remote HP 3000 site. Included in the FILE command must be a remote device specification denoting the location of the desired file. The utility is run as if it were accessing a local file. Complete files can be transferred in this fashion using the FCOPY system utility. For large files, more efficient transfer is accomplished using program-to-program communications.

Remote peripheral access

By treating peripheral devices the same as files, MPE-III allows the same access methods available for accessing remote files—HP 3000 utility programs and standard language input/output statements—to be used for accessing remote peripheral devices.

Remote data base access

You have the capability to directly access data bases on a remote HP 3000 under the same security protection available for local data bases. Just like the local user, the remote user must have the proper passwords. The same IMAGE intrinsics used for local data base access are used for remote data base access.

Prior to remote access, the data base must be identified. DS/3000 offers three different ways to identify remote data bases—with on-line file commands, with MPE command intrinsics, and by using a data base access file.

A remote data base can be specified using the standard MPE FILE command in the same way that remote files can be defined. By identifying the data base in this way, the application program is completely independent of the data base location. Just prior to running the application, the data base location is specified in the FILE command.

The MPE COMMAND intrinsics allow you to embed, within your application program, the commands needed to connect the remote system to your session and to access the remote data base. The user sitting at the terminal need not know that the data being used has come from a remote HP 3000 using DS/3000.

The Data Base Access file method allows your data base manager to set up a directory, called a data base access file, which defines the location of your data base and the authorized users. DS/3000 handles the rest. Neither the application nor the user has to be aware of the location of the data base. This flexibility allows you to relocate data bases without affecting your users' operating procedures or making changes in your application.

(continued)

Program-to-Program Intrinsics

Intrinsic	Master Requested Action	Intrinsic	Slave Requested Action
POPEN	Initiates and activates a slave process in the remote system and initiates program-to-program communications with the slave program	GET	Receives the next request from the remote master program.
PREAD	Sends a read request to the remote slave program asking the slave to send a block of data back to the master.	ACCEPT	Accepts (and completes) the request received by the preceding GET intrinsic call.
PWRITE	Sends a block of data to the remote slave program	REJECT	Rejects the request received by the preceding GET intrinsic call.
PCONTROL	Transmits a tag field (containing user-defined control information) to the remote slave program and receives a tag field back from the slave.	PCHECK	Returns an integer code specifying the completion status of the most recently executed slave program-to-program intrinsic. (HP 3000 only.)
PCLOSE	Terminates (kills) the remote slave process.	FINIS	Terminates communication with the master program (HP 1000 only.)

Program-to-program communication

A set of procedures (called intrinsics) makes it possible for two or more programs residing on different computer systems to interactively exchange information with one another. The two programs work together; one of the programs is the master and the other is the slave. The master program opens the data link, initiates the slave program, and is always in control. The slave program responds to requests received from the master. Programs may be written in COBOL, FORTRAN, SPL, or BASIC.

Each HP 3000 computer may have several program-to-program communications (PTOPC) active at the same time. Furthermore, the same system may be simultaneously executing the master program of one PTOPC application and the slave program for a different application. Any number of PTOPC may be concurrently active.

Data compression

When transferring data between two HP 3000 computers, you can select an optional data compression feature. By merely including a single parameter in the DSLINE command which opens the remote communications line for your session, you enable the system to compress blanks and strings of repeated characters for your data before transmission and then restore the data to its original form at the destination node. Data compression can also be specified by the system manager or console operator. This data compression capability can save transmission time, decrease response time, and increase throughput, especially when moving large amounts of data on slower, modem-linked lines. Data compression can be enabled regardless of which capability (remote command execution, remote file access, remote peripheral access, remote data base access, or program-to-program communications) is being used.

(continued)

HP 3000-HP 1000 Communication

Features

- Remote command execution
- Remote file access
- Remote peripheral access
- Program-to-program communication

Remote command execution

From a terminal on an HP 1000, a user can execute the entire set of MPE commands on the HP 3000. The output resulting from the executed commands appears at your local log-on terminal. Using DS/3000, a user on an HP 3000 can execute all RTE operator commands on the remote HP 1000. Issuable HP 3000 commands include:

AB	Abort current batch program
BL	Set buffer limits
BR	Set break flag in named program's ID segment
DN	Declare I/O device unavailable
EQ	Examine status of I/O device
GO	Restart programs out of suspension
IT	Set time intervals for programs
LU	Examine/alter device logical unit assignments
OF	Turn programs off (abort)
ON	Turn programs on
PR	Change priority of programs
RU	Start a program immediately
RT	Release program's disc tracks
SS	Suspend programs
ST	Examine the status of programs
TI	Display the current time
TM	Set the RTE real-time clock
TO	Examine/alter an I/O device's time-out parameter
UP	Declare I/O device available

Remote file access

By using various file calls (intrinsics), an HP 1000 user can access any HP 3000 file from a FORTRAN or Assembly language program. In like manner, a HP 3000 program can access HP 1000 files from FORTRAN and SPL programs. The same file actions available to a program running on the remote system are available across the DS connection using the file intrinsics.

INTRINSICS			Requested Action
1000/3000		3000/3000	
Local HP 3000 Call to Remote HP 1000	Local HP 1000 Call to Remote HP 3000	HP 3000 File Intrinsics	
DCRET			Creates a file.
DNAME	FRNAM	FRENAME	Renames a specified file.
DPURG			Removes a file and directory entry.
DOPEN	FOPEN	FOPEN	Opens a specified file.
DCLOS	FCLOS	FCLOSE	Closes a specified file.
DREAD	FREAD	FREAD	Transfers one record from a file (sequential file on HP 3000).
	FRDIR	FREADDIR	Reads a record from a direct access file.
	FRLAB	FREADLABEL	Reads a user file label.
DWRIT	FWRIT	FWRITE	Transfers one record to a file (sequential file on HP 3000).
	FWDIR	FWRITEDIR	Writes a record to a direct access file.
	FWLAB	FWRITELABEL	Writes a user's file label.
	FSTMD	FSETMODE	Changes file access mode.
	FLOCK	FLOCK	Dynamically locks a file.
	FUNLK	FUNLOCK	Dynamically unlocks a file.
	FUPDT	FUPDATE	Updates a record in a file.
DPOSN	FSPAC	FSPACE	Positions a file.
DAPOS			Positions a file to a known record.
	FPOIN	FPOINT	Resets pointer for sequential file.
	FRDSK	FREADSEEK	Prepares for reading a direct access file.
DCONT			Sends control request to peripheral device identified as a type 0 file.
	FCNTL	FCONTROL	Performs control of file or terminal device.
DWIND			Resets file to first record.
DLOCF	FINFO	FGETINFO	Returns file status.
	FCHEK	FCHECK	Requests details on file I/O errors.
DSTAT			Returns 250 bytes (125 words) of disc directory.
	FRLAT	FRELATE	Determines if file pair is interactive (requires human intervention for all input operations), or duplicative (echoes all input operations to a display device without intervention by the operating system software), or both.

(continued)

Remote peripheral access

Peripheral devices on an HP 3000 can be accessed from an HP 1000 using the same intrinsics used for files. HP 1000 peripherals can also be accessed from an HP 3000 using calls to the RTE operating system. HP 3000 FORTRAN or SPL programs can make system executive calls to write to, read from, control, or get the status of I/O devices. Calls can also be used to schedule programs without wait, request system clock time, and set the execution interval or start time of a program.

Program-to-program communication

The same intrinsics that permit two programs on different HP 3000 systems to interactively communicate can also be used between a program on an HP 3000 and a program on an HP 1000. The master program may be on either system. Each HP 3000 or HP 1000 computer in the network can be running both master and slave programs at the same time.

These procedure calls can be used from both FORTRAN and Assembly programs on the HP 1000 and from COBOL, BASIC, FORTRAN, or SPL programs on the HP 3000. The interacting programs can be in different languages.

Required Communications Hardware

Each DS/3000 link requires a hardware interface on both systems.

HP 3000 Series 30 or Series 33

Only the Intelligent Network Processor (INP), Model 30020A is supported.

HP 3000 Series II or III

There is a choice of three different interfaces on the Series II or III:

Synchronous Single Line Controller (SSLC), Model 30055A

Intelligent Network Processor (INP), Model 30010A

Hardwired Serial Interface (HSI), Model 30360A

HP 3000-HP 3000 Maximum Data Rates

Switched (dial-up)/ leased line connection	Maximum Data Rates
INP-INP	56,000 bits/second
INP-SSLC	9600 bits/second
Direct Connection (Hardwired)	
INP-INP	56,000 bits/second (up to 4000 feet)
HSI-HSI (Series III only)	2.5M bits/second (up to 1000 feet)
	1.25M bits/second (up to 2000 feet)
	(Note: For HSI links, values are the nominal transfer rate. Actual throughput depends on your application.)

HP 3000-HP 1000 Maximum Data Rates

Direct Connection (Hardwired)	Maximum Data Rates
HSI 30360A (Series II or III only)	
HSI 12889A (HP 1000)	2.5M bits/second (up to 1000 feet)
	1.25M bits/second (up to 2000 feet)
	(Note: Nominal transfer rates)

For the Series II and Series III, the INP is generally recommended with leased or dial-up lines for flexibility and enhanced HP 3000 performance. The SSLC may be used in environments characterized by light use of DS/3000, a lightly-loaded HP 3000 cpu, and data rates less than 9600 bps.

For the Series II and Series III, the HSI is recommended where systems are less than 2000 feet (610 meters) apart. Your Hewlett-Packard Systems Engineer can assist you in determining which interface is best suited for your particular DS/3000 application.

Availability

HP 3000-HP 3000 Communications

HP 3000:
- Series 30
- Series 33
- Series II
- Series III

HP 3000-HP 1000 Communications

HP 3000:
- Series II
- Series III

HP 1000: RTE-III
RTE-IV
RTE-M-II
RTE-M-III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

DS/3000 Reference Manual (32190-90001)

DS/3000 to DS/1000 Reference Manual for HP 3000 Users (32190-90005)

Training

Hewlett-Packard offers the following training course:

HP 36900E

This three-day on-site course provides the student with a working knowledge of the capabilities and features of DS/3000. Completion of the Programmer's Introduction course (22801B) and programming experience in COBOL, FORTRAN, or SPL are prerequisites.

RJE/3000 (2780/3780 Emulator) Remote Job Entry Software

HEWLETT PACKARD

Model 30130E

RJE/3000 makes the HP 3000 appear to a mainframe ("host") system as either an IBM 2780 or 3780 data communications terminal, and thus allows batch jobs for the host system to be submitted to and received from the HP 3000. RJE/3000 provides greater flexibility than the IBM data communication terminals it emulates, in that jobs may be submitted from the HP 3000 via any ordinary input device (disc file, magnetic tape, card reader, or terminal) and output received to the HP 3000 via any output device (printer, disc file, tape, card punch, or terminal.) RJE/3000 users can activate job transmission to the host, and specify job input and output devices on the HP 3000, through either interactive sessions at an HP terminal or by batch jobs running on the HP 3000.

RJE/3000—Where to use it

- For remote job entry to any mainframe ("host") computer which supports the IBM 2780 or 3780 communications terminals
- For batch input/output to an interactive application on an IBM mainframe, running under CICS or similar communications monitor that supports the IBM 2780 or 3780.
- To communicate (transmit and receive files) with the IBM 2780 or 3780, or other devices that either emulate or support the 2780 or 3780.

Features

- Runs in a full multi-programming environment on the HP 3000
- Any input/output devices on the HP 3000 may be used to submit or receive jobs (disc files, magnetic tape, card reader, card punch, printer, or terminal)
- Supports lines to multiple hosts and/or multiple lines to a single host
- Can be used with either switched (dial-up) or leased lines
- Modem communications up to 19.2 Kilobits per second using an RS-232 interface with the Intelligent Network Processor (INP) hardware interface for the HP 3000; modem or hardwired communications up to 9600 bits per second with the Synchronous Single Line Controller (SSLC) hardware interface
- Users may control job transmission to the host, and specify job input/output devices on the HP 3000, from either an interactive session at an HP terminal, or by an HP 3000 batch job

RJE/3000 includes standard 2780 capabilities with the exception of six-bit transcode, and standard 3780 capabilities except for conversational mode. Also offered are numerous user-specifiable options, including:

- Blank compression
- Short record truncation (suppression of trailing blanks) with no EM (End of Medium) control characters required in data
- EBCDIC and ASCII transparency
- Horizontal tabulation
- All 2780/3780 vertical format control
- Multi-record transmission (can optionally transmit more than seven records per block under user control)
- Print/punch component select

RJE/3000 provides ten commands for controlling RJE input and output. The commands (listed below) may be used from either an on-line session or a batch job on the HP 3000.

Command	Function
#RJLINE	Defines the characteristics of the communications line.
#RJIN	Transmits input data from your HP 3000 Computer System to the remote processor.
#RJOUT	Requests and processes routed output from the remote processor.
#RJIO	Initiates transmission of a one-line message to the remote processor.
#RJINFO	Initiates a file display printing of the communications line.
#RJDEBUG	Sets RJE/3000 into debug mode.
#RJLIST	Requests and processes unrouted list output from the remote processor.
#RJPUNCH	Requests and processes unrouted punched output from the remote processor.
#RJEOD	Transmits an EOT control character (end-of-file) to the remote processor.
#RJEND	Terminates the RJE/3000 subsystem.

Note: The terms "routed" and "unrouted" refer to the presence or absence of component select codes in conjunction with output transmitted from the remote processor.

In addition to these commands, the full capabilities of the HP 3000 are available with related service utilities (e.g. EDIT/3000) to expedite job control programming and to facilitate program/data file transfer and maintenance.

(continued)

Hardware interface specifications

- For Series 30 or 33: Model 30020A Intelligent Network Processor (INP)
- For Series II or III: Model 30010A Intelligent Network Processor (INP) or Model 30055A Synchronous Single Line Controller (SSLC)

For the Series II or III, the INP is generally recommended for flexibility and enhanced HP 3000 performance. The SSLC may be used in environments characterized by light use of RJE/3000, a lightly-loaded HP 3000 cpu, and data rates less than 9600 bps.

Your Hewlett-Packard Systems Engineer can assist you in determining which interface is best suited for your particular RJE/3000 application.

Availability

- Series 30
- Series 33
- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

RJE/3000 (2780/3780 Emulator) Reference Manual (30000-90047)

HP 3000 Guide to Synchronous Modems (see Appendix H)

MRJE/3000
Multileaving Remote Job Entry Software

HEWLETT PACKARD

Model 32192A

MRJE/3000 allows multiple users on the HP 3000 to simultaneously submit and receive batch jobs to any remotely connected host computer system using a HASP II, JES 2, JES 3, or ASP job entry subsystem. MRJE/3000 makes the HP 3000 appear to the mainframe as an IBM 360/30 "HASP" workstation. Job data may be submitted from the HP 3000 via any ordinary input device (disc files, magnetic tapes, card readers, or terminals) and output received to the HP 3000 via any output devices (printers, disc files, tapes, card punches, or terminals.) MRJE/3000 users can activate job transmission to the host, and specify job input and output devices on the HP 3000, through either interactive sessions at an HP terminal or by batch jobs running on the HP 3000. MRJE/3000 provides for an MRJE Manager, who can interactively monitor and control job activity. Host console commands can be entered by the MRJE Manager (and other users at the MRJE manager's discretion) from any HP terminal.

Features

- Runs in a full multiprogramming environment on the HP 3000
- Flexible, easy-to-use commands for job control and status inquiry
- MRJE commands may be used from both interactive terminals and traditional batch devices
- Any input/output devices on the HP 3000 may be used to submit or receive jobs (disc file, magnetic tape, card reader, card punch, printer, or terminal)
- Supports for concurrent use: an operator console, up to seven logical print streams, seven logical card reader streams, and seven logical punch streams, all interleaved on the same communication line
- Supports multiple hosts and/or multiple lines to a single host
- May use either switched (dial-up) or leased lines; modem communication speeds up to 9600 bits per second

Submit jobs on-line or off-line

MRJE users may submit MRJE jobs even if no connection exists between the HP 3000 and the mainframe. Jobs submitted off-line are spooled on the HP 3000 and are automatically transmitted when the connection is made. Output from the host is then directed to the proper HP 3000 peripheral device or file (specified by the user when he submits his job) without further intervention. If no output destination has been indicated, job output is routed to the default device designated by the MRJE manager. MRJE/3000 output may be sent directly to a printer without intermediate spooling, while simultaneously spooling local HP output for the printer, if desired.

Requirements

Host Job Entry Subsystem: HASP II, JES 2, JES 3, or ASP

Minimum HP 3000 Memory Size: 256 Kbytes
512 Kbytes of memory is often recommended for use of MRJE/3000. Consult your Hewlett-Packard Systems Engineer.

Hardware Interface: Model 30055A Synchronous Single Line Controller (SSLC)

Availability

- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

MRJE/3000 Reference Manual (32192-90001)

HP 3000 Guide to Synchronous Modems (See Appendix H)

MTS/3000
Multipoint Terminal Software

HEWLETT PACKARD

Model 32193A

Multipoint Terminal Software (MTS/3000) permits two-way, half duplex data transmission between an HP 3000 computer system and multiple multipoint terminals in the HP 2640 series via a single communications line. The terminals may be connected to the computer by means of a modem (remote access) or may be hardwired to the computer (local access) through the Synchronous Single Line Controller. MTS/3000 also provides hardwired support for the HP 307X family of data capture terminals.

Features

- Allows several terminals to share a single line
- Transmission speed of up to 9600 bps
- Full page mode operation
- Automatic error detection and retransmission of data
- Remote synchronous communication over either switched or leased lines for supported members of the HP 264X family of terminals
- Transmission of variable length blocks of data

Functional capabilities

In both the log-on (interactive) mode and the data entry mode, you can enter data into the terminal's memory using the cursor positioning capabilities, TAB key, and RETURN key. This data can then be edited as much as desired until the ENTER key is pressed, transferring the data to the computer.

Each of the multipoint terminals can be viewed as a processor with several peripheral devices—keyboard, display, two cartridge tape units, and printer. MTS/3000 allows the application program to control these peripheral devices by transmitting the appropriate escape sequences to the terminal. The capabilities provided by the escape sequences include data transfers from the HP 3000 to a peripheral device, from a peripheral device to the HP 3000, and from one peripheral to another within the same terminal. An escape sequence also exists to retrieve status information for a peripheral device.

SPECIFICATIONS

- Data transfer at up to 9600 bps
- Two-way, half duplex operation
- Line protocol is similar to IBM Binary Synchronous Communication
- 255 maximum terminal identification numbers
  —26 group identifiers
  —10 terminal identifiers per group
- Terminals supported:
  HP2641A, HP2645A, HP2645S,HP2647A, HP2648A, HP3075A, HP3076A, HP3077A
- For appropriate cabling, check the HP 3000 Price/Configuration Guide

Availability

- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard manuals:

MTS/3000 Reference Manual (32193-90002)

HP2641A, 2645A, 2645S, Display Station Reference Manual (02645-90005)

HP2647A Graphics Terminal Reference Manual (02647-90002)

HP2648 Graphics Terminal Reference Manual (02848-90002)

Models 37210T, 37220T

The HP models 37210T and 37220T provide high speed synchronous data transmission capability for use with DS/3000, RJE/3000, MRJE/3000 and MTS/3000. They require a Synchronous Single Line Controller (SSLC) or Intelligent Network Processor (INP) interface with HP 3000 computers.

Features

- 4800 and 9600 bps synchronous operation
- Automatic equalization
- RS232C (V.24/V.28) data interface
- Built in diagnostic test

Model 37210T—4800 bits/second

The Hewlett-Packard 37210T is an automatically equalized 4800 bits/sec synchronous modem. It is microprocessor controlled and is designed for point-to-point and multidrop operation over four-wire leased lines or point-to-point operation over two-wire dial-up circuits.

The modem uses automatic equalization to compensate for a wide range of group delay and attenuation distortions. At the start of transmission, the equalizer is set up by a special training sequence, after which it adapts automatically to compensate for any variations in circuit quality. The equalizer will withstand dropouts lasting up to one second without retraining and will, if necessary, retrain using the normal data signal.

Specification Summary (37210T)

Data Rate: 4800 bits/sec. Fallback to 2400 bits/sec.

Mode of Operation: Point-to-point or multipoint; 4-wire Full and half duplex, 2-wire half duplex.

Data Interface: EIA RS232C and CCITT V.24/V.28.

Circuit Requirement: 2 or 4-wire telephone circuits equivalent to type 3002 or better.

Transmit Level: 0 to −15dBm selectable in 1dB steps.

Equalization: Automatic, adaptive equalizer.

Equalizer Initialization Time: 50ms normal, 708ms long.

Model 37220T—9600 bits/second

The Hewlett-Packard model 37220T 9600 bits/sec Modem is an automatically equalized synchronous modem designed for transmission of data over four-wire point-to-point leased telephone circuits. It employs a D1 conditioned line, and LSI circuits are employed to give cost effective and trouble free performance.

The modem includes local and remote loopback tests which may be used with the built in pattern generator/error indicator to allow a modem self test to be carried out and also to enable performance checks on transmission quality to be carried out from one end of the link. As a further aid to ensuring system integrity, LED monitors are provided on each of the principal interface leads.

Specification Summary (37220T)

Data Rate: 9600 bits/sec. Fallback to 4800 bits/sec.

Mode of Operation: 4-wire full duplex transmission.

Data Interface: EIA RS232C and CCITT V.24/V.28.

Circuit Requirements: 4-wire leased telephone circuits equivalent to type 3002 or better.

Transmit Level: 0 to −16dBm selectable in 2dB steps.

Equalization: Automatic, adaptive equalizer.

Equalizer Initialization Time: 2.8 seconds.

General (37210T and 37220T)

Dimensions: 133 x 425 x 425mm (5.25 x 16.75 x 16.75 in.)

Weight: 9.2 kg (20.3 lb.)

Operating Temperature Range: 0-45° C

Supply Voltage: 100, 120, 220, 240V nominal +5 −15%
48 to 66Hz AC only.

Power Consumption: 85VA max.

Guidelines for ordering telephone lines with HP modems

FCC Modem Registration No: ABA-979-67889-DM-E

Ringer Equivalence No: 0.4B

Telephone Line Ordering Information (specify to phone company):

RJ45S programmable data jack with RTC telephone

RJ36X phone jack

RTC telephone with following options:

Aural monitoring

Modem Controls Line

Touch-tone or rotary dial

Voice mode indicator

Availability

- Series 30
- Series 33
- Series II
- Series III

Intelligent Network Processor (INP) for HP Distributed Systems Network

HEWLETT PACKARD

Models 30010A and 30020A

Models 30010A and 30020A Intelligent Network Processor (INP) allows HP 3000 computers to be linked to other computers in an HP Distributed System Network. An INP performs character handling and provides buffer storage, freeing the central processor for other tasks. It uses a Hewlett-Packard high speed silicon-on-sapphire (SOS) microprocessor.

The INP uses state-of-the-art technology to provide flexibility, growth potential and cost effective performance. The SOS microprocessor performs protocol handling, and on-board memory is used for data buffering. SOS is a Hewlett-Packard technology which offers a combination of high speed and low power consumption.

An INP can be used with the DS/3000 and RJE/3000 communications subsystems. IBM Binary Synchronous Communications (bisync or BSC) protocol is used. Data rates up to 19,200 bits/sec using modems, or up to 56,000 bits/sec hardwired or with CCITT V.35 standard interface are supported. Your Hewlett-Packard Systems Engineer must be consulted if you are planning DDS communication via V.35. Maximum data rates are a function of the communication subsystem used and hardware to which the INP is linked.

A INP requires a single Series 30 or 33 I/O slot, or two I/O slots for Series II and III, and has a self-test capability. Buffered data is protected and maintained in the event of a power failure. The INP may be used for non-concurrent support of different communication subsystems. Subsystem software is down-loaded to the INP from the HP 3000 CPU when subsystem activity is initiated. Typically, no reconfiguration of the INP is required between use with multiple subsystems.

Features

- On-line self-test
- Collection of data volume and error statistics
- Character handling and buffer storage capability
- Compatible with HP and Bell modems in both full duplex and half duplex mode
- EIA RS232-C and CCITT V.24 compatibility
- Direct connect (hardwired) capability is standard

SPECIFICATIONS

Model:

Series II and III: 30010A Intelligent Network Processor

Series 30 and 33: 30020A Intelligent Network Processor

Interface: EIA RS232C, CCITT V.24, V.35 and hardwired

Data Transfer rate: Modem—Up to 19,200 bits/sec in half or full duplex mode and up to 56 Kbps for V.35 standard interface

Hardwired—Up to 56,000 bits/sec for INP to INP

Up to 9,600 bits/sec for INP to SSLC

Modem Compatibility: HP 32210T, 37220T
Bell 201C, 208A, 208B, 209A

For Series II and Series III systems, the combined total of INPs and Synchronous Single Line Controllers (SSLCs) must not exceed seven.

Power fail protection: Buffered data is protected during a power failure

See Appendix I for a summary of HP synchronous interfaces and the communications subsystems they support.

Availability

- Series II or III—Model 30010A
- Series 30 or 33—Model 30020A

Documentation

For further technical information, consult the following HP manuals:

HP 30010A/30020A Intelligent Network Processor Diagnostic Procedures Manual (Part No. 30010-90002)

HP 30010A Intelligent Network Processor Installation and Service Manual (Part No. 30010-90001)

HP 30020A Intelligent Network Processor Installation and Service Manual (Part No. 30020-90001)

Synchronous Single Line Controller (SSLC) for HP Distributed Systems Network

HEWLETT PACKARD

Model 30055A

The HP 30055A Synchronous Single Line Controller (SSLC) provides the hardware to link the HP 3000 computer systems to other computers in an HP Distributed Systems Network by means of modems. Used in communications subsystems such as DS/3000, MRJE/3000, MTS/3000, and RJE/3000, the printed circuit card fits into a single I/O slot of the computer. The interface connects to a modem to facilitate data transfers up to 9600 bits per second.

Features

- EIA RS232C and CCITT V.24 compatibility
- Compatible with HP 37210T, 37220T, Bell 201, 208, 209 modems
- Half or full duplex operation
- Hardware data transfer rates to 9,600 bits per second
- Double character buffering
- Special character recognition
- Compatible with IBM Binary Synchronous Communications protocol

SPECIFICATIONS

Interface level: EIA RS-232-C/CCITT V.24
Operation Mode: Half or full duplex modems

Data transfer rate: Up to 9600 bps/hardware rate

Character buffering: 2

Program parity generation/checking: None, odd, even

Special character recognition: Program selectable

Synch character: Program selectable

Maximum length cable between SSLC and modem: 50 ft

Modem interface capability: HP37210T, 37220T, Bell 201C, 208A and B, 209A. (User must supply the modem.)

See Appendix I for a summary of HP synchronous interfaces and the communication subsystems they support.

Availability

- Series II
- Series III

Hardwired Serial Interface (HSI) for HP Distributed Systems Network

HEWLETT hp PACKARD

Model 30360A

The HP 30360A Hardwired Serial Interface card provides the hardware to link the HP 3000 computer systems to other computers in an HP Distributed Systems Network by means of a coaxial cable.

Features

- 2.5 M bits per second transfer rate up to 1000 ft./hardware data rate
- 4 software selectable channels
- Hardware transmission of an acknowledged word without program interruption
- Optically isolated reception
- Immediate line turnaround

Used for high-speed point-to-point data transfer, the interface fits into a single I/O slot of the computer. The maximum hardware data rate is 2.5M bits per second at up to 1000 feet. The card may also be used with cables up to 2000 feet long at 1.25 million bits per second (hardware rate).

The cable is isolated at the receiver end by an optically isolated gate. This enables long distance data transmission with high reliability against errors due to common mode noise or ground level shifting.

The HP 30360A also features 4 selectable channels under system operator control, automatic data acknowledgement or handshaking via hardware, and CRC generation, transmission, and processing through hardware.

The CRC accumulation is a 16-bit feedback shift register that implements a 15th degree CRC polynomial.

The Hardwired Serial Interface must be inter-connected with 30220A cable kits. Several optional lengths are available in lengths from 25 to 2000 feet.

See Appendix I for a summary of HP synchronous interfaces and the communications subsystems they support.

Availability

- Series II
- Series III

Asynchronous Terminal Controller

HEWLETT hp PACKARD

Model 30032B

The Asynchronous Terminal Controller (ATC) is designed to interface up to 16 terminals to the HP 3000 via the IOP bus. Terminals may be hardwired, or connected through asynchronous modems. Terminals interfaced through the ATC can be configured to the Multiprogramming Executive (MPE) as on-line transaction processing terminals accessing all the capabilities of the computer system.

Features

- EIA RS232C Compatibility
- Speed detection for up to six standard rates (110, 150, 300, 600, 1200, and 2400 bits per second)
- Automatic break detection
- Automatic answering
- Character assembly performed by hardware
- Full duplex (Echoplex), or half duplex transmission mode
- Parity generation and checking
- Modem support for Bell type 103, 212A, and 202S*/T

**For performance reasons, use of the 202S modem is not recommended.*

Availability

- Series II
- Series III

Documentation

For further technical information, consult the following Hewlett-Packard Manuals:

Asynchronous Terminal Controller Installation and Service Manual (30032-90004).

Asynchronous Data Communications Controller

HEWLETT hp PACKARD

Model 30018A

The Asynchronous Data Communications Controller (ADCC) is designed to interface terminals to the HP 3000 Series 30 or 33 over the Inter-Module Bus. Two types of ADCC boards may be used, the Main ADCC and the Extender ADCC. Each board contains four ports for connection to devices through an RS 232C interface. ADCC main supports 103, 212 and 202S*/T modems in the U.S. Addition of ADCC extender adds support for Data Rate Select (CCITT 112-required for European half-duplex modems) to all eight ADCC ports. The Main ADCC is used when four or fewer devices are connected to a channel. The Extender ADCC extends the device capability of the channel to eight. When more than eight devices are to be attached, additional Main ADCC's are required, since each Main ADCC can accommodate only one Extender.

Features

- EIA RS232C compatibility
- Operation at speeds up to 9600 bits per second
- Speed detection for baud rates to 2400 bits per second
- Automatic break detection
- Automatic answering
- Full duplex and half duplex transmission supported
- Even parity generation and checking or no parity
- Modem support for Bell type 103, 212A and 202S*/T

Availability

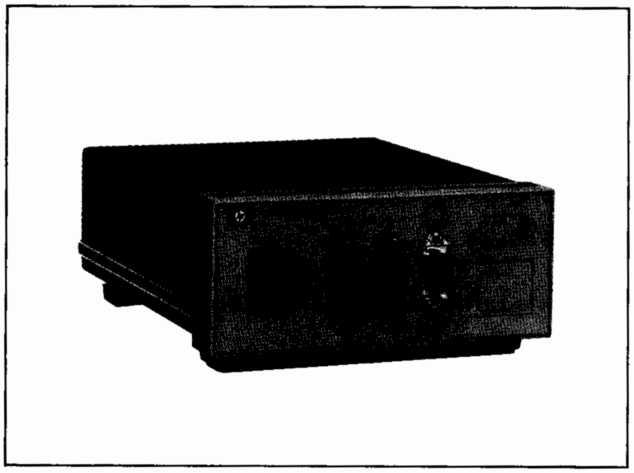
- Series 30
- Series 33

**For performance reasons the use of 202S modem is not recommended.*

Asynchronous Repeater

HEWLETT PACKARD

Model 30037A



The HP 30037A Asynchronous Repeater (AR) is a standalone device which converts standard RS232C communication signals to levels compatible with the HP family of terminals which operate in the asynchronous multipoint communication mode, such as the HP 2641A, HP 2645A, 2647A and HP 2648A display terminals.

Features

- Operates in either asynchronous multipoint or point-to-point environment
- Extends maximum cable distance between the computer and the first asynchronous multipoint terminal to 2000 feet
- Extends maximum cable distance for asynchronous point-to-point communication to 4000 feet

The addition of the AR permits the first directly connected asynchronous multipoint terminal to be located up to 2000 feet (610 meters) from the computer. This removes the 50 foot (15 meters) limitation imposed when an RS232C direct connect interface is used. In addition, each AR used can extend the distance between individual asynchronous multi-point terminals by 2000 feet (610 meters). The AR also permits extended asynchronous point-to-point communication with a maximum cable distance of 4000 feet (1220 meters). Two built-in diagnostic facilities, Self test and Loop test, are available for daisy chain verification.

SPECIFICATIONS

Communications facility: Twisted pair (22 AWG)

Data transmission rate: Up to 20k bps

Modulation: Baseband

Line conditioning: None required

Synchronization: Asynchronous

Data format: Serial

Operational mode: Full duplex

Terminal interface: RS232C

Line interface: Four/two twisted pair (22 AWG)

Diagnostic facilities: Self test and loop test

Power requirements: 88 to 126 VAC, 198 to 252 VAC

Line frequency: 47-66 Hz

Line current: .17 A RMS, .5 A Turn-on surge

Physical characteristics

Width: 8.3 inches (21.08 cm)

Height: 3.5 inches (8.89 cm)

Depth: 10.5 inches (26.27 cm)

Weight: 7 lb. 4 oz. (3.289 kg)

Environmental conditions

Since the Asynchrónous Repeater may be used outside the immediate system area, it is not subject to the environmental specs of the system itself.

Temperature, free space ambient

Non-operating: −40° to +75°C (−40° to +167°F)

Operating: 0° to +55°C (32° to 131°F)

Humidity: 5% to 95% (non-condensing)

Availability

- Series 30 (point-to-point)
- Series 33 (point-to-point)
- Series II (point-to-point or Multipoint)
- Series III (point-to-point or Multipoint)

Documentation

For further technical information, consult the following Hewlett-Packard manual:

HP30037A Asynchronous Repeater Installation and Service Manual (30037-90003)

General I/O Channel

HEWLETT PACKARD

Model 30079A

The General I/O Channel is the primary channel for communication between the HP 3000 Series 30 or Series 33 CPU and I/O devices other than terminals. Each GIC controls a Hewlett-Packard Interface Bus (HP-IB) and translates I/O commands from the CPU into the proper HP-IB protocol. Nearly all transactions with I/O devices are accomplished without software interrupts because I/O is achieved with microcoded channel programs. Once the GIC takes control over I/O operations, device control and data flow are normally carried to completion with no software intervention and without altering the system environment.

The General I/O Channel contains Direct Memory Access (DMA) hardware which allows large records of data to be transferred at the maximum speed of the HP-IB (1 Mb). The channel microcode enables the device and then initializes the DMA hardware on the GIC. After initial addressing of a device to talk or listen, the CPU relinquishes control of the Inter-Module Bus and allows the GIC to perform its function through DMA operation. During this time the GIC becomes the master of the Inter-Module Bus and memory and controls traffic flow. On a read operation the DMA hardware will read the bytes, pack them into words and place them directly into memory, all without assistance from the CPU. The CPU is free to service other devices while DMA is in progress. Upon completion of a DMA transfer, the GIC returns to a slave condition and awaits the next operation.

Features

- 1 Mbyte bandwidth
- Provides interface for up to 8 device controllers
- Direct Memory Access (DMA)

Availability

- Series 30
- Series 33

Flexible Disc Drive

HEWLETT hp PACKARD

Model 7902A

The HP 7902 is a 1-megabyte data storage system employing a flexible disc. As an integral part of the HP 3000 Series 30 and Series 33 computer systems, the HP 7902 can be used for data storage, software distribution, data transfer, diagnostic loading, and system backup.

Features

- 1.18 Mbyte of data storage
- Double density, dual-sided data recording
- Microprocessor controlled
- Automatic self-test

The flexible medium used in the HP 7902 is a flexible disc, commonly called a diskette. The flexible disc is 20 centimeters (7.9 inches) in diameter and has a 3.8 centimeter (1.5 inch) hole for alignment on the spindle of the disc drive. The disc is enclosed in a protective jacket with a slot for head access to the recording surface. Both sides of the flexible disc are used for data storage.

The controller contains a microprocessor, a processor-to-HP-IB interface, read-only memory (ROM), random-access memory (RAM), and the associated logic necessary to provide an interface between the disc drive and the HP-IB interface channel. The microprocessor handles data and commands directly at the byte level, eliminating the need for direct-memory access (DMA) hardware.

The controller also contains an extensive self-test capability, including options for reading from an already formatted flexible disc and reading and writing on a previously unformatted disc. All self-test functions except reading from a previously formatted disc may be initiated from the HP 3000 Series 30 or Series 33 self-test or by a manual switch on the flexible disc controller board. The preformatted read self-test is switch initiated. The controller performs a subset of self-test each time power is applied to the HP 7902. This subset does not include reading or writing on the flexible disc. Self-test results are available as a four-bit binary word displayed on an LED array mounted on the controller board, or as two bytes of status information which can be read by the Series 30 or Series 33 Customer Engineer during the system self-test.

SPECIFICATIONS

Seek time

Track-to-track: 3 ms plus 15 ms settling

Average seek: 90 ms

Average access time: 174 ms

Maximum data access: 410 ms

Capacity

256 bytes per sector

30 sectors per track

77 tracks per surface

2 surfaces per diskette

1.18 Mbytes per diskette

1.03 Mbytes available to the user

Rotation

Speed: 360 rpm ± 3.5%

Average rotational delay: 88 ms

Data transfer rate

Bits per second: 184,000 (average)

Kilobytes per second: 21 (average)

Media life

Revolutions: 3,000,000 minimum on any track, heads loaded

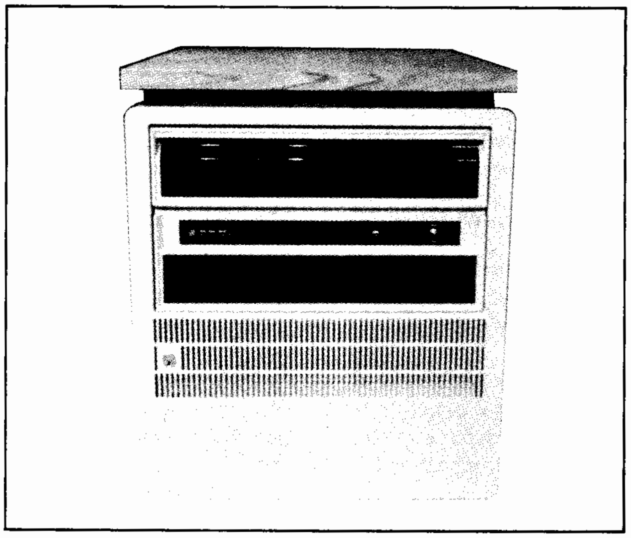
Insertions: 30,000 minimum

Availability

- Series 30
- Series 33

Disc Drives

Model 7906



The HP 7906 is a cartridge disc drive that combines convenient storage media, high performance, and high reliability. The removable cartridge allows duplication of the data on the fixed platter for backup or archival storage.

Features

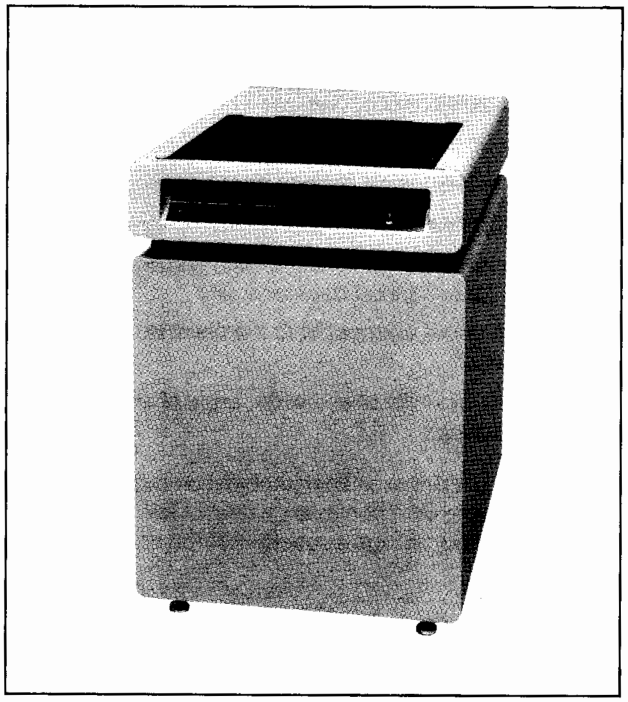
- 19.6 Mbytes of total HP formatted storage capacity
- 9.8 Mbytes of removable HP formatted storage capacity available to the user in a convenient, inexpensive, front loading disc cartridge
- 9.8 Mbytes of fixed HP formatted storage capacity available to the user
- Can be configured on the same disc controller as 7920 and 7925 disc drives

The microprocessor based controller provides advanced features of error correction and automatic alternate track switching. Up to eight HP 7906 disc drives can be supported by a single controller, or they may be mixed with HP 7920 and HP 7925 discs for a total of eight drives per controller.

Availability

- Series 30
- Series 33

Models 7920, 7925



The HP 7920 family of disc drives provides exceptionally high performance, high reliability mass storage of formatted information. Their extremely fast data access rate using track-follower technology (25 ms average random seek time, 5 ms average track-to-track seek time) is among the fastest in the industry.

Features

- 50 Mbytes of HP formatted storage capacity with 7920 drive
- 120 Mbytes of HP formatted storage capacity with 7925 drive
- Easily removable multiple platter disc packs
- Protective upper and lower platters in each pack

Designed for low noise operation and ease of service, the 7920 family of drives are equipped with a compact, removable disc pack which consists of multiple platters. The upper and lower platters are protective. Interior platters have two surfaces each. One interior surface on the disc is used for servo information to control the rapid and precise positioning of the actuator mechanism. Incorporation of positioning information on the disc pack provides for accurate track positioning over wide temperature variations. All other interior surfaces are used for formatted data storage.

Availability

- Series 30
- Series 33
- Series II
- Series III

Models 7906, 7920, 7925 common features

- Fast-access, track-following technology
- Advanced serviceability features
- Constant spindle speed, independent of line frequency
- Extended maintenance periods through use of prefilter
- Exceptionally fast stop/startup speed
- Microprocessor-based controller with hardware error correction for enhanced data reliability
- Up to eight drives connected to the common disc controller
- Exceptional reliability over a wide range of environmental conditions

Since up to eight drives in any combination may be connected to a single disc controller, a total of 960 megabytes of formatted storage capacity may be accessed per controller.

Operator switches and indicators

The outside operator's panel has five backlit indicators: Unit Select, Drive Ready, Read Only, Door Unlocked, and Drive Fault. There is one switch: Run/Stop.

On the inside operator's panel a group of eight LED indicators light if the internal fault detection circuitry detects a fault condition. This advanced serviceability feature facilitates trouble shooting and reduces the time to diagnose and repair failures.

In the event of a power failure, heads are retracted and carriage locked using energy from the filter capacitors, supplemented by the spindle motor acting as a generator.

Automatic error correction and detection

The error correction code (ECC) hardware increases data reliability and system availability by reducing the effects of media errors.

The ECC hardware and algorithm are together capable of correcting one single burst data error per sector, if the error is less than or equal to 32 bits. Every single burst data error greater than 32 bits but less than or equal to 48 bits will be detected without sending correcting information. For burst errors greater than 48 bits, 99.9999% are detected.

If the controller detects an error within a sector, it notifies the CPU at the end of that sector. The CPU then requests the location (displacement) of the error within the sector and three words of mask which are used to correct the record now in CPU memory. The controller calculates these masks from information accumulated in special registers. The three words are merged with data in CPU main memory to obtain a corrected record.

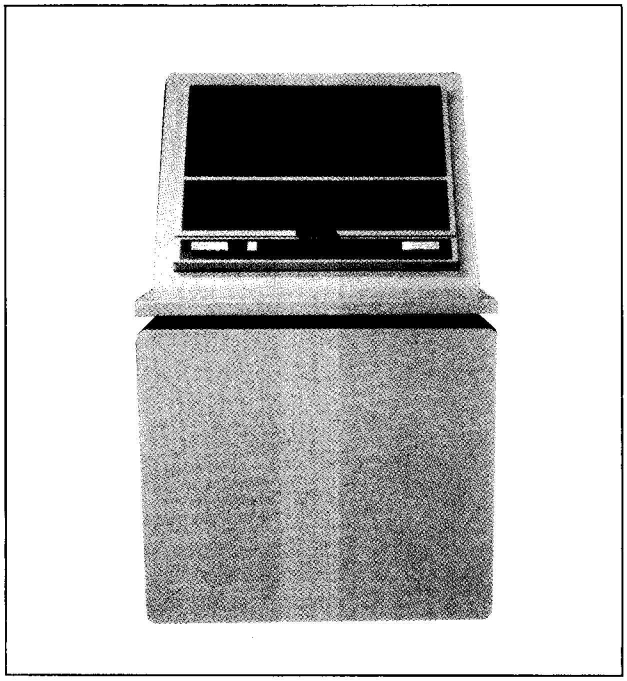
SPECIFICATIONS

	7906	7920	7925
Dimensions			
Height:	28.25 in (71.8 cm)	32.5 in (82.5 cm)	32.5 in (82.5 cm)
Width:	21.78 in (55.3 cm)	19.65 in (50 cm)	19.65 in (50 cm)
Depth:	31.13 in (79.1 cm)	32 in (81.3 cm)	32 in (81.3 cm)
Shipping weight:	340 lb (154 kg)	415 lb (188 kg)	420 lb (191 kg)
Seek time			
Track-to-Track:	5 ms max	5 ms max	5 ms max
Average random:	25 ms	25 ms	25 ms
Maximum stroke:	45 ms	45 ms	45 ms
Capacity			
	19.6 Mbytes/drive 2 data surfaces/drive 800 data tracks/surface 48 sectors/track 256 bytes/sector	50 Mbytes/drive 5 data surfaces/drive 815 data tracks/surface 48 sectors/track 256 bytes/sector	120 Mbytes/drive 9 data surfaces/drive 815 data tracks/surface 64 sectors/track 256 bytes/sector
Rotation:			
Speed:	3600 rpm	3600 rpm	2700 rpm
Avg. rotational delay:	8.3 ms	8.3 ms	11.1 ms
Data transfer rate			
Bits/second:	7,500,000	7,500,000	7,500,000
k-bytes/second:	937.5	937.5	937.5

Digital Magnetic Tape Drives

HEWLETT PACKARD

7970B, 7970E



Hewlett-Packard Digital Magnetic Tape Units are high performance, reliable magnetic tape drives for use in an HP 3000 computer system. IBM compatible NRZI recording mode is used at a density of 800 bpi (7970B). High packing density and data transfer rates are achieved by using ANSI-compatible 1600 bpi phase encoded data electronics (7970E). Data written on any IBM or ANSI-compatible equipment can be read. Four tape drives can be operated from a single controller.

Features

- Fast data transfer—up to 36k bytes/sec (NRZI)—up to 72k bytes/sec (phase encoded)
- 9-track configuration
- 800 bpi, NRZI electronics (7970B)
- 1600 bpi, phase encoded data electronics (7970E)
- 45 ips read/write, 160 ips rewind/fast forward
- Dynamic braking
- Up to 10½ inch (26.7 cm) reels
- IBM/ANSI compatible

Reel motors provide direct drive, eliminating troublesome belts and pulleys. Tape tensioning is performed by photo-resistive controlled tension arms, eliminating the need for vacuum system components. Head assemblies consist of read stack, write stack and full width erase head.

SPECIFICATIONS

Number of tracks: Nine

Read/write speed

45 ips (114 cm/s)

Density

800 bits/cpi, NRZI electronics (7070B)

1600 bits/cpi, phase encoded electronics (7970E)

Data transfer rate

36,000 characters per second maximum. NRZI electronics (7970B)

72,000 characters per second maximum, phase encoded electronics (7970E)

Write enable

Supply reel write enable ring and switch. Ring removal precludes writing.

Reel diameter

Up to 10½ inches (26.7 cm)

Tape (Computer Grade)

Width: 0.5 inches (12.7 mm)

Thickness: 1.5 mils (0.038 mm)

(continued)

Rewind speed

160 ips (406 cm/s)

Start/stop times

8.33 ms (read-after-write) at 45 ips (114 cm/s)

EOT and BOT reflective strip detection

Photoelectric, IBM compatible

Operator control panel

Reset Switch: Stops tape travel in any mode and returns unit to local control

Rewind Switch: Rewinds tape at 160 ips (114 cm/s)

On-Line Switch: Places unit under remote control

Load Switch: Initiates loadpoint (BOT) search

Write Enable Indicator: Illuminated when write enable ring is installed on the supply reel

Unit select switch

Physical characteristics

The 7970B/E is packaged in a stand alone low-boy cabinet. The shipping weight is 190 lbs (86.2 kg).

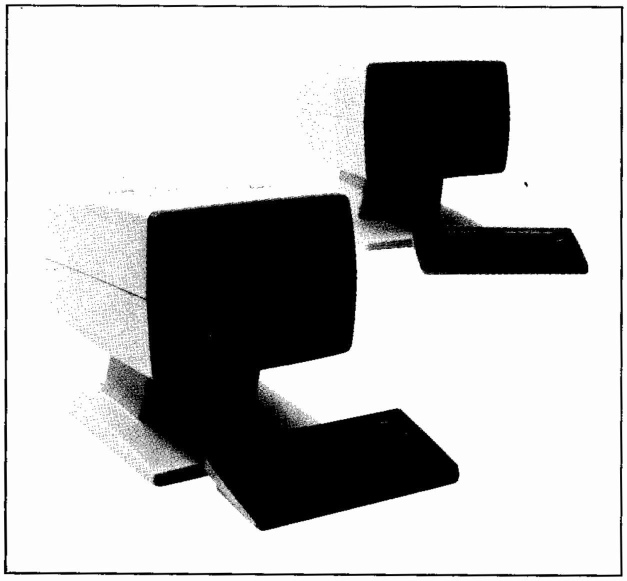
Availability

- Series 30 (7970E only)
- Series 33 (7970E only)
- Series II
- Series III

Interactive Display Terminals

HEWLETT hp PACKARD

Models 2621 A/P



The 2621A and 2621P terminals emphasize simplicity, reliability, and quality in products designed for interactive applications. The 2621P with an integral thermal printer combines the convenience of local hard copy with the speed of a video terminal.

Features

- Familiar 68-key, typewriter-style, detached keyboard
- Embedded numeric pad
- Screen-labeled control keys
- 8 pre-programmed function keys
- Soft configuration
- Full cursor positioning
- Tabulation and margin control
- High resolution CRT
- Enhanced 9 x 15 dot character cell
- Upper/lower case characters with displayable control codes
- Character-by-character underline
- Cursor column indicator
- Two full pages (48 lines) display memory
- Roll, home-up, and home-down
- Clean, quiet 120 character-per-second integral thermal printer (2621P)
- Bidirectional printing
- Upper/lower case characters
- Underline
- Excellent character definition
- Automatic data logging
- Go/no-go and diagnostic self-test

High resolution display

The 2621 displays 1,920 characters in a 24 line by 80 column format on a 6 by 8.5 inch high-resolution display. The 9 x 15 dot character cell allows precise formation of complex symbols with wide line and character separation. These features combine to produce a bright, easy to read display for improved operator satisfaction.

Hard Copy

In addition to the features of the 2621A, the 2621P Interactive Terminal contains an integral hard-copy unit. Upper/lower case letters and underline are printed on thermal paper at 120 characters-per-second. This self-contained unit is ideal for the occasional walk-away copy. The printer is accessed by user application programs

Ease of use

The familiar typewriter-style keyboard of the 2621 presents a friendly interface designed to minimize training time. Eight screen-labeled control keys provide quick access to editing, configuration, self-test, printer control, and other functions. Rapid numeric entry is assisted by an embedded numeric pad. The 8 screen-labeled keys also double as special function keys which can be used to call computer-resident routines. Traditional mechanical switches selecting baud rate, parity, and various communication parameters are replaced by soft configuration using non-volatile memory which can be displayed and changed easily from the keyboard.

SPECIFICATIONS

General

Screen size: 150 mm (6 inches) x 215 mm (8.5 inches)

Screen capacity: 24 lines x 80 columns (1,920 characters)

Character generation: 7 x 9 enhanced dot matrix: 9 x 15 dot character cell; non-interlaced raster scan

Character size: 2.4 mm (.094 inches) x 3.5 mm (138 inches)

Character set: Upper/lower case, displayable control codes

Cursor: Blinking-underline

Display enhancements: Underline

Memory: 4096 bytes (two full pages)

256 bytes, non-volatile (battery powered)

Keyboard: Full ASCII code keyboard; 8 screen labeled keys; cursor controls; embedded numeric pad; auto-repeat; N-key roll-over; detached; with 1.2M (4 ft.) cable.

(continued)

Data communications

Data rate: 110, 150, 300, 1200, 2400, 4800*, 9600* baud, (110 selects two stop bits).

Asynchronous interface: EIA standard RS232C (fully compatible with Bell 103A modems) CCITT V.24. Cable is required.

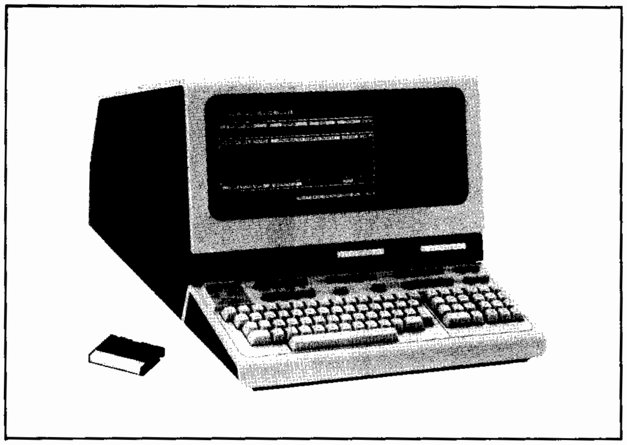
Transmission modes: Full duplex, asynchronous

Operating modes: On-line; off-line; character, line

Parity: Selectable; even, odd, zero, one

Series 30 and 33 only

Models 2640B, 2640C, 2640N, 2640S, 2641A, 2645A, 2645N, 2645R, 2645S, 2647A, 2648A



The HP 2640 series of interactive display terminals brings to your HP 3000 an unparalleled set of features designed to complement your system in all areas of your organization.

Features

- Enhanced high-resolution display
- Plug-in character sets
- Dynamically allocated memory
- Character/block mode
- Self-test
- Full editing capability
- Multi-task keyboard
- Off-screen storage with scrolling capability
- Programmable protected fields
- Inverse video for highlighting; and optional blinking, underline, half-bright
- Cursor addressability and positioning control tabulation
- Hard copy interface
- Multipoint asynchronous and synchronous communications capability

Enhanced high-resolution display

The terminals have a 5 inch by 10 inch rectangular display providing a 1,920 character capacity in 24 lines of 80 characters per line. The characters are formed by a 7 x 9 dot matrix generated in a 9 x 15 dot character cell. The high resolution of the 7 x 9 dot matrix is enhanced by half dot shifting for precise character definition, and by the use of the enlarged character cell for wide character and line separation, underlining, line descenders, and inverse video. These display features are engineered to increase clarity and ease sessions at the terminal.

(continued)

Plug-in character sets

Recognizing the demand for terminals that speak many languages and fill diverse sets of needs, the terminals have the capacity to include up to four 128-character sets resident concurrently in the terminal. Adjacent characters on the display may be from any of the four character sets and are available with the optional underline, blinking and half-bright feature.

Dynamically allocated memory

Because of the efficient linking memory organization (transparent to you) spaces to the right of the last character typed on a line are normally not stored in memory. Consequently, the basic terminal equipped with 4K characters of display memory can store from 32 to 200 lines of information dependent on line length. Optional memory can expand this line capacity to a maximum of over 400 lines of information. Lines are viewed 24 at a time by using the roll up, roll down, next page, and previous page keys.

Pop-in modularity and expandability

The modular computer-like construction is designed for ease of service. Digital electronics are contained on printed-circuit cards that can be exchanged within the terminal; up to 14 cards can be accommodated to allow a choice of options. The combination of microprogramming and modularity means that the terminals can be expanded as new technologies and devices become available.

Microprocessor controlled

The operating characteristics of the terminal are controlled through firmware. The terminal's microprocessor manages memory allocation, data communications, keyboard scanning, and display control. This microprocessor implementation and the use of a single bus architecture yield a terminal utilizing electronics and mechanics with a wide range of capabilities and potential for future enhancements.

Character/block mode

The terminals will operate character-by-character as a completely interactive terminal or are capable of operating on a block at a time. Text can be composed and edited locally, thus allowing you to verify and correct data before transmission to the CPU. Editing and CPU connect times are significantly reduced by such features as character or line insert and delete, cursor sensing and positioning control, programmable protected fields, off-screen storage with scrolling and page select capability, tabulation, transparent display control codes, 8 special function keys for user defined routines and a positionable memory protect.

Self-test

The terminals have been engineered for high reliability, ease of maintenance, testing, and rapid repair when needed. By depressing the TEST button on the keyboard you receive a Go/No-Go indication from results of a memory test, firmware test, and display verification.

Hard-copy interface

The terminals accommodate most RS232C-compatible serial printers or any HP-manufactured printer which uses the HP parallel interface. The serial printers are connected via the 13250A interface card, while the parallel printers use the 13238A duplex register interface card. Commands to print data can be initiated locally from the terminal keyboard or remotely from the CPU under control of a user's program. The MPE operating system does not recognize hardcopy devices connected to the terminals. All control of hardcopy devices is assumed by your program.

Display enhancements

You can highlight specific fields by using any of the 16 possible combinations of inverse video, blinking, underline and half-bright to emphasize areas of the display.

Optional and standard features

Fully integrated mass storage (optional on the 2641, 45, 48; standard on the 2647); Benefits of mass storage: Many operations normally requiring connection to a computer system can now be done off-line with the terminal. Optional dual in-board cartridge tape units allow batching of information, and add extensive stand-alone capabilties which can significantly reduce user time, conserve both computer and communications resources; provide a tape backup; and, very importantly, allow the terminal to keep on working even when a computer is unavailable.

Mini-cartridge: Highly reliable, interchangeable mini-cartridges each provide the capacity for many hours of typing, up to 110,000 characters of storage formatted in variable length records and files. The tape units feature rapid data transfer and bi-directional high-speed search to access any file in seconds. The mini-cartridge is ideally suited for storing data, forms, programs or text, and is an excellent substitute for paper tape. The HP 3000 FCOPY utility may be used to transfer files to and from mini-cartridges to other HP 3000 peripherals.

Hardcopy interface: Certain peripherals may be run under direct terminal control. The data sheets for each terminal specify the number and models of allowable drivers.

(continued)

User-defined soft keys (2641, 45, 47, 48): Each of the eight special function keys can be easily used to issue a user-defined string of up to 80 characters or several control sequences stored in the 2645A. This feature allows the keyboard to adapt to specialized applications, and can considerably simplify use of the keyboard and result in greater efficiency—each soft key performs the operations of several key sequences. For example, the soft keys could issue frequently used programming sequences, search for files, aid forms construction for data entry, dynamically configure the terminal, or issue instructions to the operator, computer or both.

Character sets

APL: With the 2641, the APL basic set and APL overstrike set are both available for development and display of APL programs. The clarity and visual representation of the characters are facilitated by the 9 by 15 dot character cell. The basic set is comprised of the 96 APL graphics, 64 overstrike graphics and includes the ability to display control codes which help in debugging programs.

Also standard in the 2641A is a 64 character Roman set which is optionally expandable to include lowercase alphacapability. This second set allows the terminal to be switched from APL to the ASCII mode where the 2641A can then double as a fully Teletype-compatible terminal.

Math: The optional Math set is available to include equations right in the text that's composed on the screen. This feature helps to impart realism and assists in understanding technical material.

Line drawing: The optional Line Drawing set assists users in defining forms that appear like those normally filled out in your company's day-to-day operations. Now operator errors can be minimized due to the familiarity associated with the constructed forms. Application programs are responsible for constructing forms from the supplied segments.

Large character set: With this option, characters approximately ½ in. tall can be constructed. This is useful for warehousing operations where, for example, fork-lift drivers may have to read computer derived instructions from a distance. Application programs are responsible for constructing characters from supplied segments.

Norwegian: The 2640N and 2645N have the Danish/Norwegian set in place of the standard uppercase set. The non-standard characters must be handled by application programs.

Cyrillic: The 2640C has 128 character Cyrillic set in addition to standard uppercase set. The non-standard characters must be handled by application programs.

Swedish: The 2640S and 2645S have 64 character Swedish (Finnish) set in place of standard uppercase set. The non-standard characters must be handled by application programs.

Arabic: The 2645R has a combined 64 character Roman/31 character Arabic character set in place of the standard uppercase set. The non-standard characters must be handled by application programs.

SPECIFICATIONS

General

Screen size: 5″ x 10″ (127 x 254 mm)

Screen capacity: 24 lines x 80 columns (1,920 characters)

Character generation: 7 x 9 enhanced dot matrix; 9 x 15 dot character cell; non-interlaced raster scan

Character size: 0.097″ x 0.125″ (2.46 x 3.175 mm)

Cursor: Blinking-underline

Display modes: White on black; black on white (inverse video) blinking, half-bright, underline

Refresh rate: 60 Hz (50 Hz optional)

Tube phosphor: P4

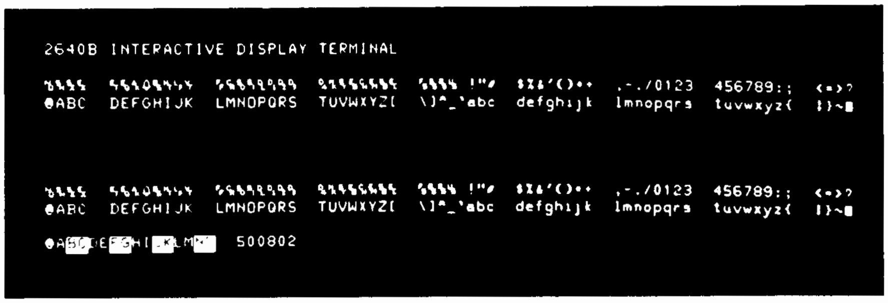
Implosion protection: Bonded implosion panel

Memory: MOS, ROM: 22k bytes (program); RAM; standard 4096 bytes; 12 kilobytes maximum (16k including maximum data communications buffer)

Keyboard: Detachable, additional control and editing keys; ten-key numeric pad; cursor pad; multispeed auto-repeat, N-key roll-over; 4 ft. cable (1.22 m)

Cartridge tape (optional): Two mechanisms

(continued)



```
2640B INTERACTIVE DISPLAY TERMINAL

                                              !"#  $%&'()*+  ,-./0123  456789:;  <=>?
@ABC  DEFGHIJK  LMNOPQRS  TUVWXYZ[  \]^_`abc  defghijk  lmnopqrs  tuvwxyz{  |}~


                                              !"#  $%&'()*+  ,-./0123  456789:;  <=>?
@ABC  DEFGHIJK  LMNOPQRS  TUVWXYZ[  \]^_`abc  defghijk  lmnopqrs  tuvwxyz{  |}~

@ABCDEFGHIJKLMN   500802
```

Actual photograph of 2640B display

Read/write speed: 10 ips

Search/rewind speed: 60 ips

Recording: 800 bpi

Mini cartridge: 110 kilobyte capacity (maximum per cartridge)

Data communications

Data rate: 110, 150, 300, 1200, 2400 baud (4800 and 9600 baud also available on Models 2645A, 2647A, and 2648), and external switch selectable (110 selects two stop bits).

Standard asynchronous communications interface: EIA standard RS232C; compatible with Bell 103A modems, Bell 202S†/T modems with reverse channel, and Vadic VA3400 series modems. Bell 212 is supported on the Series 30 and Series 33.

Transmission modes: Full or half duplex, asynchronous
Operating modes: On-line; off-line; character; block
Parity: Switch selectable; even, odd, none

†The use of 202S modems is not recommended due to performance reasons.

Terminals at a Glance

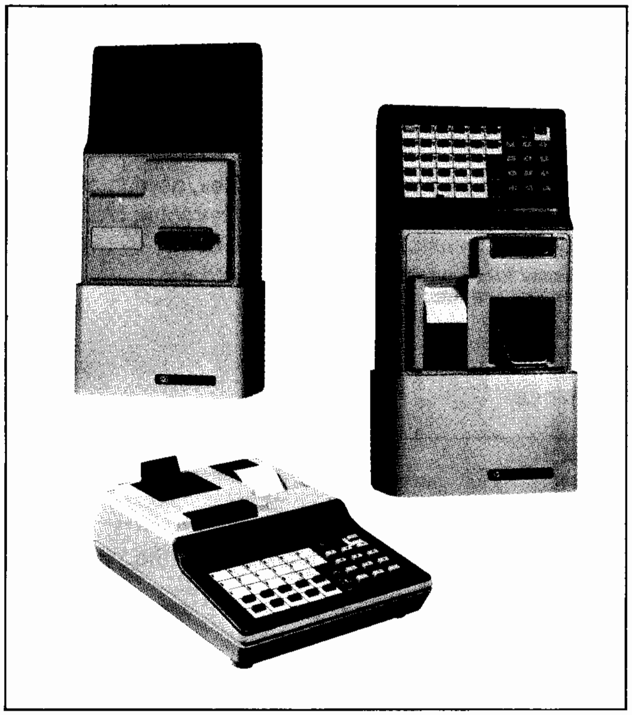
	FEATURES	2621A 2621P	2640B	2640N 2640S 2640C	2645A	2645N 2645S 2645R	2641A	2647A	2648A
Base Level Features	Two Tape Cartridge Drives	—	—	—	Opt.	Opt.	Opt.	Std.	Opt.
	Inverse Video	—	Std.	Std.	Std.	Std.	Std.	Std.	Std.
	Underline, Blinking, ½ Bright Line Drawing	Std. Note 3	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.
	Soft Keys	Std.	—	—	Std.	Std.	Std.	Std.	Std.
	Lower Case	Std.	Std.	Opt.	Std.	Opt.	Opt.	Std.	Std.
	Maximum Memory	48 lines	8kb	8kb	12kb	12kb	12kb	6kb	12kb
	Math Character Set	—	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.
	Large Character Set	—	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.
	Roman Character Set	Std.	Std.	—	Std.	—	Std.	Std.	Std.
'International' Character Set Replaces 'Roman' to Produce a Unilingual International Terminal	Norwegian/Danish	—	—	N Model	—	N Model	—	—	—
	Swedish/Finnish	—	—	S Model	—	S Model	—	—	—
	Arabic	—	—	—	—	R Model	—	—	—
'Roman' plus 'International' Character Set Produces a Bilingual Terminal	Cyrillic	—	—	C Model	—	—	—	—	—
Advanced Features	APL Character Set	—	—	—	—	—	Std.	—	—
	Graphics	—	—	—	—	—	—	Note 1	Note 1
	Hardcopy Interface	Note 4	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.	Opt.
Availability	Series II and III	Yes	Yes	Yes	Yes	Yes	Yes	Note 2	Note 2
	Series 30 and 33	Yes	Yes	Yes	Yes	Yes	Note 2	Note 2	Note 2

Note 1. The 2647A and 2648A are supported on the HP 3000 as a 2645A.
Note 2. Supported as 2645A.
Note 3. Underline only.
Note 4. 2621P has an integral thermal printer as standard. No other hardcopy interface offered.

Data Capture Terminals

HEWLETT hp PACKARD

Models 3075A, 3076A, 3077A



The HP 3075A/3076A/3077A family of data capture terminals fulfills a wide range of data collection requirements in industry; in applications ranging from stock control to time data reporting.

Features

- Modularity, choice of:
  —numeric or alphanumeric display
  —numeric or alphanumeric keyboard
  —multifunction reader
  —type V badge reader
  —alphanumeric strip printer
- User definable prompting lights
- User definable special function keys
- Time clock (HP 3077A only)
- Built-in self test
- RS232C/CCITT V.24
- High speed data transmission
  —up to 9600 baud

HP 3075A/3076A data capture terminals

The HP 3075A Desktop terminal and HP 3076A Wall Mounting terminal (with Cradle) are both workstations equipped with user-definable special function keys and prompting lights. These can be individually defined for specific tasks under programmatic control.

The user-definable keys and lights are labelled with their specific functions. This enables people with little or no experience of using computer terminals to operate these terminals with no special training.

The terminals' modular construction and wide range of options enable them to be built in over 50 different combinations. Thus each terminal can be configured to suit the required application.

"Primarily designed for use in manufacturing environments" does not mean they are restricted to the shop floor. They are also perfectly adapted for applications in finance, order processing or any department with a data processing requirement.

HP 3077A time reporting terminal

The HP 3077A is a time reporting terminal equipped with a large time display and a type V badge reader (or optionally a multifunction reader). It can be used, for example, to register personnel arrival and departure or to control access to restricted areas (using a relay built into the wall mounting cradle).

Data communications

System support is for point-to-point connection hardwired or via Bell 103A or 212A type full duplex modems. Multipoint support, hardwired, is available via the MTS/3000 communication subsystem.*

Keyboard

Standard: Numeric only

Optional: Includes both numeric and alphanumeric

Display

Standard: 15 positions, numeric

Optional: 24 positions, numeric and alphanumeric

Other optional features.

Multifunction reader: Reads punched plastic badges (industry type III), 80 column punched cards and optical mark forms.

Type V badge reader: Reads punched plastic badges with or without clips.

Alphanumeric printer: 40 lpm with 20 characters per line. Thermal paper.

Availability

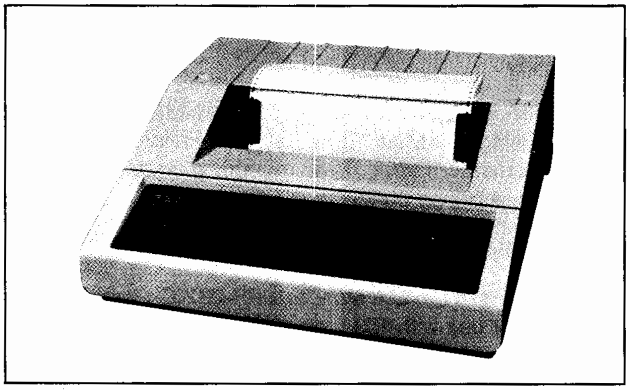
- Series 30
- Series 33
- Series II
- Series III

**MTS/3000 is available on the Series II and III only.*

Printing Terminal

HEWLETT hp PACKARD

Model 2635A



The HP 2635A is a dot matrix hard copy terminal featuring high throughput and a wide range of user-definable printing functions. Utilizing Hewlett-Packard's advanced SOS microprocessor technology, the HP 2635A can serve either as the HP 3000 Series III system console or as an optional terminal on any HP 3000 system.

Features

- High throughput
- High resolution dot matrix characters
- 128 USASCII character set
- Long life print head
- Cartridge ribbon
- Single or multipart forms
- 8 channel fixed VFC
- Special user features

  variable horizontal pitch
  variable vertical pitch
  horizontal tabulation
  display functions
  self test
  paper out detection
  auto line feed
  auto-underline
  selectable view mode

- Typewriter style keyboard
- Numeric keypad
- RS232C interface (point-to-point only)

High throughput

The HP 2635A has a print speed of 180 characters per second. The HP 2635A optimizes throughput by printing bidirectionally, suppressing leading and trailing blanks, skipping over embedded blanks, and returning the carriage at high speed. The HP 2635A utilizes an HP CMOS/SOS microprocessor to optimize printing functions, data manipulation, and other control functions.

High resolution dot matrix

The HP 2635A utlizes a high resolution seven column by nine row dot matrix character cell for crisp, clear character formation. The nine high matrix allows true descenders, commas, semi-colons, and underline. Dot matrix is especially well suited for multipart forms; since each dot is formed with equal intensity, the sixth copy has surprisingly good character resolution.

Long life print head

The nine-wire print head is rated at 130 million characters. When replacement is required, this self-aligning head is easily replaced using the hex key supplied with the terminal.

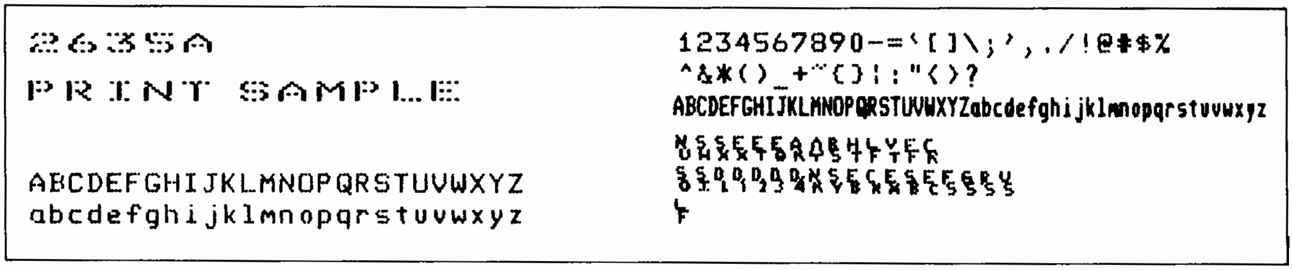
Cartridge ribbon

The cartridge ribbon has a long life of more than 10 million characters (about 3 months of normal use). This is made possible by an internal mobius loop design that permits the use of the entire ribbon area. When a ribbon change is necessary, it is a quick, simple and clean task.

Single or multipart forms

Up to 6 part forms (up to 0.43mm thick) may be accepted by the HP 2635A. Due to a unique head design, adjustment for forms thickness variations between 0.08mm (.003") and 0.43mm (.017") is usually not required. A lever adjustment on the print head carriage is provided to allow optimization of print quality on various forms.

(continued)



```
2635A
PRINT SAMPLE

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
```

```
1234567890-=`[]\;',./!@#$%
^&*()_+~{}|:"<>?
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
N S S E E E A B B H L V F C
U H X X T Q K L S T F T F R
S S D D D D D N S E C E S E F G R U
O I L 1 2 3 4 K Y B N M B C S S S S
L
F
```

8 Channel fixed VFC

Vertical forms control is provided through an eight channel fixed electronic VFC. There is no tape or paper tape reader to worry about; the VFC is located in read only memory (ROM). The eight channels provide the following forms advance capability:

		Line # ^{c} *6 LPi*
■ Channel 1	top of next page	0
■ Channel 2	bottom of current page	59
■ Channel 3	single space	0, 1, 2, ... 59
■ Channel 4	next double space	0, 2, 4, ... 58
■ Channel 5	next triple space	0, 3, 6, ... 57
■ Channel 6	next half page	0, 30
■ Channel 7	next quarter page	0, 15, 30, 45
■ Channel 8	next tenth space	0, 10, 20, ... 50

Note: "Page" is defined to be 11 inches. "Space" is defined to be 1/6 inch, unless 8 lpi mode is being used, in which case "space" is defined as 1/8 inch.

Special user features

By entering at the keyboard specific keystrokes of control sequences or embedding these sequences in the data stream, the user can customize 2635A output.

- Horizontal printing in any of three modes: compressed (16.7 characters per inch); normal (10 cpi); expanded (5 cpi)

  Normal mode is the standard on most terminals and will provide 136 characters per line on 14⅞ inch wide paper. Compressed printing is useful for printing long lines (227 characters per line on 14⅞ inch paper) or saving paper (132 characters per line on 8½ inch wide paper). Expanded mode creates characters which are useful for titles and headings. The three different modes may be intermixed on a single line
- Auto-underlining in all print modes provides emphasis to key words and phrases in your output
- Vertical spacing is variable at vertical pitches of 1, 2, 3, 4, 6, 8, or 12 lines per inch
- A view mode allows the user to see the last character printed by moving the print head to the right after a keyboard entry
- In addition to control sequence operation, certain special features may also be triggered by setting buttons or switches on the 2635A
- Horizontal tabs may be set, cleared, and accessed in any of 227 print positions
- In display functions mode, embedded control sequences are printed rather than executed. Carriage return (control M), on-line (escape n), off-line (escape o), and display functions off (escape Z) are exceptions; they are displayed and executed

SPECIFICATIONS

Reliability

The 2635A is recommended for 1 million characters per day, resulting in a Mean Time Between Failures (MTBF) of approximately 1 year. Basic monthly maintenance charges are based on this level of usage.

RS232C interface

The interface is the EIA standard RS-232-C asynchronous interface for use with 103 type modems.

Data Rate: 110, 150, 300, 1200, 2400 bits per second and external switch selectable; (110 selects two stop bits.)

Transmission Mode: full duplex

Parity: odd,* even, or none; switch selectable

Connector: 25 position, similar to Cinch DBM (D subminiature socket)

Power requirements

Input voltage: 100, 120, 220, 240 volts AC (+10%, −12%) selectable from rear panel; 48-66 Hz

Power consumption: 140 VA max nonprinting, 265 VA max printing

Environmental conditions

Temperature, free space ambient: −40° to 75°C (−40° to 167°F) non-operating, 0° to 55°C (32° to 131°F) operating

Humidity: 5% to 95% RH (non-condensing) excluding media

Physical specifications

595mm (23.1″)D x 640mm (25.2″)W x 215mm (8.5″)H

23.5 kg (56 lb) net weight (without pedestal)

81 kg (72 lb) ship. weight (without pedestal)

Availability

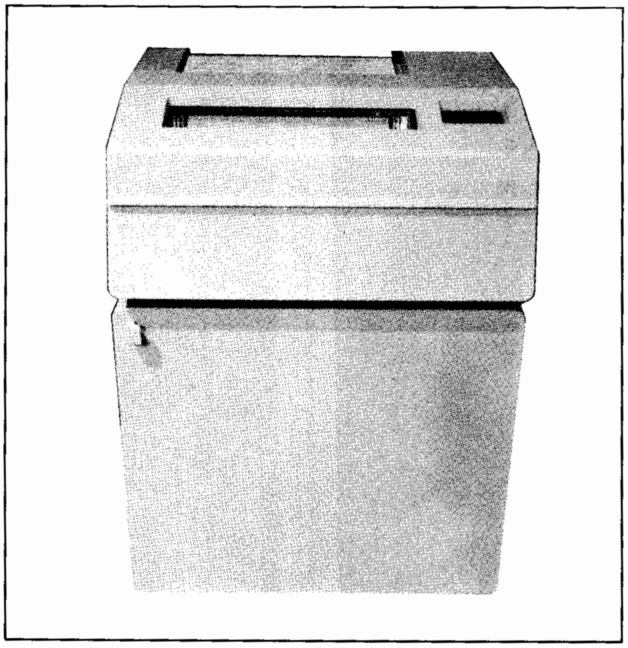
- Series 30
- Series 33
- Series II
- Series III

*Series II/III only

Dot Matrix Line Printer

HEWLETT PACKARD

Model 2608A



The HP 2608A is a low cost, highly reliable, medium speed dot matrix line printer designed for use in most computer applications. Microprocessor control provides increased reliability and added capabilities such as self test/self diagnostic mode.

Features

- Prints 132 columns of upper case ASCII at 400 lines per minute
- Prints 132 columns of lower case ASCII at 340 lines per minute
- 16-channel electronic Vertical Forms Control
- Single through 6 part forms
- Long life cartridge ribbon
- 6 or 8 lines per inch
- High resolution characters
- 5 x 9 dot matrix for true lower case and descenders (commas, etc.)

The HP 2608A uses a high resolution dot matrix character cell for crisp, clear character formation. Dot matrix printing is especially well suited for the printing of multipart forms. Impact variations, which cause embossing, smearing, and light or partially formed letters, do not occur with the 2608A dot matrix process. Each dot is formed with equal force, thus eliminating "ghosting" and "too light" characters which occur with full character impact printing. Up to 6-part forms (0.61 mm maximum pack thickness) may be used with the HP 2608A. To maintain best print quality when changing pack thickness, the platen-to-hammer gap can be easily adjusted by the operator with a platen adjustment knob. To ensure smooth paper feed, forms are loaded through a bottom paper feed slot. The use of 8-pin forms tractors allows edge perforated, fan fold forms fo be advanced quickly and reliably.

The HP 2608A will print the upper case USACII character set at a speed of 400 lines per minute. This print speed, coupled with a vertical slew rate of 14 inches per second and a single line advance rate of 15 milliseconds, gives the HP 2608A its high throughput.

Reliability

The HP 2608A has been designed with reliability in mind. The printing mechanism has few moving parts, operates virtually without friction, and requires minimum maintenance. In addition, the printer is microprocessor controlled to enable incorporation of diagnostics in a self test/self diagnostic mode.

The HP 2608A provides reliable and long lasting service for most applications. Duty cycle restrictions have been replaced by a multi-level service contract, which allows the 2608A to be used as much as required for the application.

The basic usage levels are:

Up to 1.65 million lines per month

Up to 3.74 million lines per month

Over 3.74 million lines per month

It is suggested that those applications requiring over 4 million lines per month use multiple 2608A printers.

SPECIFICATIONS

Character formation	Dot matrix (5x7, 5x9)	
Line length	Up to 132 characters	
Print speed	Lines per minute	Matrix size
	400	5x7
	340	5x9
Line feed rate (6 or 8 lines/inch)	15 ms	
Form feed rate (6 or 8 lines/inch)	14 inches/second	
Copies	1 to 6 (up to 0.61 mm pack thickness)	
Vertical format control	16 channel electronic	

(continued)

Physical

Width	679.5 mm (26.5 inches)
Depth	554.6 mm (21.8 inches)
Printer height	306.0 mm (12 inches)
Stand assembly height	736.0 mm (29 inches)
Weight	97 kg (215 lbs.)

Features supported under MPE III

- Print and space operation
- Downloadable Vertical Forms Control
- Programmatically adjustable left margin
- Programmatic selection between primary and secondary character sets

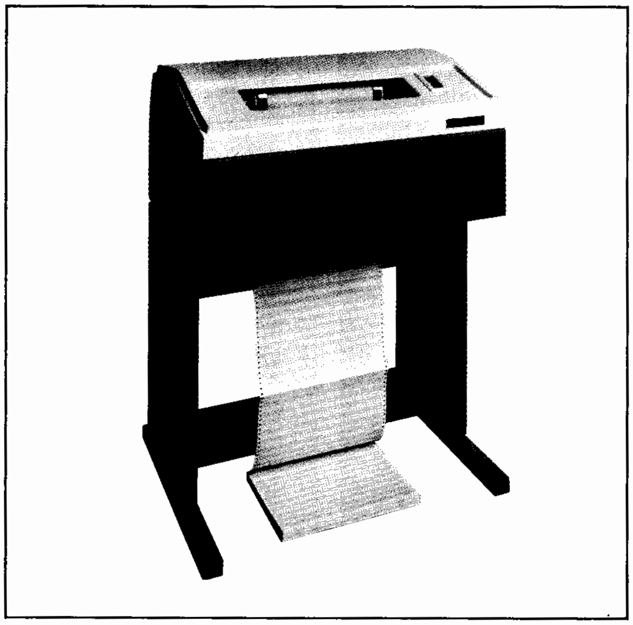
Availability

- Series 30
- Series 33
- Series II
- Series III

Low Speed Line Printer

HEWLETT hp PACKARD

Model 2613A



The HP 2613A Line Printer represents an optimized combination of high print quality, moderate speed, and low price in a drum printer. This unit is ideally suited to applications requiring a medium speed printer.

Features

- Prints 136 columns of 64 characters at 300 lines per minute
- Prints 136 columns at 240 lines per minute with 96 character set
- 12-channel vertical forms control
- Multiple copies—prints up to 6-part forms
- 64-character or 96-character ASCII available
- 6 or 8 lines per inch—operator selectable

A 12-channel Vertical Format Unit allows convenient and efficient printing in predetermined formats such as business forms. The VFU uses industry standard control tapes. Under program control the VFU may be commanded to slew to the next hole in a given channel of the tape or to slew an absolute number of lines from 0 to 15.

A switch allows the operator to choose 6 or 8 lines per inch (60 or 80 lines/page respectively). A matching VFU tape will synchronize the forms to the VFU. An ASCII 64-character set is standard with 96-character ASCII optional. Six part forms are easily handled by the HP 2613A. Forms alignment is simplified by horizontal and vertical paper alignment guides. Fine vertical paper alignment adjustments can be made while printing. A paper receptacle is provided.

Differential line drivers allow the printer to be located up to 500 feet from the computer for flexibility of installation design.

SPECIFICATIONS

Paper feed

One set of pin tractors above drum, friction paper tensioner below drum.

Line advance: 50 milliseconds
Slew rate: 20 inches (50.8 cm) per second

Paper dimensions

Standard fanfold, edge punched paper 4 inches (10.2 cm) to 16 inches (40.6 cm) wide

Paper type

Single copy, 15 lb. bond minimum weight

Multi-copy up to 6 parts, 12 lb. bond with single slot carbon

Character drum

Characters per line: 136

Character type: Open Gothic

Symbol size (typical): 0.095 inch (2.4mm) high, 0.064 inch (1.65mm) wide

Vertical format unit

Number of channels: 12

Addressing: Slew to next hole in channel "n" or slew 0 to 15 lines

Throughput data

Duty cycle: 20%

Print cycle: 40% (per page)
(425 pages/hour for heavy print periods. 681 pages/8 hour day). Refer to "A Guide to Hewlett-Packard Printers" for detailed performance data.

Printing speeds

Char. Set	Drum Speed	136 Char. Line Per Minute
64	1200	300
96	800	240

Physical characteristics

Height: 45 inches (114.5 cm)

Width: 33 inches (83.8 cm)

Depth: 22 inches (55.9 cm)

Weight: Net: 340 lb (154.2 kg)

Shipping: 365 lb (165.6 kg)

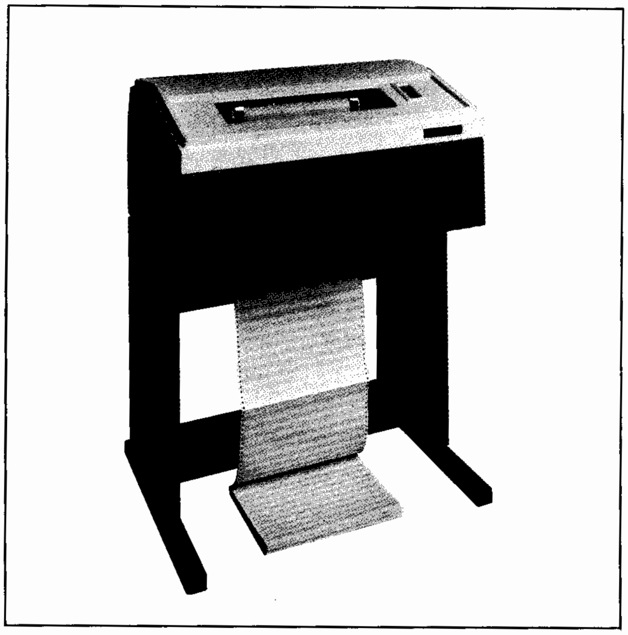
Availability

- Series II
- Series III

Medium Speed Line Printer

HEWLETT PACKARD

Model 2617A



The HP 2617A line printer represents an optimized combination of high print quality, speed and low price in a drum printer subsystem. This unit is ideally suited for applications requiring a medium speed printer.

Features

- Prints 136 columns of 64 characters at 600 lines per minute
- Prints 136 columns of 96 characters at 436 lines per minute
- 12 channel Vertical Forms Control
- Single through 6 part forms
- 64 or 96-character ASCII drums
- 6 or 8 lines per inch—operator selectable

A 12 channel Vertical Format Unit allows convenient and efficient printing in predetermined formats such as business forms. The VFU uses commonly available control tapes. Under program control the VFU may be commanded to slew to the next hole in a given channel of the tape or to slew an absolute number of lines from 0 to 15. Two VFU channels can be monitored by the computer from user-written SPL procedures. This feature may be used to further simplify forms control programming.

- Special character sets available (non-English language and OCR fonts)

A switch allows the operator to choose 6 or 8 line per inch. An appropriate VFU tape should be mounted on the printer. Six part forms are easily handled by the HP 2617A. Forms alignment is simplified by horizontal and vertical paper alignment guides. Fine vertical paper alignment adjustments can be made while printing. A paper receptacle is provided on the rear of the printer. Operation at high paper rates and low humidity is aided by inclusion of an electronic static eliminator.

An ASCII 64 character set is standard with 96 character ASCII optional. Special character sets including non-English language and OCR fonts are available as specials. Consult your local HP Sales Representative for information.

SPECIFICATIONS

Paper feed

One set of pin tractors above drum, friction paper tensioner below drum.

Line advance: 25 milliseconds

Slew rate: 25 inches (63.5 cm) per second

Paper dimensions

Standard fanfold, edge punched paper 4 inches (10.2 cm) to 16 inches (40.6 cm) wide

Paper type

Single copy, 15 lb. bond minimum weight
Multi-copy up to 6 parts, 12 lb. bond with single shot carbon

Character drum

Characters per line: 136

Symbol size (typical):
0.095 inch (2.4 mm) high
0.065 inch (1.65 mm) wide

Vertical format unit

Number of channels: 12

Addressing: Slew to next hole in channel "n" or slew 0 to 15 lines

Status channels: 9 and 12 may be read back by user programs to determine form position

(continued)

Throughput data

Duty cycle: 40%
Print cycle: 40% per page
850 pages/hour for heavy print periods: 2747 pages average throughput/8 hour day. Refer to "A Guide to Hewlett-Packard Printers" for detailed performance data.

Printing speeds

Char. Set	Drum Speed	136 Char. Line Per Minute
64	800	600
96	533	436

Dimensions

Width: 33 inches (83.8 cm)

Depth: 26 inches (66 cm)

Height: 45 inches (114.5 cm)

Shipping Weight: 415 pounds (188.2 kg)

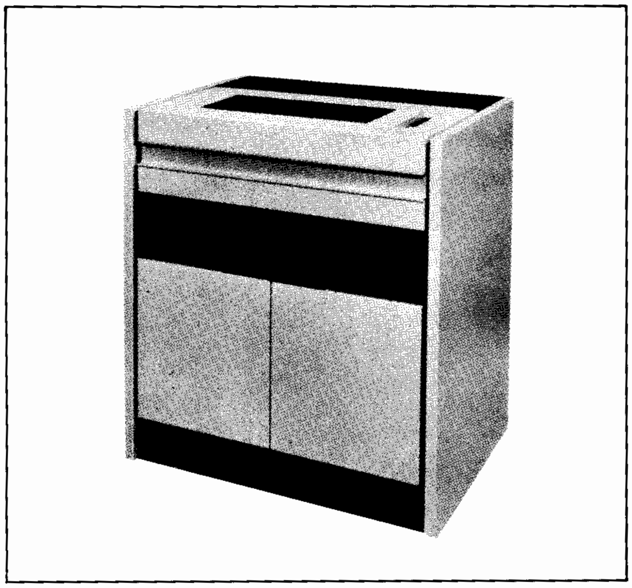
Availability

- Series II
- Series III

High Speed Chain Line Printer

HEWLETT hp PACKARD

Model 2619A



The HP 2619A line printer uses chain technology and other features to provide high speed, high quality printed output with superior horizontal registration, high reliability, and quiet operation. The printer is ideally suited for applications where both high print quality and speed are requirements.

Features

- Horizontal font (chain-type) print technology provides high quality print with excellent horizontal registration
- Prints 132 columns at rates up to 1000 lines per minute with 64 character set
- Prints 132 columns at rates up to 750 lines per minute with 96 character set
- 12-channel vertical forms control
- 6 or 8 lines per inch—operator selectable
- Multiple copies—prints up to 6-part forms
- Both 64-character and 96-character ASCII available as well as an OCR-B font option for both
- Built-in self test feature and impending paper outage warning

Utilizing a 132 column, horizontal font print technology, this unit prints at 1000 LPM using the 64-character ASCII character set or 750 LPM using the 96-character ASCII character set. Throughput is further enhanced by a 40 inch per second paper slew speed.

Special forms control may be selected easily using the paper tape 12 channel VFU and operator selectable 6 or 8 lines per inch control. Forms thickness adjustment enables the operator to achieve high quality print on a wide variety of forms up to six parts thick.

Standard options include both 64 and 96 character ASCII sets along with an OCR-B font option. A special order can also be placed for a custom heavy duty character set for applications which require nearly constant repetitive printing of a limited character set.

For ease of maintenance, the unit has a built-in self test capability with fault indicator lights. In addition, built-in sensors detect an impending paper-out condition and allow the present page to complete printing prior to indicating paper-out.

SPECIFICATIONS

Paper feed

Two sets of 8 pin tractors

Line advance: 15 Milliseconds

Slew speed: 40 inches (101.6 cm) per second

Paper dimensions

Standard fan fold edge punched

Width: 3.5 to 19.5 inches (8.89 to 49.5 cm)

Length: 0.5 to 18 inches (1.27 to.45.72 cm)

Paper type

Single copy, 15 lb. bond minimum weight

Multi-copy up to 6 parts, 12 lb. bond with single slot carbon

Character chain

Characters per line: 132

Character type: Open Gothic

Typical symbol size:
0.095 inch (2.4 mm) high
0.065 inch (1.65 mm) wide

(continued)

Vertical format unit

Number of channels: 12

Addressing: Slew to next hole in channel "n" or slew 0 to 15 lines

Status channels: 9 and 12 may be read back by user programs to determine form position

Throughput data

Duty cycle: 40%

Print Cycle: 40% per page

1400 pages/hour for heavy print periods; 4480 pages average throughput/day. Refer to "A Guide to Hewlett-Packard Printers" for detailed performance data.

Printing speeds

Character Set	132 Character Lines Per Minute
64	1000
96	750

Physical characteristics

Height: 42.75 inches (108.5 cm)

Width: 36.5 inches (92.7 cm)

Depth: 26.0 inches (66.04 cm)

Weight: 570 lb. (258.5 kg)

Availability

- Series II
- Series III

Dot Matrix Character Printer

HEWLETT PACKARD

Model 2631A

The HP 2631A is a dot matrix character printer featuring high serial throughput and a wide range of printing functions. Hewlett-Packard's advanced silicon-on-sapphire (SOS) microprocessor technology is used in the HP 2631A to optimize printing functions and to provide self-test capability.

Features

- Prints at 180 characters per second
- High resolution dot matrix characters
- 128 USASCII character set
- Long life print head
- Cartridge ribbon
- Single through 6 part forms
- 8 channel fixed Vertical Format Control (VFC)
- Special user features
  —manual self test
  —paper out detection
  —auto line feed
- 6 or 8 lines per inch
- Up to 136 characters per line
- Attaches via HP-1B cable to General I/O Channel

High throughput

The HP 2631A has a maximum print speed of 180 characters per second. Print speed, however, is only one of several factors which affect throughput. The HP 2631A optimizes throughput by printing bidirectionally, suppressing leading and trailing blanks, skipping over embedded blanks, and returning the carriage at high speed. The HP 2631A utilizes an HP CMOS/SOS microprocessor to optimize printing functions, data manipulation, and other control functions.

High resolution dot matrix

The HP 2631A utilizes a high resolution seven column by nine row dot matrix character cell for crisp, clear character formation. The nine high matrix allows true descenders, commas, semi-colons, and underline. Dot matrix is especially well suited for multi-part forms; since each dot is formed with equal intensity, the sixth copy has unusually good character resolution. The versatility of dot matrix allows the HP 2631A to print the entire 128 USASCII character set (upper and lower case alphabet, numbers, symbols, and control codes).

Long life print head

The nine-wire print head has a lifetime rated at 130 million characters. When replacement is required, this self-aligning head is easily replaced using the hex key supplied with the printer.

Cartridge ribbon

The cartridge ribbon has a long life of more than 10 million characters. This is made possible by an internal mobius loop design that permits the use of the entire ribbon area. When a ribbon change is necessary, it is a quick, simple and clean task.

Single or multipart forms

Up to 6 part forms (to 0.43mm thick) may be accepted by the HP 2631A. Due to a unique head design, adjustment for forms thickness variations between 0.08mm (.003") and 0.43mm (.017") is usually not required. A lever adjustment on the print head carriage is provided to allow optimization of print quality on various forms.

Form width may be from 31mm (1.22") to 400 mm (15.75"). Vertical forms control is provided through an eight channel fixed electronic VFC. There is no paper tape or paper tape reader to worry about, since the VFC is located in read only memory (ROM).

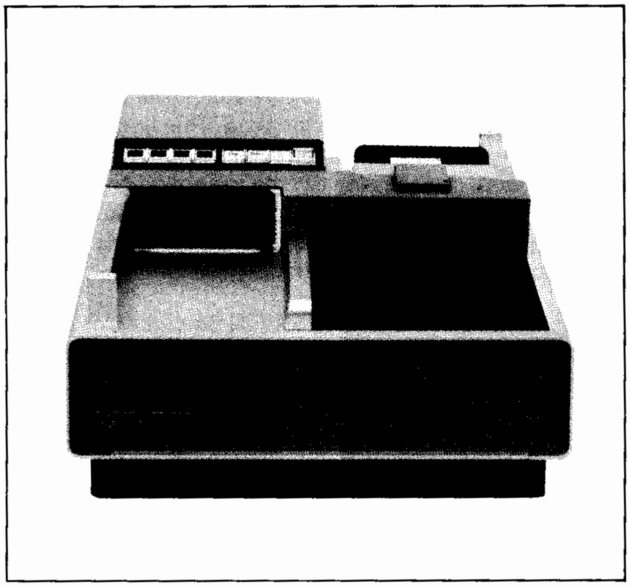
Availability

- Series 30
- Series 33

Card Reader Subsystem

HEWLETT hp PACKARD

Model 30106A



The HP 30106A subsystem provides dependable medium speed card reading capability. A vacuum pick mechanism is used in conjunction with riffle air for ease of card picking and minimum card wear. This technique also permits extremely high tolerance to damaged or worn cards. The card track is very short so that at no time is more than one card in motion.

Features

- Reads 600 punched cards per minute
- Vacuum card picking
- Slant-top design for smooth card flow
- Straight-through card-track for long card life
- Automatic feed
- 1000 card hopper/stacker

The many checking features of the reader insure safe, dependable operation. These include light/dark check, motion check, pick check for stapled cards, and hopper checks.

SPECIFICATIONS

Card rate: 600 cards per minute

Card type: Standard 80-column EIA card Hopper/stacker

Capacity: 1000 cards

Light source; Infrared light emitting diodes

Read station: Photo transistor, 12 bits simultaneously

Controls: Stop, Reset, End of File, Power

Indicators: Read Check, Motion Check, Pick Check, Hopper/Stacker

Data formatting

The HP 3000 interface controller provides Hollerith to ASCII conversion with packing and column binary conversion (each column plus four leading zeros packed into two bytes).

Physical characteristics

Height: 15.5 inches (39.4 cm)

Width: 23-1/16 inches (58.6 cm)

Depth: 18 inches (45.7 cm)

Shipping weight: 100 lbs (45.4 kg)

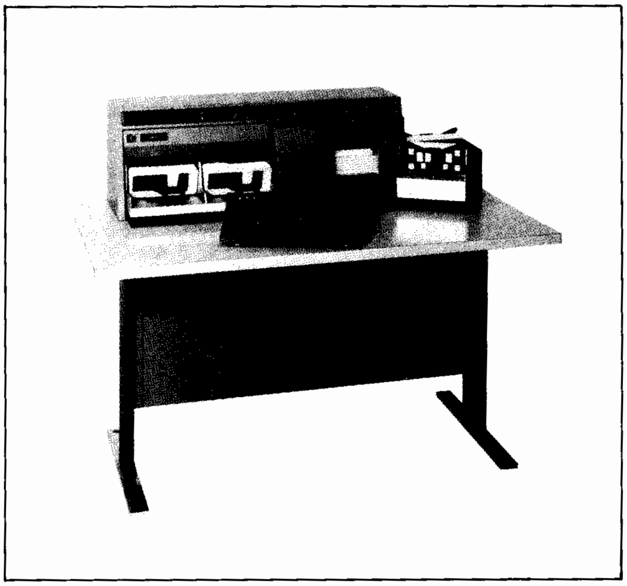
Availability

- Series II
- Series III

Card Reader Punch Subsystem

HEWLETT PACKARD

Model 30119A



The HP 30119A subsystem provides fully buffered, on-line 80-column card reading, punching, and printing capabilities for use with HP 3000 computers. Off-line data record (keypunch) capability is optionally available. Reading rate is 175 to 200 cards per minute. Punch/print rate is 45 to 75 cards per minute, depending on the number of columns involved. A fast slew rate allows rapid skipping of columns and fields not requiring punching or printing.

Features

- On-line 80-column card reading, card punching, and card printing
- Read, punch, and print functions independently controlled by the computer system
- Individual data storage buffers for each function
- Dual input hoppers and output stackers selectable under program control
- Provides software-driver plus hardware interface for use with HP 3000 computers

Separately buffered and independently controlled read, punch, and print functions may be utilized. This feature allows operations such as verification of previously punched cards, printing of information different from that punched, or duplication of existing cards.

Primary and secondary input hoppers are provided with capacities of 600 and 400 cards respectively. Under program control, cards may be selected from either input hopper and directed to either 400-card output stacker following reading, punching, or printing. This feature eliminates the need for interfiling of blank cards in decks to be duplicated, provides for automatic separation of original and duplicate cards, and allows for sort/merge operations under control of user's programs. Input and output punched code format is Hollerith.

The printer employs a modified ASCII 64-character set as standard. A 64-character Swedish/Finnish character set can be ordered as a special option.

Selection of optional configuration 002, provides the user with the following powerful off-line capabilities:

- Complete off-line preparation and on-line input/output operations in a single unit increases the cost effectiveness of the subsystem
- A comprehensive control panel allows the operator to select ON-LINE (computer controlled) operation or OFF-LINE (operator controlled) functions. OFF-LINE functions include card format program preparation, program loading, punching, verifying, reproducting, or printing. Included are illuminated indicators for machine status, error and column number.
- Keyed data is stored in an 80-character buffer memory prior to punching or printing. Backspace and Erase keys may be used to edit this data by character, field, or record.

Control programs may be prepared by the operator. These programs define the format for various card operations by designating field boundaries, data type for each field (alpha or numeric) and columns to be skipped. Memory is provided in the unit for storage of up to 4 programs, each of which is keyboard selectable. Reversion to a designated program is automatic following temporary selection of any other program.

When an error has been corrected during verification, the errored card is automatically placed in the unused stacker and a blank card fed and punched with the corrected data.

(continued)

SPECIFICATIONS

Code compatibility

Hollerith (ANSI X3.26-1970)

Character set (printer)

Uppercase modified ASCII standard

Uppercase Swedish/Finnish available as special order

Performance

Card type: 80 column

Reading rate: 200 cards per minute

Punching and/or printing rate: 45-75 cards per minute depending on number of columns

Input hopper capacity, primary: 600 cards

Secondary: 400 cards

Output stacker capacities, both: 400 cards

Physical characteristics

Height: 40 inches (102 cm)

Width: 42 inches (107 cm)

Depth: 27 inches (69 cm)

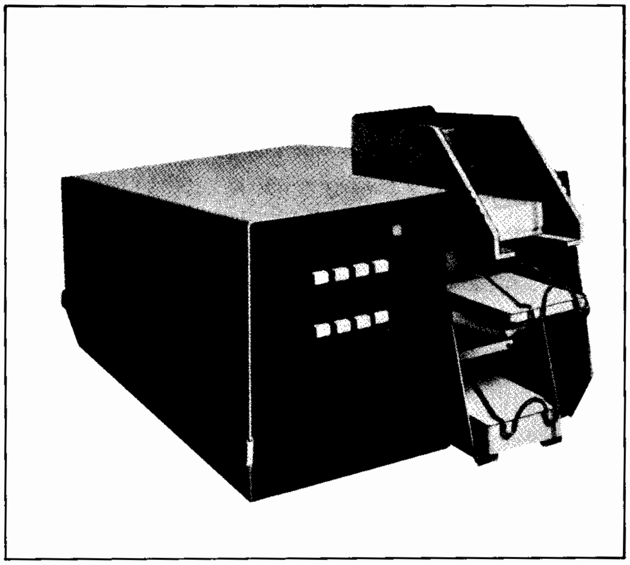
Approximate shipping weight: 350 lb. (158.9 kg)

Availability

- Series II
- Series III

Optical Mark Reader

Model 7260A



The 7260A Optical Mark Reader (OMR) offers mark sense card reading capability for terminals of an HP 3000 computer system.

Features

- Single card used as source document and for data entry
- Reads marks made by ordinary soft lead pencil
- Reads standard punched cards
- ASCII and column image reading formats
- Switchable off-line operation with terminals
- Select hopper available
- Data retransmission capability
- FORTRAN, BASIC, COBOL, SPL callable
- Mixed marked/punched cards

Each form may contain pencil marks and any combination of prepunched holes and/or preprinted marks.

Customized forms

Any number of columns from one to 96 may be read. Many variations in layout and color may be used to produce a form. Different colors may be used to visually identify different forms or sections on a form. Clock marks allow vertical columns to be positioned where desired on the form. An optional select hopper is available which allows selected cards to be fed into a separate hopper under program control.

A terminal oriented document reader

The 7260A Optical Mark Reader is a serial ASCII device. The unit is designed for operation in conjunction with a terminal for remote applications via full duplex modems or hardwired connection to the asynchronous terminal controller of the HP 3000. The HP 3000 has full control over the 7260A. By adding a 7260A to a terminal on the same port of the HP 3000, you are able to supply input from a medium other than the keyboard, saving on line computer time. The terminal can be muted when the reader is transmitting data to the HP 3000.

Multiple data rates

Forms can be read at a rate up to 300 cards per minute. Card feed rate depends on the amount of data to be read on each card and the data transmission rate. Rates of 110, 150, 300, 1200, and 2400 baud can be selected. By proper use of end of record marks, only data marks on cards will be read, trailing spaces will be ignored, thus optimizing data transmission.

ASCII or image reading modes

In the ASCII mode, the card form marks (or punched holes) are interpreted as standard 128 character Hollerith code and the data is transmitted in 7-level ASCII code with even parity. In image mode, the Optical Reader transmits a two character representation of each column of data. This permits all 4096 possible combinations of marks to be read. Switching between ASCII or image mode can be done under program control.

Operating mode

The HP 3000 supports demand mode operation of the HP 7260A in conjunction with the 2640 series of interactive display terminals.

The maximum number of 7260A Optical Mark Readers that can transmit data to the computer simultaneously is outlined in the table below.

1 unit operating at 2400 baud
1 unit operating at 1200 baud and 2 units at 300 baud
1 unit operating at 1200 baud and 4 units at 150 baud
5 units operating at 300 baud
8 units operating at 150 baud

Because the 7260A is used in conjunction with a terminal (in session mode) it cannot be used as a job accepting device.

(continued)

SPECIFICATIONS

Forms

Standard tab card size 8.26 × 18.73 cm to 8.26 × 28.26 cm.

Hopper capacity

450-card input; 450-card output.

Controls

Line Switch, Ready, Stop, Terminal Mute, Single Pick, Continuous Pick (not used with the HP 3000), Line/Local, and Full/Half

Indicators

Ready, Pick Fail

Interface

The unit uses Electronic Industries (EIA) RS232C specification.

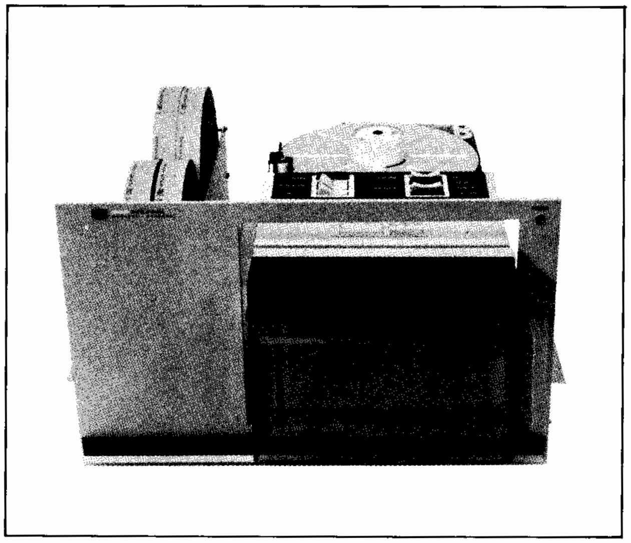
Availability

- Series 30
- Series 33
- Series II
- Series III

Tape Punch Subsystem

HEWLETT PACKARD

Model 30105A



The HP 30105A Tape Punch Subsystem provides a compact and quiet-running paper, plastic, and mylar tape punching facility.

Features

- Compact and quiet-running
- Punches tape at 75 characters per second
- Punches both paper tape and mylar tape

SPECIFICATIONS

Punch speed

75 characters per second, asynchronous

Tape type

Paper, mylar or plastic

Tape width

Standard 5 level 11/16 inch (17.5 mm) and 8 level 1 inch (25.4 mm)

Tape thickness

Paper: 0.003 to 0.005 inch (0.08 to 0.13 mm) oil-base or dry

Mylar: 0.003 to 0.004 inch (0.08 to 0.10 mm)

Plastic: 0.003 to 0.0045 inch (0.08 to 0.11 mm)

Physical characteristics

Height: 10-1/2 inches (26.7 cm)

Width: 16-3/4 inches (42.5 cm)

Depth: 21-3/16 inches (53.8 cm)

Shipping weight: 47 lb (21.3 kg)

Operating controls

Power On, DC On, Tape Feed, Ext (external interrupt), Feed Holes, Code Holes

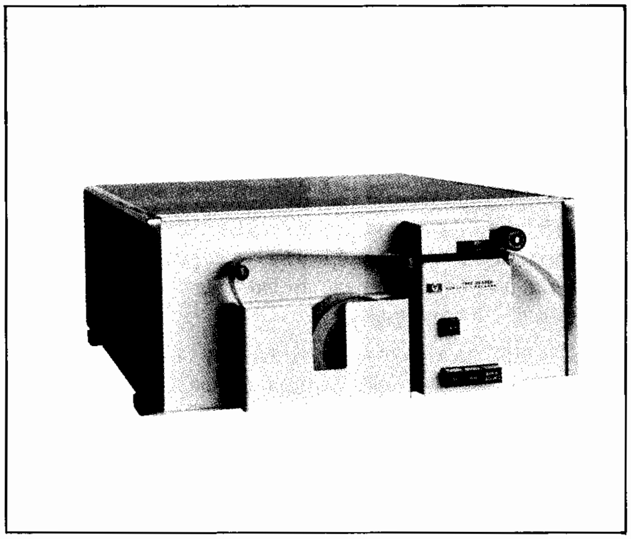
Availability

- Series II
- Series III

Punched Tape Reader Subsystem

HEWLETT hp PACKARD

Model 30104A



The HP 30104A Punched Tape Reader Subsystem is a high-speed device for the error-free reading of both dry and oilbase tape.

Features

- Reading speed to 500 characters per second
- Error-free reading of both dry and oil-base tape without adjustment
- Simple operation and rugged construction for long life

The high speed of the reader permits rapid read-in while offering the economy and versatility of punched tape input. A significant advantage in reading accuracy is also provided by using a compensating sensing technique. Data reliability is enhanced as each character is read only once with no overshooting of characters. Positive feedhole control and a reliable clutch/brake mechanism ensure that the tape will stop on the character that initiates the stop. Simple operation, rugged construction and electrically-conservative design ensures long life at top performance.

SPECIFICATIONS

Reading speed

500 characters per second (415 characters per second when operated from 50 Hz power)

Reading technique

Photoelectric, character-by-character

Tape

Code: 8 level code

Width: 1 inch (2.54 cm)

Material: Any material with less than 60% transmissivity

Start/stop times

Start time: Less than 6 milliseconds

Stop time: Less than 500 microseconds (stops on character)

Controls

Power, Load, Read, Manual Advance

Physical characteristics

Height: 7 inches (17.8 cm)

Width: 17 inches (43.2 cm)

Depth: 16 inches (40.6 cm), not including panel controls and connectors

Shipping weight: 54 lb. (24.5 kg)

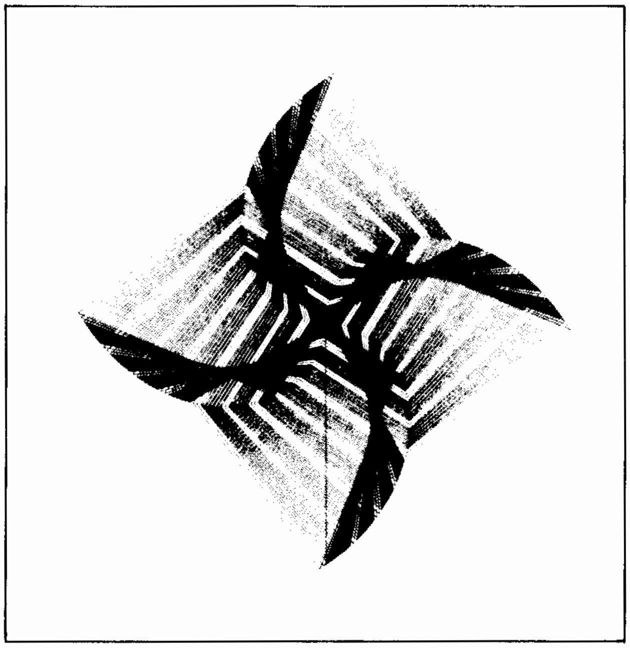
Availability

- Series II
- Series III

CalComp Plotter Interface Subsystem

HEWLETT hp PACKARD

Model 30126A



The HP 30126A subsystem interfaces the HP 3000 computer with a CalComp 565 or 702 series plotter. Zip mode format is compatible with the software supplied. The complete interface consists of a single printed circuit board, software driver, basic plotting software and a single cable for interconnecting the plotter and interface.

Features

- For CalComp series 565 and 702 plotters
- Zip mode format supported
- Simple subroutines callable from FORTRAN, COBOL SPL and BASIC

The user initiated procedures are programmed to translate computed data into distinct plotter commands necessary to direct an on-line plotter. The resulting graphic form can include graphs, three-dimensional drawings, contour maps, charts, etc., and plot annotation (ASCII alphanumeric characters and special graphic symbols). The subsystem is also responsible for file maintenance operations related to the plotter file, and input/output error-handling.

Easy to program

The basic plotting software consists of five FORTRAN callable procedures; their functions are described below.

1. PLOTS: Initializes plotter variables, initializes a user-defined plotter commands buffer, and opens the plotter file.
2. PLOT: Converts X-axis and Y-axis parameters into plotter commands, manages buffering of plotter commands, and closes the plotter file when the plotting sequence is completed.
3. FACTOR: Changes the plot factor (the ratio of the plot physical size to the plot command size).
4. WHERE: Returns the X-axis and Y-axis coordinates of the present pen position (with respect to the current origin) and returns the current plot factor.)
5. SYMBOL: Writes plot annotation in the form of ASCII characters and special symbols.

In addition, through the courtesy of Cal Comp Corporation the dollowing four additional routines are provided.

1. NUMBER: Converts a floating-point number to the appropriate decimal equivalent in order that the number may be plotted in the FORTRAN F-Type format.
2. SCALE: Examines the data values in an array, also determines a starting value and a scaling factor.
3. AXIS: Indicates the orientation and values of the plotted data points. When both the X and Y axes are needed. AXIS is called separately for each one.
4. LINE: Produces a line plot of the paired data points contained in arrays X and Y. Also computes the page coordinates and scaling factor of these points.

Availability

- Series II
- Series III

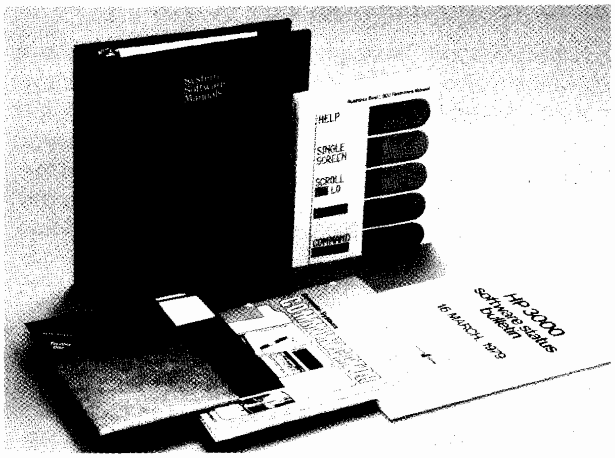
HP Computer Systems
Software Support Services

HEWLETT PACKARD

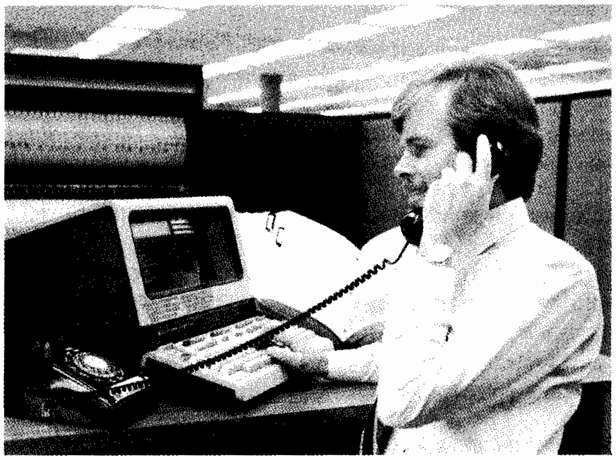
Hewlett-Packard offers a range of software support services designed to help ensure customer satisfaction with HP computer systems.

Features of Hewlett-Packard's software support include:

- A well-defined set of support services optimized for different customer needs
- The flexibility for customers to choose from a range of support services
- Support arrangements specifically designed for the needs of multiple installations
- HP's commitment to high quality support and customer satisfaction

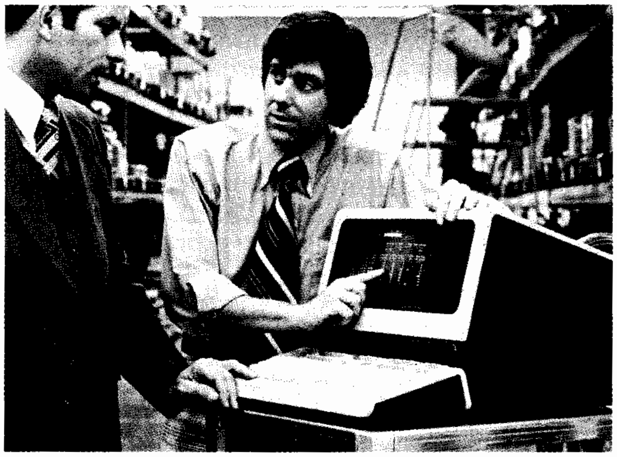


HP software support services are purchased in conjunction with each HP software product that you install on your systems.



Customer Support Service (CSS)

Customer Support Service (CSS) is HP's standard software support product. It helps to ensure that the customer who is developing applications on an HP computer system receives all the tools necessary to be successful. It offers a comprehensive set of software services and the personal attention of a trained HP Systems Engineer. For those customers who centrally develop applications for other installations, incremental services are available for the central site to include the tools the customer needs to ensure success at additional locations.



Software Subscription Service (SSS)

Software Subscription Service (SSS) is available for those customers who choose to rely upon their in-house resources for their software support. No HP Systems Engineering assistance is provided under this service.

In addition to the above two alternative software support services, two Documentation Distribution Services are also available for the customer who wants to receive or maintain multiple sets of documentation material. Also, a range of customer training and applications design consulting is available to aid customers in the proper use of HP computers.

(continued)

Customer Support Service (CSS)

Customer Support Service is Hewlett-Packard's standard software support product. CSS provides the highest level of ongoing support available from Hewlett-Packard. It is designed to assist customers who desire a close support relationship with Hewlett-Packard. From the moment the customer's system is installed, Hewlett-Packard support personnel are available to provide prompt and individual attention to the customer's needs.

Features:

Right to Use Software Updates and Applicable Firmware Updates. By purchasing Customer Support Service, the customer is granted the right to use the software updates and firmware updates (if applicable) on one HP computer system.

Delivery of Software Updates and Firmware Updates When HP enhances its software or develops solutions for known software or firmware problems, one copy of the software updates and applicable firmware is delivered to the customer's System Manager.

Reference Manual Updates To help the customer make effective use of HP software, one copy of page inserts, page replacements or new editions of appropriate software reference manuals is mailed quarterly to the System Manager. This assures that the customer has the most recent documentation on how to use all the features of the latest software releases.

Account Responsible Systems Engineer A Hewlett-Packard Systems Engineer will be assigned to the customer's account. Working through the customer's System Manager, the account Systems Engineer, a trained software specialist, will review application development, preview software releases for potential use of new features, and explain any changes which may impact the operations of the computer system. The account Systems Engineer will also coordinate the on-site resources required to resolve a software problem and ensure that all support services are used to maximize the customer's success.

Phone-in Consulting Service The System Manager will receive the telephone number of the appropriate Phone-in Consulting Service office to assure contact with an HP Systems Engineer to discuss questions and provide advice regarding HP software or documentation. Within four working hours after this request, an HP Systems Engineer will call back and help resolve any difficulties. This service is available weekdays excluding HP holidays.

On-site Systems Engineering Assistance. To resolve software and documentation problems that cannot be solved using the Phone-in Consulting Service, Systems Engineering assistance is available at the customer's site within eight working hours after the request from the System Manager. When on site, the System Engineer will help the System Manager identify, verify, isolate, report, and work around problems caused by HP software. Assistance is available weekdays, excluding HP holidays, during HP working hours, to facilities not more than 100 miles (160Km) from the nearest HP office designated to provide on-site SE services. Support for facilities further away can usually be provided at additional cost.

If the problem reported is not associated with an HP software design error or system malfunction, the on-site services are considered outside the scope of Hewlett-Packard's software and support agreement and subject to a time and materials charge.

HP is not obligated to provide any on-site services for HP software products which have been modified by the customer.

Software Status Bulletin One copy of the appropriate Software Status Bulletin is mailed twice each month to the customer's System Manager. This bulletin lists known software errors, problems, and their interim programming solutions, if known. This Bulletin helps the customer avoid time and effort involved in solving software problems that have been resolved elsewhere. A cumulative bulletin is mailed quarterly to consolidate all the current activity.

Software Problem Reporting If the customer encounters a possible software error for which no interim solution exists in the Software Status Bulletin, the System Manager can complete the appropriate software reporting form documenting the error and submit it by mail to the local HP sales office. After receiving the form, HP notifies the customer in writing as to the status of the request.

Communicator One copy of the HP 3000 Communicator is mailed to the System Manager a minimum of four times per year. It contains computer and software operational tips, programming techniques, information on available HP documentation, and articles of general interest. Information on newly released software updates is included to help the customer plan for their implementation.

(continued)

Multiple CSS installations

Customers who have multiple computer systems on which they are developing new applications will most likely benefit from full **Customer Support Service** at each site.

Customers who centrally develop applications on one computer system for use on multiple systems, and choose to support the additional sites through their central installation, may prefer to purchase **Central System Support Service for an Additional System** for each of their remote installations. This service provides the tools the customer's central staff needs to help ensure success at all the system installations they support. The service includes and is restricted to:

—The right to make one copy of the software updates which are delivered to the central site as described previously and the right to distribute and use these updates on one additional computer system. One copy of firmware updates is provided if applicable.

—The right to copy, distribute and modify one copy of the appropriate reference manual and manual updates for use with one additional computer system.

—Use of Phone-In Consulting Service by the central site System Manager on behalf of the additional site.

—On-site assistance by an HP Systems Engineer at the central system to identify and verify software problems on behalf of the additional systems. The request for assistance must originate from the central site System Manager and problems need to be recreated at the central site unless the additional system is installed at the same customer facility address. If the particular problem cannot be resolved at the central site, the System Manager may request HP assistance at the additional site, billable on a time and materials basis.

Additional PICS callers for CSS customers

Phone-in Consulting Service is available for multiple users by adding authorized callers to the same phone-in consulting telephone number used by the customer's System Manager. Each additional user is provided all of the same PICS benefits as the original System Manager except that the additional caller may not request on-site assistance by an HP Systems Engineer. Refer to Phone-in Consulting Service as previously defined for other details of the PICS service.

Software Subscription Service (SSS)

A few customers who are developing new applications may choose to rely entirely upon their in-house resources for their software support and may not need all the services from HP provided by Customer Support Service. For those customers who do not require any Systems Engineering assistance, Software Subscription Service is available. SSS does not provide the assignment of an account-responsible Systems Engineer, Phone-In Consulting, or the on-site Systems Engineering assistance services of CSS, as described previously.

SSS offers specifically the following features as described under CSS:

—Right to use software updates and applicable firmware updates on one system

—Delivery of software updates and firmware updates

—Reference manual updates

—Software Status Bulletin

—Software problem reporting through the mail

—Communicator

SSS does not include:

—Assignment of an account responsible Systems Engineer

—Phone-in Consulting Service

—On-site Systems Engineering assistance.

(continued)

Documentation Distribution Services

In addition to the Software Support Services described above, Hewlett-Packard offers two types of Documentation Distribution Services. These services are appropriate for customers with a large programming staff, many of whom wish to be individually informed of software problems and need to keep their documentation up to date.

The two documentation services are:

1. Software Notification Service
2. Manual Update Service

Software Notification Service provides the customer with one copy of each of two periodicals: the Software Status Bulletin and the Communicator. The Software Status Bulletin is mailed to the recipient of the service twice monthly with a cumulative bulletin mailed quarterly. It lists all known software design errors, problems, and any available interim programming solutions. The Communicator is mailed to the recipient of the service approximately four times yearly and contains articles of interest, information on documentation available from Hewlett-Packard and software, operational, or programming tips. Information on newly released software updates is included in a separate section to help the customer plan for their implementation.

Manual Update Service keeps reference manuals up to date by providing the customer with page inserts, page replacements, or entire new editions to reference manuals when necessary. The HP 3000 Price/Configuration Guide specifies the sets of manuals that receive this service.

Conditions for obtaining software support and documentation distribution services

—An HP trained System Manager responsible for maintaining integrity of the system's hardware and software, or a trained designated alternate, must be identified as a contact for Hewlett-Packard.

—The same level of support service must be purchased for all of the HP software products which make up one computer system. Due to the interaction among software elements, service cannot be given to specific software products while omitting others.

—All associated system hardware and firmware must be maintained at the latest required code revision level. (To reproduce and analyze your software problems, the system must be compatible with those in HP Service Centers.)

—Before beginning support, all system software products must be at the current release or revision level and must not be modified in any manner by the customer.

—If HP is not confident in its ability to support a system (because of system location or any other reason), then it may elect not to provide Customer Support Service for that system.

—Additional PICS Callers can be purchased as many times as desired. However, the name of each authorized caller must be provided and must have been trained in the same manner as the original System Manager.

—The purchase of Customer Support Service (CSS) is a prerequisite to the purchase of either Central System Support Service for an Additional System or Additional PICS Callers.

—Customer Support Service and Software Subscription Service can only be purchased for systems which have installed the associated Standard Software Product or Right to Copy Software Product.

Ordering Information

Software Support Services. CSS and SSS are normally purchased for 12 monthly periods, billable quarterly or yearly in advance as desired. The minimum purchase for each is three months. Detailed ordering information can be obtained from an HP sales representative or by referring to the HP 3000 Computer System Price/Configuration Guide.

Documentation Distribution Services. The minimum purchase for Software Notification Service or Manual Update Service is 12 months, billable quarterly or yearly in advance as desired. Software Notification Service and Manual Update Service can be purchased in any quantity for any software product. Detailed ordering information can be obtained from an HP sales representative or by referring to the HP 3000 Computer System Price/Configuration Guide.

Hewlett-Packard provides a wide range of hardware maintenance services to ensure that your HP computer products are performing their tasks at the optimum operating level and to correct any malfunctions with minimum downtime and interruption to your operations.

These services include:

- Site requirements consultation service.
- Installation service.
- Maintenance Agreement Service, including preventive maintenance, remedial maintenance, engineering improvements, etc.
- Time and material (per call) services.

There are two basic ways to obtain on-site HP maintenance services—via a Customer Support Services Agreement or on a time and materials (per call) basis.

Hewlett-Packard Customer Support Services Agreements are available for all HP 3000 computer systems and associated products, including computer system peripherals, interfaces and accessories. The HP Customer Support Services Agreement provides all the on-site maintenance services required to keep your equipment in top-notch condition for a fixed monthly charge, regardless of the amount and magnitude of services rendered. You need not concern yourself with staffing and training maintenance personnel or maintaining an expensive parts inventory for your HP computer products—it's all included in one nominal monthly charge. And, HP manages your on-site maintenance program at no extra charge.

The basic Customer Support Services Agreement provides the following services:

- Preventive Maintenance—provides regularly scheduled visits to keep your system running smoothly.
- Emergency Repair—covers the cost of all parts and labor required to restore your system in the event of a failure requiring emergency repair.
- Remote System Verification Program (RSVP)—available to HP 3000 Series 30 and Series 33 users. This service is based on the built-in ability of the Series 30 and Series 33 to be remotely diagnosed from an HP field office using a customer provided asynchronous modem.
- Support for Privileged Mode users—should Privileged Mode be required in your application, a Hewlett-Packard representative will outline the training required for Privileged Mode users. In general, all investigation of problems resulting from use of Privileged Mode, and all consulting on its use, are provided on a time and materials basis.

General customer responsibilities

- Support for Foreign Device Users—should you interface a non-Hewlett-Packard product to an HP 3000, support for the foreign device and system modifications required to drive it are your responsibility. Service under the Customer Maintenance Agreement can only be rendered if the HP 3000 system is reconfigured to its original configuration with the current supported version of the operating system. Hewlett-Packard will not warrant compatibility of operating systems, user subsystems, or modifications to HP hardware necessitated by design changes to interface a non-Hewlett-Packard product to an HP 3000.
- Customers must purchase either Customer Support Service (CSS) or Software Subscription Service (SSS) in order for hardware maintenance services to be provided on the HP 3000.
- The customer must be present at all times that Hewlett-Packard Customer Engineers (CEs) are within the customer's site. Hewlett-Packard CEs will not enter or remain in unattended customer facilities nor will they possess keys for access. Hewlett-Packard shall not be liable for delays in performing services due to the customer's failure to keep personnel on-site and/or customer or government imposed security requirements.
- The customer supplies consumables such as line printer paper, magnetic tape, ribbons, cards, format tapes, etc., normally used in operation of the equipment.

Other customer responsibilities on specific services are described in the HP Computer Systems Customer Services Data Book.

Product support life

Hewlett-Packard's objective is to extend the useful life of HP computer products by continuing to provide on-site maintenance services for as long as is feasible. However, practical limitations do exist (i.e., availability of spare parts or insufficient installed customer base required to retain local expertise). For this reason, Hewlett-Packard limits its guarantee of availability of on-site support services to a minimum of five years from the date of last sale and shipment by Hewlett-Packard of a product with the same model number. Hewlett-Packard will continue to provide services for products beyond the support life described above on a time and materials (per call) basis, provided the skills and parts required are available locally.

Warranty

Warranty shall be limited to the correction of any defective maintenance service by restoring the products to good operating condition. HP SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES. No other warranties are expressed or implied. HP SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF FITNESS OR MERCHANTABILITY FOR A PARTICULAR PURPOSE.

Additional warranty information for specific maintenance services is supplied, if applicable, in the HP Computer Systems Customer Services Data Book.

HP 3000: A PROGRAMMER'S INTRODUCTION

This is the entry level course intended for application programmers who will be using the HP 3000, and for others in the data processing field desiring a basic exposure to the HP 3000. It is currently the prerequisite for all other HP 3000 training and consulting, and is available at Hewlett-Packard technical centers or at your site.

Prerequisites:

Previous computer experience, preferably one or more years of application programming experience. Knowledge of at least one of the following languages: BASIC, COBOL, FORTRAN, or RPG.

Purpose:

To teach application programmers the use of the major features of the HP 3000 multiprogramming system in interactive, terminal intensive and batch environments. This basic exposure to the HP 3000 will enable programmers to utilize its capabilities to their best advantage in application programming.

Length:

5 days

Laboratory:

Lab time provides each student with the opportunity for hands-on experience, approximately 40%. A special lab in which students debug supplied programs provides practical experience in the day to day use of the HP 3000 by application programmers.

Course Content:

- Introduction—using the HP terminal
- MPE Fundamentals—the HP 3000 operating system; common MPE commands, syntax and parameters, special MPE commands; generic names; the accounting structure and file addressing techniques.
- Text Editor—Editor files; Editor commands; using Editor to create, maintain, and edit files.
- File System—definition of file and file system; building and specifying files; file access and security; physical aspects of files; fixed, variable and undefined length records; file backup and recovery; compilers; compiling, preparing and running commands; introduction to file copying utility (FCOPY).
- Job Control—concept of a process; interactive sessions/batch jobs; streaming; spooled environment; introduction to manipulating spoolfiles (SPOOK); job control words.
- UDC—creating and using User Defined Commands.
- Utilities—FCOPY to copy files and make dumps, plus introduction to more advanced aspects of FCOPY; using SORT to sort any file; using MERGE to combine many files.
- Program Development—concepts of run-time environment, virtual memory, segmentation, and stack architecture; contents of compiler-generated file, compiler control; contents of program file, map of program file; program execution; process execution.
- Error Detection—types of errors; abort mechanism, abort message, using abort information; file information displays; calling intrinsic to print "tombstones"; using display to find error.
- Data Management Overview-modules on KSAM, HP VIEW, IMAGE/QUERY.
- Data Communication Overview-DS/3000 remote processing.
- User Support Services—responsibilities of your System Manager; PICS; Software Status Bulletin; software releases; MIT and Communicator; Users Group; courses available.

Student Materials:

03000-90121	Using the HP 3000
03000-90012	EDIT/3000 Reference Manual
03000-90064	FCOPY/3000 Reference Manual
30000-90008	General Information Manual
30000-90009	MPE Commands Reference Manual
30000-90045	Index to MPE Reference Documentation
30000-90049	Software Pocket Guide
32214-90001	SORT-MERGE/3000 Reference Manual
22801-90001	Student Workbook
30000-90102	Using Files

Class Size:

20 students maximum at HP technical center (two students per terminal).

On-site class size limited to two students per available terminal. Standard, 10 students; maximum, 16 students.

Ordering Information:

22801A	HP technical center
22801X	On-site

HP 3000: SYSTEM MANAGEMENT AND OPERATION

This is the basic course that your company's System Manager must attend in order to properly manage and control the HP 3000 system. It is available at HP technical centers or on-site.

Prerequisites:

Successful completion of HP 3000: A Programmer's Introduction. It is recommended that the student have a clear understanding of the company's user base, and of the applications to be performed on the HP 3000.

Purpose:

To prepare the student for the responsibility of allocating system resources. The student learns how to operate the system and maintain system records; how to design, implement and modify the account/group/user structure; how to employ MPE commands to control system-wide activity; and learns about considerations affecting operating efficiency.

Length:

5 days

Laboratory:

Lab time provides each student with the opportunity for hands-on experience, approximately 30%.

Course content:

- Designing the account structure—creating, deleting, altering, listing accounts and file security; establishing and altering global security matrices
- System activity logging facility
- Operational management—controlling priority of jobs/sessions
- System generation, alteration and backup
- File backup and system reload
- Power on/off procedures, control panel
- System start-up/shutdown
- Operation/user communication
- Device and device-file-management
- System failure—interpretation and recovery
- Private volumes
- User defined commands

Student materials:

30000-90044 MPE System Utilities Manual

30000-90013 System Console Operator's Guide

30000-90014 System Manager/System Supervisor Manual

22802-90001 Student Workbook

Class size:

18 students maximum at HP technical center. On-site classes limited to two students per available terminal. Standard up to 10 students; maximum, 14 students.

Ordering information:

22802B HP technical center

22802X On-site

HP 3000: IMAGE, DATA BASE MANAGEMENT TRAINING

Prerequisites:

Successful completion of HP 3000: A Programmer's Introduction, and programming experience in any of the following: COBOL, FORTRAN, RPG, or SPL.

Purpose:

To provide the student with a working knowledge of data base concepts, IMAGE/3000 and QUERY/3000. How to create and maintain a data base using the IMAGE utilities, and how to access the data base through programming calls and QUERY. Data base privacy and security are also included.

Length:

5 days

Laboratory:

Lab time provides each student with the opportunity for hands-on experience, approximately 40%.

Course content:

- IMAGE overviews—terminology, description of the IMAGE subsystems
- Development of an IMAGE data base—problem definition, flow-charting, documentation
- Data set relationships—master and detail data sets
- Data set access—serial, directed, calculated and chained, synonym chains
- Data base description language (SCHEMA)—syntax, commands, operation
- Data base privacy and security—system management implications, establishing security levels, controlling access

(continued)

- Accessing data bases through IMAGE library procedures—programmatic access to the data base
- Data base utilities—security backup and recovery from system failures
- Data base maintenance
- QUERY—syntax, calling QUERY, commands, reports
- Case studies

Student materials:

32215-90003 IMAGE/3000 Reference Manual

30000-90042 QUERY/3000 Reference Manual

22956-90001 Student Workbook

Class size:

Maximum, 20 students

Ordering information:

22956A HP technical center

22956X On-site

SPL/FILE SYSTEM INTRODUCTION

Note: Oriented to MPE III operating system.

Prerequisites:

Successful completion of HP 3000: A Programmer's Introduction; working knowledge of a Scientific language (e.g., FORTRAN, ALGOL, PL/1) or Assembler language; completion of SPL Textbook, including exercises (30000-90025), review of System Reference Manual (30000-90024), pages 3-4 to 3-28, and 4-1 to 4-32; and minimum six months experience on an HP 3000 using EDIT/3000, FILE equations, and FORTRAN/3000 or BASIC/3000.

Purpose:

To provide the systems-level programmer/analyst an opportunity to review and/or strengthen his knowledge of SPL and its relationships to the HP 3000. In addition, the basic principles of the HP 3000 File System as it relates to the characteristics and operation of disc files are covered.

Length:

5 days

Laboratory:

Provides each student with the opportunity for hands-on SPL experience, approximately 15%.

Course content:

- SPL—A rapid review of high level syntax, commonly used machine dependent features and stack operation. (2½ days)
- File system introduction—a discussion of disc files including a definition of physical characteristics (record structure, blocking, disc space allocation); meaning of Access Modes; high level approach to the internals of opening and closing disc files, record selection, and buffer management; details on considerations when using and sharing files.

Student materials:

30000-90024 SPL Reference Manual

30000-90010 MPE Intrinsics Manual

30000-90022 Machine Instruction Set Manual

30000-90049 Software Pocket Guide

32100-90001 SPL Pocket Guide

22804-90001 Student Workbook

Class size:

Maximum, 20 students

Ordering information:

22804A HP technical center only

MPE III SPECIAL CAPABILITIES

Note: Oriented to MPE III operating system.

Prerequisites:

Successful completion of the SPL/File System Introduction and review of "Accessing and Altering Files" from the MPE Intrinsics Manual (30000-90010).

Purpose:

Provides the system level programmer/analyst with intensive exposure to special capabilities and features: interactive debugging facility (DEBUG), process handling, interprocess communication, resource management (RINS), special file handling, and customized trap handling.

A student who completes this course should:

- be able to use DEBUG to direct the flow of an SPL or FORTRAN program for the purpose of interactive debugging.
- be familiar with the possible applications, use of the intrinsics, and considerations involved with special MPE capabilities.
- be able to utilize the User Trap Facility with SPL, FORTRAN, and COBOL programs.

Length:

5 days

(continued)

Laboratory:

Provides each student with the opportunity for hands-on experience, approximately 25%.

Course content:

- DEBUG—discussion of non-privileged mode commands used for interactive debugging of SPL, FORTRAN and COBOL programs. Main emphasis is on SPL and FORTRAN.
- Process handling—discussion of MPE intrinsics, applications, and special considerations for the creation and activation of processes from user programs. Example: In SPL and FORTRAN.
- MAIL—discussion of special considerations for interprocess communication facility. Examples in SPL.
- Extra data segments—covers creation, use, sharing and deletion of extra data segments; discussion of applications and MPE intrinsics. Examples in SPL.
- RINS—presents concepts of resource management (global and local RINS), possible applications, use of intrinsics, special considerations. Example in SPL.
- Multi-Access/No Wait I/O—high level approach to internal operation of both file system features. Discussion of possible applications and special consideration and use of intrinsics. Example: In SPL and FORTRAN.
- User trap handling—discussion of the use of arithmetic, library, system and control-Y trap routines. Examples in SPL, FORTRAN, and COBOL.

Student materials:

30000-90010 MPE Intrinsics Manual

30000-90012 DEBUG Manual

30000-90049 Software Pocket Guide

22805-90001 Student Workbook

Class size:

Maximum, 20 students

Ordering information:

22805A HP technical center only

CONVERSION OF HP 3000CX OR SERIES I TO MPE III BASED SYSTEMS

Prerequisites:

Successful completion of HP 3000: A Programmer's Introduction, and HP 3000: System Management and Operation, or equivalent pre-Series II training.

Purpose:

To enable the current 3000CX or Series I owner to understand the difference between these systems and MPE III based systems. The objective of the course is to teach the student the key user-feature of MPE III, the functional differences between the previous models and the MPE III based systems, and the steps the customer must take when converting from a 3000CX or Series I installation to a MPE III based system.

Length:

1 day

Course content:

- New addressing scheme
- Changes to MPE
- New/changed intrinsics
- File system changes
- Changed utilities
- Necessary conversion considerations for SPL, BASIC and FORTRAN

Student materials:

22818-90001 Student Workbook

30000-90046 HP 3000CX to Series II Program Conversion Guide

Class size:

Maximum, 10 students.

Ordering information:

22818A On-site only

HP VIEW/3000

Prerequisite:

Successful completion of HP 3000: A Programmer's Introduction.

Purpose:

To train the student in the use of all facilities of HP VIEW/3000, including forms design, data entry, reformatting and intrinsics.

Upon completion of this course, the student will be able to:

- Create a form utilizing all advanced editing and processing specifications.
- Use the stand-alone data entry capacity of HP VIEW/3000 to enter, modify and delete data.

(continued)

- Reformat HP VIEW/3000 data entry output files to meet various record and data requirements of the user.
- Incorporate HP VIEW/3000 callable procedures into an application program in order to tailor it to meet individual needs.

Length:

4 days

Laboratory:

Interspersed throughout course to give students extensive practice in the use of HP VIEW/3000, approximately 50%.

Course content:

- Forms
  - Design
  - Editing and Data Manipulation
  - Sequences of Forms
  - Processing
- Reformatting
  - Data Format Specifications
  - Processing
- Data Entry
  - Collect Mode
  - Browse Mode
  - Modification
  - Output
- Intrinsics
  - Terminal Management
  - Data Management
  - Processing Control
  - Error Handling

Student materials:

32209-90001	HP VIEW/3000 Reference Manual
32209-90002	HP VIEW/3000 Programmer/Designer's Pocket Guide
32209-90003	Operator's Quick Reference Guide
32209-90004	Using HP VIEW/3000
7120-1189	ENTRY Template
7120-1190	FORMSPEC/REFSPEC Template

Class size:

2 students per available terminal. Standard, 10 students; maximum, 16 students.

Ordering information:

22830A HP Technical Center

22830X On-site

HP 3000: KSAM (KEYED SEQUENTIAL ACCESS METHOD)

Note: KSAM is available only on the MPE III operating system.

Prerequisites:

Successful completion of HP 3000: A Programmer's Introduction, and a working knowledge of one or more of the following languages: COBOL, RPG, SPL, FORTRAN, BASIC.

Purpose:

To teach the concepts of indexed files and how to build, maintain and use HP 3000 KSAM files. While the KSAM interface to above languages is discussed, COBOL is stressed.

Length:

2 days

Laboratory:

One-half day employing the COBOL interface to KSAM, the KSAM utility program, developing an understanding of the KSAM file structure, and optimization of key file and extra data segment sizes.

Course content:

- Indexed file concept
- Conformance to MPE
- Design considerations
- KSAM file structure
- KSAM file maintenance
- KSAM language interfaces

Student materials:

30000-90079	KSAM Reference Manual
22828-90001	Student Workbook

Class size:

Limited to 2 students per available terminal. Standard up to 10 students; maximum, 14 students.

Ordering information:

22828A On-site only

DISTRIBUTED SYSTEMS/3000

Prerequisites:

Successful completion of HP 3000: A Programmers' Introduction and programming experience in FORTRAN, COBOL or SPL. FORTRAN or COBOL programmers must be capable of writing SPL procedure calls if use of the Program-to-Program Communication capability is intended. It is suggested the student have the same in-depth knowledge of the File System as presented in the SPL/File System Course to gain maximum benefit from this training.

Purpose:

To provide the student with a working knowledge of the capabilities and features of DS/3000. How to initiate and control inter-node activity employing remote commands; how to access or transfer files between nodes interactively or under programmatic control and how to configure a DS/3000 node for optimum performance.

Length:

3 days

Laboratory:

Lab time provides each student with the opportunity for hands-on experience, approximately 40%.

Course content:

- Telecommunications Systems overview–Terminology, basic configurations, and component descriptions.
- DS/3000 product overview–Configuration, components, features and capabilities.
- Remote Command Processing—Use in the local or remote environment for network control.
- Remote File Access—Interactive or programmed access operation to local or remote files, copying and providing file backup, transferring files between nodes and directing compiler/utilities/subsystem file accesses under DS/3000.
- Program to Program communications—Discussion of intrinsics utilized, programming techniques and case studies of prepared programs.
- Error messages—Interpretation and recovery.
- System generation—Emphasis on configuring DS devices for optimum performance.

Student material:

5955-1715	Guide Book to Data Communications
32190-90001	DS/3000 Reference Manual
36900-90001	Student Study Guide
36900-90005	DS/3000 to DS/1000 Reference Manual

Ordering information:

36900E On-site only

APPENDICES

Appendix A: Documentation

Overview	Guide to a Successful Installation 30000-90135 **$3.25**	General Information Manual 30000-90008 **$5.00**	MFG/3000 General Information Manual 30000-90188				
	Using the HP 3000 03000-90121 **$8.25**	Using files 30000-90102 **$4.50**					
HP 3000 Series III Hardware	Site Prep Manual 30000-90145 **$6.00**	Site Planning Workbook 30000-90146 **$9.00**	System Reference Manual 30000-90020 **$8.25**	Machine Instruction Set 30000-90022 **$5.50**	Instruction Decoding PG 30000-90057 **$1.55**		
HP 3000 Series 30 & 33 Hardware	Site Prep Manual, Series 30 30080-60050 **$4.00**	Site Prep Manual, Series 33 30070-90007 **$4.00**	Instruction Decoding PG 30070-90026 **$2.75**				
MPE	Software Pocket Guide 30000-90049 **$5.25**	System Manager/ System Supervisor 30000-90014 **$9.00**	Series III Console Operator's Guide 30000-90013 **$13.50**	Series 33 Console Operator's Guide 30070-90025 **$12.75**	MPE Commands 30000-90009 **$13.50**	MPE Intrinsics 30000-90010 **$14.00**	
	Segmenter 30000-90011 **$3.40**	System Utilities 30000-90044 **$4.80**	Debug/Stack Dump 30000-90012 **$4.40**	Error Messages 30000-90015 **$14.00**	Index to MPE 30000-90045 **$3.70**		
Utilities	EDIT 03000-90012 **$6.00**	FCOPY 03000-90064 **$4.50**	SORT 32214-90001 **$3.50**	Compiler Library 30000-90028 **$8.50**			
Data Management	Using HP VIEW/3000 32209-90004 *	HP VIEW/3000 Reference Manual 32209-90001 **$12.75**	HP VIEW/3000 Operator's Guide 32209-90003 **$2.50**	HP VIEW/3000 Programmer's Pocket Guide 32209-90002 **$1.00**			
	KSAM/3000 30000-90079 **$10.00**	IMAGE/3000 32215-90003 **$9.50**	QUERY/3000 30000-90042 **$7.50**				
Languages	Using COBOL/3000 32213-90003 **$6.50**	RPG/3000 Listing Analyzer 32104-90003 **$.50**	FORTRAN/3000 Pocket Guide 32102-90002 **$1.55**	BASIC/3000 Pocket Guide 03000-90050 **$.75**	BASIC/3000 Compiler 32103-90001 **$2.90**	SPL Pocket Guide 32100-90001 **$2.05**	APL\3000 Pocket Guide 32105-90003 **$4.40**
	COBOL/3000 32213-90001 **$12.00**	RPG/3000 32104-90001 **$22.00**	FORTRAN/3000 30000-90040 **$8.50**	BASIC/3000 Interpreter 3000-90026 **$10.50**	BASIC/3000 for Beginners 03000-90025 **$6.00**	SPL 30000-90024 **$9.50**	APL\3000 32105-90002 **$9.50**
			Scientific Library 30000-90027 **$4.20**			SPL Textbook 30000-90025 **$7.50**	
Data Communi-cations	Guidebook to Data Communications 5955-1715 **$3.10**	Data Communications Pocket Guide 30000-90105 **$14.00**	RJE/3000 30000-90047 **$7.50**	DS/3000 32190-90001 **$11.00**	DS/3000 to DS/1000 32190-90005 **$7.25**	MRJE/3000 32192-90001 **$8.75**	MTS/3000 32193-90002 **$6.50**
Manufacturing	EDC/3000 User Manual 32380-90001 **$20.00**	EDC/3000 System Admin. Manual 32380-90002 **$8.50**	10S/3000 User Manual 32384-90001 **$25.00**	10S/3000 System Admin. Manual 32384-90002 **$11.00**	MRP/3000 User/Admin. Manual 32388-90001 **$19.50**	SPC/3000 User Manual 32392-90001 *	
Conversions	System/3 to HP 3000 Conversion 32104-90004 **$5.75**	3000CX to Series II Conversion 30000-90046 **$3.30**					

**Consult your Hewlett-Packard Sales Representative for current prices.*

Appendix B: MPE III Commands

All MPE III commands are summarized below, grouped alphabetically within capability. The last column denotes when the command can be issued: during a batch job (J), during a session (S), during a break (B), or programmatically (P), through the COMMAND intrinsic.

STANDARD CAPABILITY COMMANDS

Command	Function	When issued
:( ) command log on	Begins a session, executes the enclosed MPE command, and ends the session upon completion of the command.	S
:ABORT	Aborts the current program.	B
:ALTLOG	Alters the attributes of an existing logging identifier	J,S,B,P
:ALTSEC	Changes security provisions for a file	J,S,B,P
:APL	Accesses the APL subsystem.	J,S
:ASSOCIATE	Gives a user operator control of a device	J,S,B,P
:BASIC	Calls BASIC/3000 interpreter.	J,S
:BASICGO	Compiles, prepares, and executes a BASIC/3000 program.	J,S
:BASICOMP	Compiles a BASIC/3000 program.	J,S
:BASICPREP	Compiles and prepares a BASIC/3000 program.	J,S
:BUILD	Creates a new file.	J,S,B,P
:BYE	Terminates a session.	S
:COBOL	Compiles a COBOL/3000 program.	J,S
:COBOLGO	Compiles, prepares, and executes a COBOL/3000 program.	J,S
:COBOLPREP	Compiles and prepares a COBOL/3000 program.	J,S
:COMMENT	Inserts comment into command stream.	J,S,B,P
:CONTINUE	Disregards job-error condition.	J
:DATA	Defines data from outside standard input stream. Cannot be read on $STDINX file. Acceptable for device recognition.	J,S
:DEBUG	Invokes the MPE debug facility.	S,P
:DISASSOCIATE	Removes the control of a device from a user	J,S,B,P
:DISMOUNT	Causes a volume set that was mounted by the user to be dismounted.	J,S,B
:DSLINE	Opens or closes communication line with DS/3000.	J,S
:DSTAT	Displays the current status of the disc drives on the system.	J,S,B,P
:EDITOR	Calls the EDITOR.	J,S
:ELSE	Provides an alternate execution sequence for an IF statement.	J,S,B
:ENDIF	Terminates an IF block.	J,S,B
:EOD	Denotes end of data. Cannot be read on $STDINX file.	J,S
:EOF	Simulates hardware end-of-file on input stream from any device	J,S
:EOJ	Denotes end of batch job. Cannot be read on $STDINX file.	J
:FILE	Defines or redefines a file's characteristics.	J,S,B,P
:FORTGO	Compiles, prepares, and executes a FORTRAN/3000 program.	J,S
:FORTPREP	Compiles and prepares FORTRAN/3000 program.	J,S
:FORTRAN	Compiles a FORTRAN program.	J,S
:FREERIN	Deallocates a global RIN, and returns it to RIN pool.	J,S
:GETLOG	Establishes a logging identifier on the system	J,S,B,P
:GETRIN	Acquires a global RIN.	J,S,B,P
:HELLO	Initiates a session. Acceptable for device recognition. Requires 1A capability class.	S
:HELP	Access the HELP subsystem.	J,S,B
:IF	Used to control the execution sequence of a job.	J,S,B

Command	Function	When issued
:JOB	Initiates a batch job. Requires BA capability class.	J,S
:LISTF	Lists descriptions of files.	J,S,B,P
:LISTLOG	Lists active logging identifiers	J,S,B,P
:LISTVS	Produces a formatted listing of volume set definition information.	J,S,B,P
:MOUNT	Requests the console operator to mount a volume set.	J,S,B
:MRJE	Initiates execution of the Multi-leaving Remote Job Entry (MRJE) facility.	J,S
:PREP	Prepares a compiled program into segmented form.	J,S
:PREPRUN	Prepares and executes a program.	J,S
:PTAPE	Reads a paper tape without X-OFF control.	S,B,P
:PURGE	Deletes a file from the system.	J,S,B,P
:RECALL	Displays all pending console REPLY messages	J,S,B,P
:REDO	Allows the user to edit a command entry.	S,B
:RELEASE	Temporarily suspends all security provisions for a file.	J,S,B,P
:RELLOG	Removes a logging identifier from the system	J,S,B,P
:REMOTE	Establishes communication between a local computer and a remote computer.	S
:RENAME	Renames a file.	J,S,B,P
:REPORT	Displays total accounting information for a log-on group.	J,S,B,P
:RESET	Resets a formal file designator.	J,S,B,P
:RESETDUMP	Disables the MPE stackdump facility.	J,S,B,P
:RESTORE	Restores a complete fileset, stored off-file.	J,S,B,P
:RESUME	Resumes an interrupted program.	B,S
:RJE	Calls the 2780/3780 Emulator.	J,S

Command	Function	When issued
:RPG	Compiles an RPG/3000 program.	J,S
:RPGGO	Compiles, prepares, and executes an RPG/3000 program.	J,S
:RPGPREP	Compiles and prepares an RPG/3000 program.	J,S
:RUN	Loads and executes a program.	J,S
:SAVE	Changes a file to permanent status. Requires SF capability for saving files.	J,S,B,P
:SECURE	Restores suspended security provisions for a file.	J,S,B,P
:SEGMENTER	Calls MPE Segmenter.	J,S
:SETCATALOG	Causes the command interpreter to search a catalog of user-defined commands and to establish a directory entry for each command in the catalog	J,S,B
:SETDUMP	Enables the MPE stackdump facility on abort.	J,S,B,P
:SETJCW	Scans the JCW table for a specified JCW name and updates the value of this JCW.	J,S,B,P
:SETMSG	Disables or enables receipt of user or operator messages at standard list device.	J,S,B,P
:SHOWCATALOG	Lists user-defined command (UDC) files.	J,S,B
:SHOWDEV	Reports status of input/output devices.	J,S,B,P
:SHOWIN	Reports status of input device files.	J,S,B,P
:SHOWJCW	Displays the current state of a job control word.	J,S,B,P
:SHOWJOB	Displays job/session status.	J,S,B,P
:SHOWLOG-STATUS	Displays status information about currently opened log files	J,S,B,P
:SHOWME	Reports job/session status.	J,S,B,P
:SHOWOUT	Reports status of output device files.	J,S,B,P
:SHOWTIME	Displays current date and time-of-day.	J,S,B,P

Command	Function	When issued
:SPEED	Changes input speed or output speed of terminal.	S,B,P
:SPL	Compiles an SPL/3000 program.	J,S
:SPLGO	Compiles, prepares, and executes an SPL/3000 program.	J,S
:SPLPREP	Compiles and prepares an SPL/3000 program.	J,S
:STORE	Stores a set of files off-line.	J,S,B,P
:STREAM	Spools batch jobs or data in session or job mode.	J,S,B,P
:TELL	Transmits a message.	J,S,B,P
:TELLOP	Transmits a message from the user to the computer operator.	J,S,B,P
:VINIT	Accesses the VINIT sub-system to perform on-line conditioning and formatting of serial discs and private volumes.	J,S
:VSUSER	Prints a listing of all users of a currently mounted volume set.	J,S,B

STANDARD CAPABILITY COMMANDS
(Segmenter Commands)

Command	Function	When issued
-ADDRL	Adds a procedure to an RL.	J,S
-ADDSL	Adds a segment to an SL.	J,S
-ADJUSTUSLF	Adjusts directory space in a user subprogram library (USL) file.	J,S
-AUXUSL	Designates a source of RBM input for COPY command.	J,S
-BUILDRL	Creates a permanent, formatted RL file.	J,S
-BUILDSL	Creates a permanent, formatted SL file.	J,S
-BUILDUSL	Creates a temporary, formatted USL file.	J,S
-CEASE	Deactivates one or more entrypoints in a USL.	J,S
-CLEANSL	Copies the currently managed SL to a new SL file, removing inactive segments	J,S
-CLEANUSL	Copies the currently managed USL to a new USL file, removing inactive segments	J,S
-COPY	Copies an RBM or segment from one USL to another.	J,S
-COPYSL	Same as CLEANSL except allows user to expand SL space by a given percentage	J,S
-COPYUSL	Same as CLEANUSL except allows user to expand USL space by a given percentage	J,S
-EXIT	Exits from Segmenter, returning control to MPE command interpreter.	J,S
-EXPANDUSLF	Changes length of a USL file.	J,S
-HIDE	Sets an RBM internal flag on.	J,S
-INITUSLF	Initializes buffer for a USL file to the empty state.	J,S
-LISTRL	Lists the procedures in an RL.	J,S
-LISTSL	List specific segments in the SL	J,S
-LISTUSL	List specific segments in USL.	J,S
-NEWSEG	Changes the segment name of an RBM.	J,S
-PREPARE	Prepares RBMS from a USL into a program file.	J,S
-PURGERBM	Deletes one or more RBMs from a USL.	J,S
-PURGERL	Deletes an entrypoint or a procedure from an RL.	J,S
-PURGESL	Deletes an entrypoint or a segment from an SL.	J,S
-REVEAL	Sets an RBM internal flag off.	J,S
-RL	Designates an RL for management.	J,S
-SL	Designates an SL for management.	J,S
-USE	Activates one or more RBM entrypoints.	J,S
-USL	Designates a USL for management.	J,S

SYSTEM MANAGER CAPABILITY COMMANDS

Command	Function	When issued
:ALTACCT	Changes an account's characteristics.	J,S,B,P
:ALTVSET	Modifies volume set definitions.	J,S,B,P
:LISTACCT	Lists attributes of an account.	J,S,B,P
:NEWACCT	Creates a new account.	J,S,B,P
:NEWVSET	Defines private volume sets and classes.	J,S,B,P
:PURGEACCT	Removes an account and users from the system's or the volume set's directory.	J,S,B,P
:PURGEVSET	Deletes an existing volume set.	J,S,B,P
:REPORT	Displays an account's resource usage.	J,S,B,P
:RESETACCT	Resets resource-use counters for an account and its groups.	J,S,B,P

ACCOUNT MANAGER CAPABILITY COMMANDS

Command	Function	When issued
:ALTGROUP	Changes a group's attributes.	J,S,B,P
:ALTUSER	Changes a user's attributes.	J,S,B,P
:LISTACCT	Lists attributes of user's log-on account.	J,S,B,P
:LISTGROUP	Lists attributes of a group in user's log-on account.	J,S,B,P
:LISTUSER	Lists attributes of a user in log-on account.	J,S,B,P
:NEWGROUP	Creates a new group in log-on account.	J,S,B,P
:NEWUSER	Creates a new user in log-on account.	J,S,B,P
:PURGEGROUP	Removes a group from the system's or the volume set's directory.	J,S,B,P
:PURGEUSER	Deletes a user from log-on account.	J,S,B,P
:REPORT	Displays resource-usage counts for log-on account and its groups.	J,S,B,P

SYSTEM SUPERVISOR CAPABILITY COMMANDS

Command	Function	When issued
:ALLOCATE	Permanently allocates a program or procedure in virtual memory.	J,S
:DEALLOCATE	Removes a program or procedure from virtual memory.	J,S,P
:JOBPRI	Sets or changes the priority for batch jobs or sessions.	J,S,B,P
:QUANTUM	Changes a circular subqueue time-quantum.	J,S,B,P
:RESTORE	Returns files to the system.	J,S,B,P
:RESUMELOG	Resumes logging following suspension caused by an error.	J,S,B,P
:SHOWLOG	Displays log file status.	J,S,B,P
:SHOWQ	Displays scheduling subqueue information.	J,S,B,P
:STORE	Stores disc files onto magnetic tape or serial disc.	J,S,B,P
:SWITCHLOG	Closes the current log file, and creates and opens a new log file.	J,S,B,P
:SYSDUMP	Starts configurator dialog and copies MPE to magnetic tape or serial disc.	J,S

CONSOLE OPERATOR COMMANDS

Command	Function
:ABORTIO	Aborts pending I/O requests for a device.
:ABORTJOB	Aborts a job or session.
:ACCEPT	Permits a device to accept job/sessions and/or data.
:ALLOW	Grants a user access to a specific operator command
:ALTJOB	Alters attributes of waiting job or session.
:ALTSPOOLFILE	Alters attributes of output spooling files.
:BREAKJOB	Suspends an executing job.

CONSOLE OPERATOR COMMANDS

Command	Function
:CONSOLE	Changes the system console from its current device to another job-accepting (non-DS) terminal
:DELETESPOOL-FILE	Deletes a spooled device file.
:DISALLOW	Prohibits a user access to a specified operator command.
:DOWN	Removes a device from normal system use.
:DOWNLOAD	Downloads information to an output device.
=DSLINE	Enables or disables the data communications link under control of the DS/3000 subsystem.
:GIVE	Assigns a DOWNed device to the diagnostics.
:HEADOFF	Stops header/trailer output to a device.
:HEADON	Resumes header/trailer output to a device.
:JOBFENCE	Defines input priorities.
:JOBSECURITY	Controls the availability of certain job commands to a user.
:LDISMOUNT	Logically dismounts a private volume set/class (UV capability required).
:LIMIT	Limits the number of concurrently running jobs/sessions.
:LMOUNT	Logically mounts a private volume/class on a non-system domain disc drive.
:LOG	Starts, restarts, stops User Logging (LG capability required).
=LOGOFF	Aborts all jobs/sessions and prevents further log-ons by all except HIPRI jobs/sessions.

Command	Function
=LOGON	Enables job/session processing following a LOGOFF command.
=MPLINE	Enables or disables the data communications link under control of the MTS/3000 subsystem.
=MRJE	Enables or disables the data communications link under control of the MRJE/3000 subsystem.
:OUTFENCE	Defines priorities for output spooled files.
:REFUSE	Disallows jobs/sessions and/or data on a designated device.
:REPLY	Replies to a pending console request.
=REPLY	Same as :REPLY
:RESUMEJOB	Resumes a suspended job.
:RESUMESPOOL	Resumes a spooled device.
=SHUTDOWN	Closes down the operating system.
:STARTSPOOL	Initiates spooling of a device.
:STOPSPOOL	Terminates spooling of a device.
:STREAMS	Enables or disables the users' ability to submit job/session and/or data streams.
:SUSPENDSPOOL	Causes a spooled device to stop operation.
:TAKE	De-assigns a device that was GIVEn to the diagnostics.
:UP	Allows a DOWNed device to function again.
:VMOUNT	Enables or disables the private volumes facility.
:WARN	Sends an urgent message to jobs and sessions.
:WELCOME	Defines the message users receive when they log on the system.

Appendix C: MPE III Intrinsics

All MPE III intrinsics (system procedures) are summarized below, listed alphabetically. Any special capabilities required to call a particular intrinsic are noted.

Intrinsic	Function
ACCEPT	Accepts (and completes) the request received by the preceding GET intrinsic call.
ACTIVATE	Activates a process. (Requires PH capability.)
ADJUSTUSLF	Adjust directory space in a USL file.
ALTDSEG	Changes the size of an extra data segment. (Requires DS capability.)
ARITRAP	Enables or disables internal interrupt signals from all hardware arithmetic traps.
ASCII	Converts a number from binary to ASCII code.
BINARY	Converts a number from ASCII to binary code.
CALENDAR	Returns the calendar date.
CAUSEBREAK	Requests a session break.
CLOCK	Returns the time of day.
CLOSELOG	Closes access to the logging facility.
COMMAND	Executes an MPE command programmatically.
CREATE	Creates a process. (Requires PH capability.)
CTRANSLATE	Converts a string of characters from EBCDIC to ASCII or from ASCII to EBCDIC.
DASCII	Converts a value from double-word binary to ASCII code.
DBINARY	Converts a number from ASCII to double-word binary value.
DEBUG	Sets breakpoints and modifies or displays stack or register contents.
DLSIZE	Changes size of DL-DB area.
DMOVIN	Copies block from data segment to stack. (Requires DS capability.)
DMOVOUT	Copies block from stack to data segment. (Requires DS capability.)
EXPANDUSLF	Changes length of a USL file.
FATHER	Requests PIN of father process. (Requires PH capability.)

Intrinsic	Function
FCARD	Drives the HP 7260A Optional Mark Reader.
FCHECK	Requests details about file input/ output errors.
FCLOSE	Closes a file.
FCONTROL	Performs various control operations on a file or terminal device.
FERRMSG	Returns message corresponding to FCHECK error number.
FGETINFO	Requests access and status information about a file.
FINDJCW	Searches the job control word table for a specified job control word (JCW).
FLOCK	Dynamically locks a file.
FMTCALENDAR	Converts the calendar date obtained with the CALENDAR intrinsic.
FMTCLOCK	Converts the time of day obtained with the CLOCK intrinsic.
FMTDATE	Converts calendar date and time of day obtained with the CALENDAR and CLOCK intrinsics.
FOPEN	Opens a file.
FPOINT	Resets the logical record pointer for a sequential disc file.
FREAD	Reads a logical record from a sequential file.
FREADDIR	Reads a logical record from a direct-access disc file.
FREADLABEL	Reads a user file label.
FREADSEEK	Prepares, in advance, for reading from a direct-access file.
FREEDSEG	Releases an extra data segment. (Requires DS capability.)
FREELOCRIN	Frees all local RINs from allocation to a job.
FRELATE	Declares a file pair interactive or duplicative.
FRENAME	Renames a disc file.
FSETMODE	Activates or deactivates file access modes.
FSPACE	Spaces forward or backward on a file.
FUNLOCK	Dynamically unlocks a file.
FUPDATE	Updates a logical record residing in a disc file.

Intrinsic	Function
FWRITE	Writes a logical record to a sequential file.
FWRITEDIR	Writes a logical record to a direct-access disc file.
FWRITELABEL	Writes a user file label.
GENMESSAGE	Accesses the message system.
GET	Receives the next request from the remote master program.
GETDSEG	Creates an extra data segment. (Requires DS capability.)
GETJCW	Fetches contents of job control word.
GETLOCRIN	Acquires a local RIN.
GETORIGIN	Determines source of process activation call. (Requires PH capability.)
GETPRIORITY	Reschedules a process. (Requires PH capability.)
GETPRIVMODE	Dynamically enters privileged mode. (Requires PM capability.)
GETPROCID	Requests PIN of a son process. (Requires PH capability.)
GETPROCINFO	Requests status information about a father or son process. (Requires PH capability.)
GETUSERMODE	Dynamically returns to non-privileged mode. (Requires PM capability.)
INITUSLF	Initializes a buffer for a USL file to the empty state.
IODONTWAIT	Initiates completion operations for an I/O request.
IOWAIT	Initiates completion operations for an I/O request.
KILL	Deletes a process. (Requires PH capability.)
LOADPROC	Dynamically loads a library procedure.
LOCKGLORIN	Locks a global RIN.
LOCKLOCRIN	Locks a local RIN.
MAIL	Tests mailbox status. (Requires PH capability.)
MYCOMMAND	Parses (delineates and defines parameters) for user-supplied command image.
PAUSE	Suspends the calling process for a specified number of seconds.
PCHECK	Returns an integer code specifying the completion status of the most recently executed slave program-to-program intrinsic.
PCLOSE	Terminates the remote slave program's process.
PCONTROL	Transmits a tag field to the remote slave program and receives a tag field back from the slave.
POPEN	Initiates and activates a slave process in a remote HP 3000 and initiates program-to-program communication with the slave program.
PREAD	Sends a read request to the remote slave program asking the slave to send a block of data back to the master.
PRINT	Prints character string on job/session list device.
PRINTFILEINFO	Prints a file information display on the job/session list device.
PRINTOP	Prints a character string on operator's console.
PRINTOPREPLY	Prints a character string on the operator's console and solicits a reply.
PROCTIME	Returns a process's accumulated central-processor time.
PTAPE	Accepts input from tapes not containing X-OFF control characters.
PUTJCW	Puts the value of a particular job control word (JCW) in the JCW table.
PWRITE	Sends a block of data to the remote slave program.
OPENLOG	Provides access to a logging facility.
QUIT	Aborts a process.
QUITPROG	Aborts a program.
READ	Reads an ASCII string from an input device.
READX	Reads an ASCII string from an input device.
RECEIVEMAIL	Receives mail from another process. (Requires PH capability.)
REJECT	Rejects a request received by the preceding GET intrinsic call and returns an optional tag field back to a remote master program.

Intrinsic	Function
RESETCONTROL	Resets a terminal to accept a CONTROL-Y signal.
RESETDUMP	Disables the abort stack analysis facility.
SEARCH	Searches an array for a specified entry or name.
SENDMAIL	Sends mail to another process. (Requires PH capability.)
SETDUMP	Enables the stack analysis facility.
SETJCW	Sets bits in job control word.
STACKDUMP	Dumps selected parts of stack to a file.
SUSPEND	Suspends a process. (Requires PH capability.)
SWITCHDB	Switches DB register pointer. (Requires PM capability.)
TERMINATE	Terminates a process.

Intrinsic	Function
TIMER	Returns system-timer bit-count.
UNLOADPROC	Dynamically unloads a library procedure.
UNLOCKGLORIN	Unlocks a global RIN.
UNLOCKLOCRIN	Unlocks a local RIN.
WHO	Returns user attributes.
WRITELOG	Writes a record to a logging file.
XARITRAP	Arms the software arithmetic trap.
XCONTRAP	Arms or disarms the CONTROL-Y trap.
XLIBTRAP	Arms or disarms the software library trap.
XSYSTRAP	Arms or disarms the sytem trap.
ZSIZE	Changes size of Z-DB area.

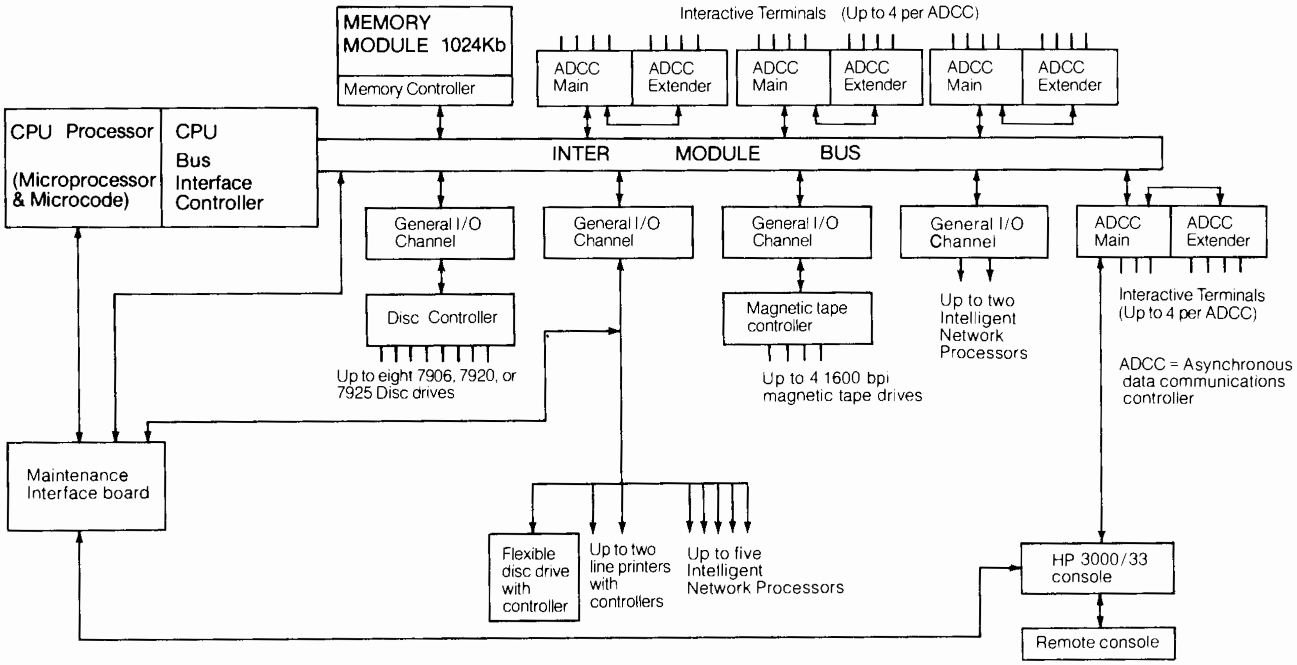
Appendix D: HP 3000 Series 30 & Series 33 Hardware Features

This appendix summarizes the hardware features of the HP 3000 Series 30 and Series 33. To learn more about the HP 3000 architecture, refer to Chapter 1 System Introduction and Chapter 6 System Architecture.

The HP 3000 Series 30 and Series 33 are designed around independent elements that are organized together in a central bus structure. The elements of the system consist of a central processor that operates through a bus interface controller, memory arrays with a memory controller, general I/O channels, asynchronous data communications controllers, and a bus system that enables communications between the I/O elements. Also, the system includes a system console, system front panel, and a maintenance facility. Peripheral elements attach to the system through the general I/O channels. Interactive terminals attach to the system through the asynchronous data communications controllers.

The CPU is controlled by a specially designed Hewlett-Packard silicon-on-sapphire (SOS) microprocessor to allow a great deal of flexibility in the machine instruction set. The HP 3000 also employs high-speed, semiconductor memory modules which use automatic fault detection and correction.

The design of the HP 3000 Series 30 and Series 33 hardware provides an efficient and powerful foundation upon which the software is built, as illustrated in Figure D-1. The configuration of hardware modules and peripheral devices is easily changed to accommodate system expansion.



MEMORY MODULE 1024Kb
Memory Controller
Interactive Terminals (Up to 4 per ADCC)
ADCC Main
ADCC Extender
ADCC Main
ADCC Extender
ADCC Main
ADCC Extender
CPU Processor
(Microprocessor & Microcode)
CPU Bus Interface Controller
INTER MODULE BUS
General I/O Channel
General I/O Channel
General I/O Channel
General I/O Channel
ADCC Main
ADCC Extender
Disc Controller
Up to eight 7906, 7920, or 7925 Disc drives
Magnetic tape controller
Up to 4 1600 bpi magnetic tape drives
Up to two Intelligent Network Processors
Interactive Terminals (Up to 4 per ADCC)
ADCC = Asynchronous data communications controller
Maintenance Interface board
Flexible disc drive with controller
Up to two line printers with controllers
Up to five Intelligent Network Processors
HP 3000/33 console
Remote console

Figure D-1. HP 3000 SERIES 33 HARDWARE ORGANIZATION (MAXIMUM CONFIGURATION)

Central processing unit (CPU)

The significant features of the HP 3000 Series 30 and Series 33 central processing unit (CPU) are listed in Table D-1.

Table D-1. HP 3000 CPU Features

Architecture

- Hardware-implemented stack
- Separation of code and data
- Non-modifiable, re-entrant code
- Variable-length code segmentation
- Virtual memory for code
- Dynamic relocatability of programs

Implementation

- Microprogrammed SOS/CMOS CPU
- 90 nanosecond cycle time; microinstructions execute in 3 to 7 cycles (4.5 cycle average)
- Automatic restart after power failure
- Inter-module bus
- Overlapping CPU and I/O operations

Instructions

- 214 powerful instructions
- Instructions are 8, 16, or 32 bits in length
- 16- and 32-bit integer arithmetic
- 32- and 64-bit floating point arithmetic
- 28-digit packed decimal arithmetic
- Special instructions that optimize the efficiency of the operating system.

The CPU converts an instruction in the current instruction register (CIR) into a starting address for the microcode contained in a read-only memory (ROM), and determines various initial conditions required for executing the instruction. As the current instruction is being executed, the next instruction is fetched and placed into the next instruction register (NIR). Upon completion of the current instruction, the contents of NIR are loaded into CIR and the cycle is repeated. This "pipelining" of the current instruction execution with the next instruction-fetch improves throughput by overlapping operations. The HP 3000 Series 30 and Series 33 instruction set is presented in Appendix E. Instructions are 16 or 32 bits in length except stack operations, which are 8-bit instructions. These include a variety of memory reference, branch, arithmetic and data manipulation instructions that operate on integer, real, logical, packed decimal, character and string data. Floating point arithmetic can be performed in single precision (32 bits) or double precision (64 bits), integer arithmetic in 16-bit and 32-bit lengths, and packed decimal instructions extend to 28 digits of precision. In addition, there are a number of instructions designed to aid in creating the multiprogramming environment of the system. These include procedure call and exit instructions and others which implement various operating system functions previously done in software.

Firmware storage and control consists of microcode, stored in read-only memory (ROM), and associated logic control. Microcode routines control the operation of the instruction decoder and the hardware processor, in order to create the HP 3000 operating environment. The control storage has a cycle time of 90 nanoseconds and a microinstruction execution time which is variable from 270 to 630 nanoseconds.

The hardware processor consists of an arithmetic-logic unit, shifting network, 27 specific purpose registers—13 of which are accessible to user programs—and related data manipulating and testing logic. Since the HP 3000 architecture (see Chapter 6) is structured on code segments and data segments, most of the CPU registers are used for defining the segment limits and operating elements within the segments. As shown in Figure D-2, three of the CPU registers point to locations in a code segment defined as the current code segment. Six of the registers point to locations in a data segment defined as the current data segment. Table D-2 lists all 27 registers and their associated functions.

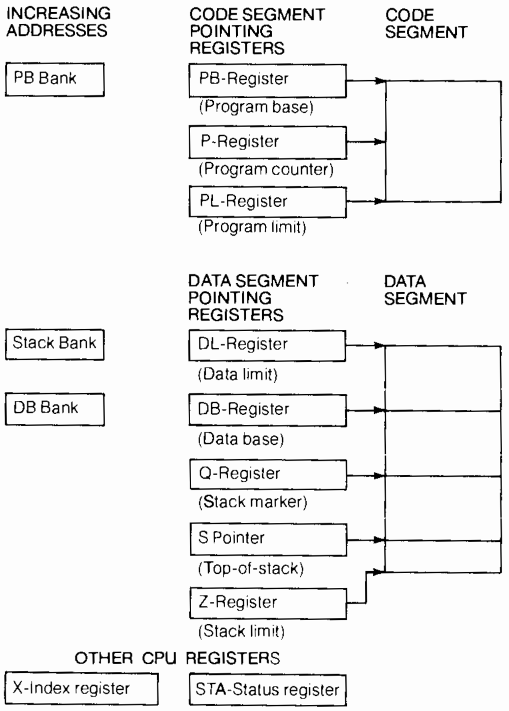


Figure D-2. HP 3000 CPU REGISTERS

Table D-2 Series 30 and Series 33 Hardware Registers

Registers accessible to user programmers

Register	Function
PB	Code Segment Pointers
P	
PL	
DL	Stack Pointers
DB	
Q	
SB	
S	
DST	
Z	
TOSA	Top of Stack Registers
TOSB	
X	Index Register
STA	Status Register

Registers dedicated for system use

Register	Function
CIR	Current Instruction Register
NIR	Next Instruction Register
SP0	Scratch Pad, Flag, and Interrupt Registers
SP1	
SP2	
SP3	
SP4	
SP5	
ISR	
CNTR	
MEMA	Memory Address and Data Registers
BUSD	
RAR	Firmware Address Register

The two top of stack registers are of special interest. In order to improve execution speed, up to two elements from the top of a data stack may be contained in these registers. This allows many functions to be treated as register-to-register operations rather than the slower speed memory-to-register or register-to-memory type operations. These registers are manipulated by the CPU, and their use is fully transparent.

Main memory

The significant features of the HP 3000 Series 30 and Series 33 main memory are listed in Table D-3.

Table D-3: HP 3000 Series 30 and Series 33 Main Memory Features

- High-speed, semiconductor, random access memory
- Automatic fault detection and fault correction
- Memory sizes ranging from 256k bytes to 1 megabyte
- Write: 860 nsec minimum cycle time
- Read: 430 nsec access, 860 nsec cycle time
- 0.5- to 4-hour rechargable battery packs to maintain memory data during power failure.

The HP 3000 Series 30 and Series 33 use high-speed, semiconductor, random access memory (16k MOS RAM) which provides single-bit correction and all double bit fault detection.

Due to the modular design of the HP 3000, any system can be easily expanded from one memory size to another.

The word length transmitted over the bus is 16 bits. In the memory modules the word length is expanded to 22 bits; 16 bits of data and six bits for automatic fault detection and correction.

All detected memory faults are automatically logged to special (non-user) storage. MPE periodically reads this storage and writes the information to the disc file. This file is accessed by an HP Customer Engineer, from a terminal on the system, while performing preventative maintenance. If memory chips have a history of failures, they are replaced during maintenance.

Operating power for the memory modules is supplied by a rechargeable battery pack in the semiconductor memory power supply. When the power supply input voltage is removed, battery power is available for up to 4 hours (depending on memory size and battery condition) to maintain memory data.

Communication between the CPU, memory and I/O modules is carried over the inter-module bus (IMB). Because the CPU generates greater than 90 percent of the bus activity, it is given continuous access to the bus and relinquishes control to the I/O channels only on request.

The IMB has separate address and data paths, each with handshake controls that operate in a master/slave mode to transfer data between modules. The CPU talks to memory and to the I/O system and always functions as a master. The I/O channels function as masters to memory but become slaves when talking with the CPU. To access memory, the I/O channels must request the bus through a priority structure. Any channel request will cause the CPU to relinquish control of the IMB so that the request can be serviced.

Input/output

All access to input/output devices is by way of the device-independent MPE file system. All location of data, buffering, data transfers, and deblocking are handled automatically by MPE. When you ask to read a named file, you are only implicitly specifying the actual disc address of the file; the file system determines the explicit address and performs the read. At another level, when you ask the file system for a certain type of device by specifying a device class name (e.g., magnetic tape, line printer, etc.), the file system takes care of allocating an actual device. If you must have actual contact with specific devices, you may address them directly. Below this single, flexible interface is a powerful and carefully balanced hardware/ software input/output system.

All devices can be operated concurrently (within system bandwidth). Peripherals that fail are taken off-line from the operating system by operator command.

Peripheral I/O hardware

HP 3000 Series 30 and Series 33 peripheral I/O hardware consists of the general I/O channels (GIC), Hewlett-Packard Interface Bus (HP-IB), device controllers, and the peripheral units. Asynchronous data communications controllers interface log-on and data-entry terminals. Table D-4 lists the features which this peripheral I/O hardware offers.

Table D-4: HP 3000 Peripheral I/O Hardware Features

- 4 ports per Asynchronous Data Communications Controller
- Up to 8 terminal controllers per system
- Type 103, 112 and 202S modem support

When an I/O request is issued, the device driver in the CPU assembles the channel program, then issues a Start I/O Program (SIOP) instruction to one of two types of channels on the Inter-Module Bus: the General I/O Channel (GIC) and the Asynchronous Data Communications Controller (ADCC). The GIC is the hardware I/O channel which provides the electrical interface between the computer system via the IMB and peripheral devices connected to the Hewlett-Packard Interface Bus (HP-IB). The HP-IB is HP's implementation of the IEEE standard 488-1975 interface, used on the Series 30 and Series 33 to connect peripheral devices to the channel. The HP-IB consists of eight data lines and eight control lines. The ADCC provides a bit-serial data interface between the computer system and terminals. The two channels operate in a similar manner; however, the GIC has a DMA facility to permit high-speed transfer of large blocks of data, while the ADCC can transfer data only one character at a time.

General I/O channel

The General I/O Channel is the primary channel for communication between the CPU and the I/O devices other than terminals. Each GIC controls a Hewlett-Packard Interface Bus (HP-IB) and translates I/O commands from the CPU into the proper HP-IB protocol. Nearly all transactions with I/O devices are accomplished without software interrupts, since I/O is achieved with channel programs. Software is responsible for setting up a channel program, but the execution of this program is performed by the CPU's channel microcode. The CPU's channel microcode is devoted to I/O tasks and implements the necessary algorithms for decoding the channel instructions and effecting the required I/O operations. Once the channel program is running, device control and data flow are normally carried to completion with no software intervention and without altering the system environment. If special situations arise, software may alter the program or even halt execution.

Several devices may simultaneously need service, and the CPU must decide which one will receive attention. First, all channels are polled, and the highest priority GIC with a device request pending is chosen. The CPU then obtains from that channel the number of the highest priority device needing service. Once the device number is determined, execution of the channel program will begin. The CPU fetches each channel instruction and breaks it down into several IMB commands addressed to the proper GIC. The GIC interprets these commands and directs them onto the HP-IB device.

The GIC contains DMA (Direct Memory Access) hardware which allows large records of data to be transferred at the maximum speed of the HP-IB (about 1 Mb/sec.). The channel microcode enables the device and then initializes the DMA hardware on the GIC. After initial addressing of a device to talk or listen, the CPU relinquishes control of the IMB and allows the GIC to perform its function through DMA operation. During this time the GIC becomes the master of the bus and memory and controls traffic flow. On a read operation the DMA hardware will read the bytes, pack them into words and place them directly into memory, all without assistance from the CPU. The CPU is free to service other devices while DMA is in progress. Upon completion of a DMA transfer, the GIC returns to a slave condition and awaits the next operation.

Asynchronous data communications controller

The Asynchronous Data Communications Controller (ADCC) is the second channel type used in the system. This channel performs for terminals essentially the same functions as the GIC but not in the same manner. Data is transferred from memory to the ADCC in parallel form, then converted to a serial bit stream for transmission over the RS-232C lines to the devices. Information being read from a device is in serial form and is converted to eight-bit bytes for transfer to memory.

Two types of ADCC boards may be used, the Main ADCC and the Extender ADCC. Each board contains four ports for connection to devices through RS-232C data communications lines. The Main ADCC supports full duplex (103, 212, and 202S type modems in the U.S.) only; Extender ADCC boards are required for European half duplex modem support. The Main ADCC is used when four or fewer devices are connected to a channel. The Extender ADCC extends the device capability of the channel to eight. Most of the control circuitry is on the Main ADCC. For this reason, the Main ADCC is required for the Extender ADCC to function.

When more than eight devices are to be attached to ADCC channels, additional Main ADCCs are required, since each ADCC can accommodate only one Extender.

Unlike the GIC, the ADCC does not have a DMA facility and therefore cannot be a master of the IMB or of memory. Also, terminals on the channel do not respond to a parallel poll. As a result, the ADCC is always a slave and must be directly controlled by the CPU through the use of channel programs. Circuitry on the ADCC decodes address information relating to channel and devices, and selects the correct device for operation.

The ports on the ADCC (Main and Extender) may be either hardwired to devices or to modems.

Device controller

The device controller is the hardware linkage between a peripheral device and the computer system. Its primary function is to translate I/O commands from a general I/O channel to the unique signals required to control a particular device. When an I/O program is in execution, the device controller responds to and requests service from the general I/O channel. The device controller also generates interrupts when required by some device condition or by channel command.

Device reference table (DRT)

Device controllers are identified by a logical device number which is used to access the device reference table (DRT). The DRT is known to both hardware and software containing, among other things, a pointer to the start of the SIO program for each device controller. Each device controller connects to a General I/O Channel (GIC). Certain device controllers may control several logical devices. In such cases, each logical device attached to the controller is addressed separately using a unit number assigned when the device is installed.

Data service and interrupt priorities

In addition to a logical device number, there are two other characteristic numbers associated with each device controller. These are the data service priority and interrupt priority. In the Series 30 and Series 33, both of these are determined by the logical device number in the DRT: the lower the number, the higher the priority.

Interrupt system

The interrupt system provides for up to 105 external interrupt levels. When interrupts occur, the microprogrammed interrupt handler identifies each interrupt and grants control to the highest priority interrupt. Current operational status is saved by the microprogram, which then sets up the interrupt processing environment and transfers control to the interrupt routine.

Interrupt routines operate on a common stack (interrupt control stack) which is known to both hardware and software. This feature permits nesting of interrupt routines in the case of multiple interrupts, thus allowing higher priority devices to interrupt lower priority devices.

The interrupt system also provides for 20 internal interrupts (for user errors, system violations, hardware faults, and power fail/restart) plus fourteen traps for arithmetic errors and illegal use of instructions.

Peripherals

The peripheral devices used on the HP 3000 Series 30 and Series 33 are connected primarily to GICs, while the ADCC is reserved solely for terminals. Peripherals attached to GICs through the HP-IB include the system disc drive and controller (which can have up to 7 additional drives "daisy-chained" from it), flexible disc drive, line printers, and magnetic tape drives. For a complete configuration of the supported peripherals on the HP 3000 Series 30 and Series 33, you are referred to the current HP 3000 Price/Configuration Guide.

Automatic restart after power failure

An integral part of the HP 3000 Series 30 and Series 33 is a power fail/automatic restart capability. When the system AC line voltage falls below 10% of nominal voltage, the system initiates a powerfail warning (PFW). During PFW the system (hardware and MPE) writes all register contents to a reserved section of main memory, activities in the system are successfully completed, and then the power down signal is generated and the system is shut down. The battery back-up power supply refreshes main memory and ensures its validity for up to 4 hours, depending on memory size and battery condition.

The system is automatically restarted when all power supply voltages reach 90% of their normal values and all register values are automatically restored and processing resumes.

Remote system verification program (RSVP)

The Series 30 and Series 33 are designed to be both extremely reliable and easy to service. The packaging makes all system components as accessible as possible. The power supplies and flexible disc units are mounted on sliding "rails" for easy removal and servicing. All power distribution is through quick release connectors rather than cumbersome screw terminal strips.

A totally new feature that enhances serviceability both in hardware and software is the use of the system console as the maintenance console. Through a new Maintenance Interface board and maintenance mode software loaded through a data cartridge in the system console, HP field personnel will have a complete maintenance display in English and octal values on the system console. Values such as the contents of all registers, dynamically selected memory contents (16 words at a time), and system status displays are available quickly from keystrokes entered on the console keyboard.

The first use of the maintenance console is by the customer: a system self-test is provided with the HP 3000 on a terminal data cartridge. In less than two minutes the diagnostic will check out all hardware components involved in a system "cold-load." Faults are isolated to the module level, with concise, yet easy to understand messages printed on the console CRT display. Because of the simplicity and ease of use of this self test, you must run the system self-test prior to calling Hewlett-Packard for hardware maintenance.

If a service call is necessary, HP Customer Engineers and Operating System Specialists can use the console CRT display to inquire into the status of diagnostics initiated from the console system and even into the status of hardware registers for detailed trouble shooting.

Making this new maintenance console even more valuable is the ability to transmit the display and control functions to a remote HP 2645 terminal via a modem and telephone link. With this facility, the CE on site can call the HP Service Office and have a Specialist get "on-line" to the system over the telephone via the remote system console/maintenance console. The CE loads the remote maintenance code data cartridge into the console (15 seconds), then switches the modem (user-supplied) to the console using a switch built into the terminal junction panel to establish the telephone link. The Specialist now has a duplicate display of the HP 3000 system console/ maintenance console display, with the ability to send the CE and/or system manager messages that are not transmitted to the computer. This "remote maintenance console" facility is a standard part of all Series 30 and Series 33 systems. You are required to have a Bell 103 type modem (300 baud) or Bell 212 (1200 baud) for use in connecting the console to the phone line. Throughout the procedure, complete control over access to the system remains with you. As a back-up capability to the system console, the system front panel is hardwired to perform console control commands (only) as well.

Diagnostics

Several levels of diagnostic software help identify and diagnose hardware problems in the HP 3000 computer system. The levels of diagnostics are:

- Verification programs for peripherals run under MPE
- Stand-alone diagnostics to verify all system modules
- PC board microdiagnostics in PROM for CPU, Memory, and I/O

Appendix E: HP 3000 Series 30 and Series 33 Machine Instructions

HEWLETT PACKARD

STACK OP INSTRUCTIONS

ADAX	Add A to X
ADBX	Add B to X
ADD	Add A to B
ADXA	Add X to A
ADXB	Add X to B
AND	Logical AND of A and B
BTST	Test byte on TOS and set CC
CAB	Rotate A-B-C
CMP	Integer compare B, A and set CC
DADD	Double integer add D, C + B, A
DCMP	Double integer compare and set CC
DDEL	Double delete TOS
DDIV	Double integer divide
DDUP	Double duplicate TOS
DECA	Decrement A
DECB	Decrement B
DECX	Decrement X
DEL	Delete TOS
DELB	Delete B
DFLT	Float a double integer
DIV	Integer divide B by A
DIVL	Divide long integer C, B ÷ A
DMUL	Double integer multiply
DNEG	Double integer negate
DSUB	Double integer subtract D, C − B, A
DTST	Test double word on TOS and set CC
DUP	Duplicate TOS
DXCH	Double exchange
DZRO	Push double zero onto stack
FADD	Floating point add, D, C + B, A
FCMP	Floating point compare and set CC
FDIV	Floating point divide D, C ÷ B, A
FIXR	Fix and round
FIXT	Fix and truncate
FLT	Float an integer
FMPY	Floating point multiply
FNEG	Floating point negate
FSUB	Floating point subtract D,C—B,A
INCA	Increment A
INCB	Increment B
INCX	Increment X
LADD	Logical add A + B
LCMP	Logical compare B, A and set CC
LDIV	Logical divide C, B ÷ A
LDXA	Load X into A
LDXB	Load X into B
LMPY	Logical multiply B × A
LSUB	Logical subtract B − A
MPY	Multiply integers, integer product
MPYL	Multiply integers, long integer product
NEG	Integer negate
NOP	No operation
NOT	Logical complement TOS
OR	Logical OR of A, B
STAX	Store A into X
STBX	Store B into X
SUB	Integer subtract B − A
TEST	Test TOS and set CC
XAX	Exchange A and X
XBX	Exchange B and X
XCH	Exchange A and B
XOR	Logical exclusive OR of A, B
ZERO	Push integer zero onto stack
ZROB	Zero B
ZROX	Zero X

SHIFT INSTRUCTIONS

ASL	Arithmetic shift left
ASR	Arithmetic shift right
CSL	Circular shift left
CSR	Circular shift right
DASL	Double arithmetic shift left
DASR	Double arithmetic shift right
DCSL	Double circular shift left
DCSR	Double circular shift right
DLSL	Double logical shift left
DLSR	Double logical shift right
LSL	Logical shift left
LSR	Logical shift right
QASL	Quadruple arithmetic shift left
QASR	Quadruple arithmetic shift right
TASL	Triple arithmetic shift left
TASR	Triple arithmetic shift right
TNSL	Triple normalizing shift left

LEGEND

TOS	Top of stack	A	Top of stack	D	Location below C
CC	Condition Code	B	Location below A	DB	Data Base
X	Index Register	C	Location below B	DL	Data Limit

FIELD AND BIT INSTRUCTIONS

DPF	Deposit field, A bits to B	TCBC	Test and complement bit, set CC
EXF	Extract specified field, right-justify	TRBC	Test and reset bit, set CC
SCAN	Scan bits	TSBC	Test and set bit, set CC
TBC	Test specified bit and set CC		

BRANCH INSTRUCTIONS

BCC	Branch on specified CC	BRO	Branch on TOS odd (bit 15 = 1)
BCY	Branch on carry	CPRB	Compare range and branch
BNCY	Branch on no carry	DABZ	Decrement A, branch if zero
BNOV	Branch on no overflow	DXBZ	Decrement X, branch if zero
BOV	Branch on overflow	IABZ	Increment A, branch if zero
BR	Branch unconditionally	IXBZ	Increment X, branch if zero
BRE	Branch on TOS even (bit 15 = 0)		

MOVE INSTRUCTIONS

CMPB	Compare bytes in two memory blocks	MVB	Move bytes in memory, addresses +/−
MABS	Move using absolute addresses	MVBL	Move words from DB+ to DL+ area
MDS	Move using data segments	MVBW	Move bytes while of specified type
MFDS	Move from data segment	MVLB	Move words from DL+ to DB+ area
MOVE	Move words in memory, addresses +/−	SCU	Scan bytes until test or terminal byte
MTDS	Move to data segment	SCW	Scan bytes while equal to test byte

PRIVILEGED MEMORY REFERENCE INSTRUCTIONS

LDEA	Load double word from extended address	PSTA	Privileged store into absolute address
LSEA	Load single word from extended address	SDEA	Store double word into extended address
LST	Load from system table	SSEA	Store single word into extended address
PLDA	Privileged load from absolute address	SST	Store into system table

IMMEDIATE INSTRUCTIONS

ADDI	Add immediate to integer in A	LDXI	Load X immediate
ADXI	Add immediate to X	LDXN	Load X negative immediate
ANDI	Logical AND immediate with A	MPYI	Multiply immediate with A
CMPI	Compare A with immediate, set CC	ORI	Logical OR immediate with A
CMPN	Compare A with negative immediate	SBXI	Subtract immediate from X
DIVI	Divide immediate into A	SUBI	Subtract immediate from A
LDI	Load immediate to TOS	XORI	Logical exclusive OR immediate
LDNI	Load negative immediate to TOS		

REGISTER CONTROL INSTRUCTIONS

ADDS	Add operand to stack pointer	SETR	Set specified registers from stack
PSHR	Push specified registers onto stack	SUBS	Subtract operand from stack pointer
RCLK	Read clock	XCHD	Exchange DB and TOS
SCLK	Store clock		

PROGRAM CONTROL AND SPECIAL INSTRUCTIONS

DISP	Dispatch	PCN	Push CPU code (% 10)
EXIT	Exit from procedure	PSDB	Pseudo interrupt disable
HALT	Halt	PSEB	Pseudo interrupt enable
IXIT	Interrupt exit	RSW	Push cold load chan/dev
LLBL	Load label	SCAL	Subroutine call
LLSH	Linked list search	SXIT	Exit from subroutine
LOCK	NOP	UNLK	NOP
PAUS	Pause, interruptable	XEQ	Execute stack word
PCAL	Procedure call		

I/O INSTRUCTIONS

CBKP	Clear breakpoint	SCLR	Set system clock limit
HIOP	Halt I/O program	SED	Set enable/disable external interrupts
INIT	Initialize I/O channel	SMSK	Set device mask
RCCR	Read system clock	TOFF	Hardware timer off
RIOC	Read I/O channel	TON	Hardware timer on
RMSK	Read device mask	WIOC	Write I/O channel
SBKP	Set breakpoint	IOCL	I/O clear
SEML	Semaphore Load	ROCL	Channel roll call
STRT	Programmatic warm start	MCS	Memory controller read status
SIOP	Start I/O channel program	DUMP	Programmatic dump

LOOP CONTROL INSTRUCTIONS

MTBA	Modify variable, test against limit, branch	TBA	Test variable against limit, branch
MTBX	Modify X, test against limit, branch	TBX	Test X against limit, branch

MEMORY ADDRESS INSTRUCTIONS

ADDM	Add memory to TOS	LDX	Load X
CMPM	Compare TOS with memory	LOAD	Load word onto stack
DECM	Decrement memory	LRA	Load relative address onto stack
INCM	Increment memory	MPYM	Multiply TOS by memory
LDB	Load byte onto stack	STB	Store byte on TOS into memory
LDD	Load double word onto stack	STD	Store double on TOS into memory
LDPN	Load double from program, negative	STOR	Store TOS into memory
LDPP	Load double from program, positive	SUBM	Subtract memory from TOS

EXTENDED INSTRUCTION SET

	Extended-Precision Floating Point		Decimal Arithmetic
EADD	ADD	ADDD	Decimal Add
ECMP	Compare	CMPD	Decimal compare
EDIV	Divide	CVAD	ASCII to decimal conversion
EMPY	Multiply	CVBD	Binary to decimal conversion
ENEG	Negate	CVDA	Decimal to ASCII conversion
ESUB	Subtract	CVDB	Decimal to binary conversion
		DMPY	Double logical multiply
		MPYD	Decimal multiply
		NSLD	Decimal normalizing left shift
		SLD	Decimal left shift
		SRD	Decimal right shift
		SUBD	Decimal subtract

Appendix F: HP 3000 Series III Hardware Features

The design of the HP 3000 Series III hardware provides an efficient and powerful foundation upon which the software is built, as illustrated in Figure F-1. Communication between modules occurs over the central data bus. The central processing unit (CPU) and the input/output processor (IOP), although independent of one another, share a common module address. This is resolved by giving the IOP a higher priority in the case where both processors concurrently request use of the bus. The CPU is controlled by a specially designed microprocessor to allow a great deal of flexibility in the machine instruction set. The HP 3000 also employs high-speed, semiconductor memory modules which use automatic fault detection and correction with no loss in performance.

The basic structure of independent modules organized around a central data bus permits high-speed internal data rates. When not communicating over the bus, each module can run independently at its own speed.

The presence of a separate IOP bus which is totally dedicated to input/output data transfers means that the HP 3000 can always respond immediately to the needs of I/O devices regardless of what transfers are currently in progress between the various system modules. It also means that many I/O operations can be handled concurrently with CPU, main memory, and high-speed selector channel operations.

Data may be transferred directly between main memory and high-speed peripheral devices in block mode by way of high-speed selector channels connected to the central data bus. For lower speed devices, data may be multiplexed on a word-by-word basis by way of the IOP and a multiplexer channel. In both cases the I/O channels execute in parallel with CPU operations. Direct control of devices attached to the IOP bus is also possible through the use of the CPU's direct I/O instructions.

The configuration of hardware modules and peripheral devices is easily changed to accommodate system expansion.

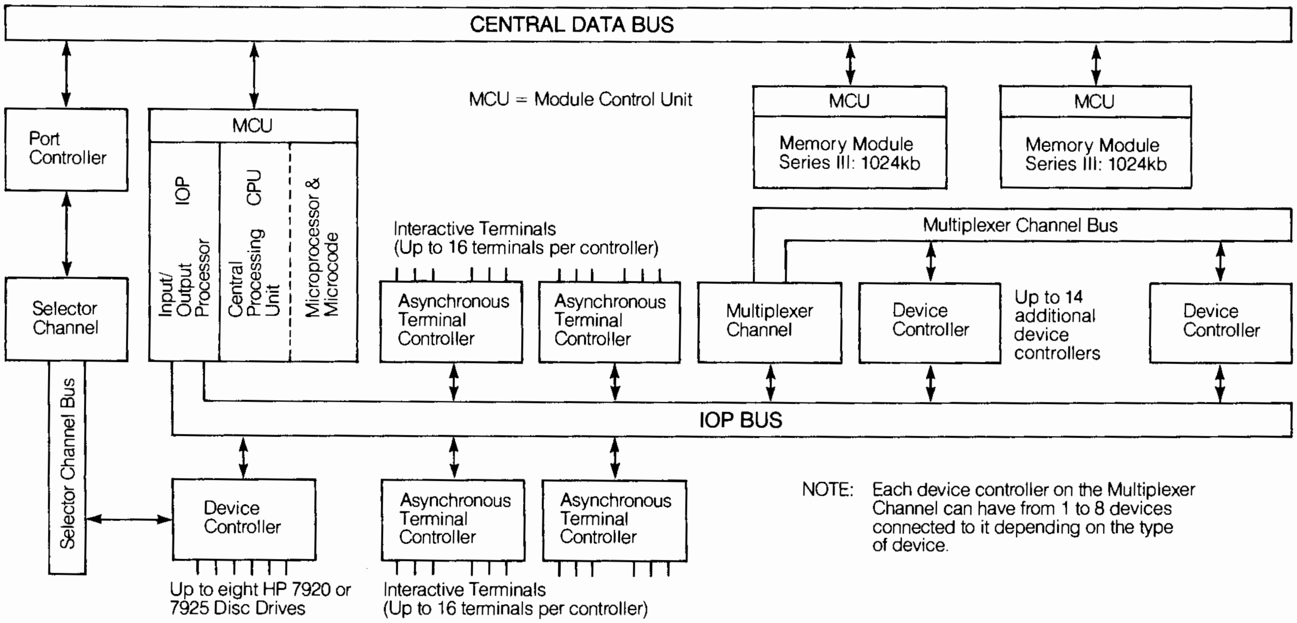

CENTRAL DATA BUS
MCU = Module Control Unit
MCU
Memory Module
Series III: 1024kb
MCU
Memory Module
Series III: 1024kb
Port Controller
MCU
IOP
Input/Output Processor
CPU
Central Processing Unit
Microprocessor & Microcode
Interactive Terminals
(Up to 16 terminals per controller)
Multiplexer Channel Bus
Selector Channel
Asynchronous Terminal Controller
Asynchronous Terminal Controller
Multiplexer Channel
Device Controller
Up to 14 additional device controllers
Device Controller
IOP BUS
Selector Channel Bus
Device Controller
Up to eight HP 7920 or 7925 Disc Drives
Asynchronous Terminal Controller
Asynchronous Terminal Controller
Interactive Terminals
(Up to 16 terminals per controller)
NOTE: Each device controller on the Multiplexer Channel can have from 1 to 8 devices connected to it depending on the type of device.

FIGURE F-1. HP 3000 HARDWARE ORGANIZATION (MAXIMUM CONFIGURATION)

Central processing (CPU)

The significant features of the HP 3000 central processing unit (CPU) are listed in Table F-1.

TABLE F-1. HP 3000 CPU Features

Architecture

- Hardware-implemented stack
- Separation of code and data
- Non-modifiable, re-entrant code
- Variable-length code segmentation
- Virtual memory for code
- Dynamic relocatability of programs

Implementation

- Microprogrammed CPU
- 175 nanosecond microinstruction time
- Automatic restart after power failure
- Central data bus
- Bus parity checking
- Concurrent CPU and I/O operations

Instructions

- 209 powerful instructions
- All instructions except stack operations are 16 bits in length (stack operations may be packed two per word)
- 16- and 32-bit integer arithmetic
- 32- and 64-bit floating point arithmetic
- 28-digit packed decimal arithmetic
- Special instructions that optimize the efficiency of the operating system.

The three major components of the CPU are the instruction decoder, firmware storage and control, and hardware processor.

The instruction decoder unit converts an instruction in the current instruction register (CIR) into a starting address for the microcode contained in a read-only memory (ROM), and determines various initial conditions required for executing the instruction. As the current instruction is being executed, the next instruction is fetched and placed into the next instruction register (NIR). Upon completion of the current instruction, the contents of NIR are loaded into CIR and the cycle is repeated. This "pipelining" of the current instruction execution with the next instruction-fetch improves throughput by overlapping operations. The HP 3000 III Series instruction set is presented in Appendix G. All instructions are 16 bits in length except stack operations, which are 8-bit instructions. These include a variety of memory reference, branch, arithmetic and data manipulation instructions that operate on integer, real, logical, packed decimal, character and string data. Floating point arithmetic can be performed in single precision (32 bits) or double precision (64 bits), integer arithmetic in 16-bit and 32-bit lengths, and packed decimal instructions extend to 28 digits of precision. In addition, there are a number of instructions designed to aid in creating the multi-programming environment of the system. These include procedure call and exit instructions and others which implement various operating system functions previously done in software.

Firmware storage and control consists of microcode, stored in read-only memory (ROM), and associated control logic. Microcode routines control the operation of the instruction decoder and the hardware processor, in order to create the HP 3000 operating environment—including the 209 instructions available to the programmer. The control storage has a cycle time of 175 nanoseconds and an average microinstruction execution time of 175 nanoseconds.

The hardware processor consists of an arithmetic-logic unit, shifting network, 38 specific purpose registers—20 of which are accessible to user programs, and related data manipulating and testing logic. Since the HP 3000 architecture (see Chapter 6) is structured on code segments and data segments, most of the CPU registers are used for defining the segment limits and operating elements within the segments. As shown in Figure F-2, three of the CPU registers point to locations in a code segment defined as the current code segment. Six of the registers point to locations in a data segment defined as the current data segment. Table F-2 lists all 38 registers and their associated functions. The four top of stack registers are of special interest. In order to improve execution speed, up to four elements from the top of a data stack may be contained in these registers. This allows many functions to be treated as register-to-register operations rather than the slower speed memory-to-register or register-to-memory type operations. These registers are manipulated by the CPU, and their use is fully transparent.

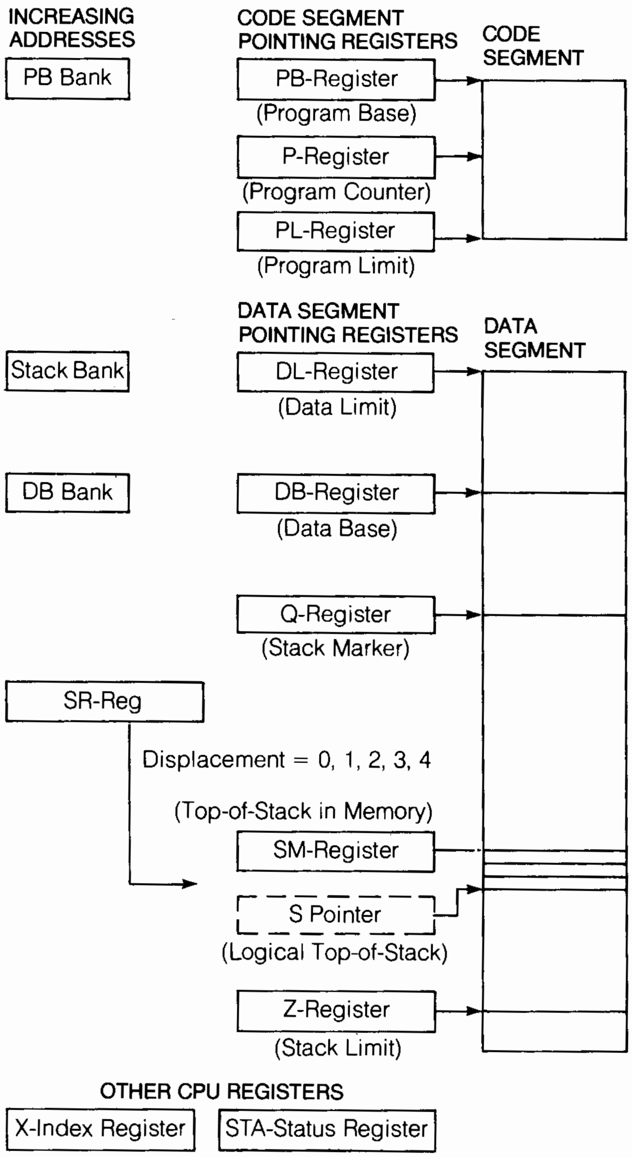


Figure F-2. HP 3000 CPU REGISTERS

TABLE F-2. HP 3000 Hardware Registers

Registers accessible to user programmers

Register	Function
PB	Code Segment Pointers
P	
PL	
PB-Bank	
DL	Stack Pointers
DB	
Q	
SM	
SR	
Z	
DB-Bank	
S-Bank	
RA	Top of Stack Registers
RB	
RC	
RD	
X	Index Register
STA	Status Register
SWCH	Switch Register
PCLK	Program Clock Register

Registers dedicated for system use

CIR	Current Instruction Register
NIR	Next Instruction Register
SP0	Scratch Pad, Flag, and Interrupt Registers
SP1	
SP2	
SP3	
CTR	
ABS-Bank	
CPX1	
CPX2	
MOD	
IOA	I/O Registers
IOD	
ACOR	Memory Address and Data Registers
DCOR	
OPND	
RAR	Firmware Address Registers
SAVE	

Main memory

The significant features of the HP 3000 main memory are listed in Table F-3.

TABLE F-3. HP 3000 Main Memory Features

- High-speed, semiconductor, random access memory
- Automatic fault detection and fault correction
- Memory sizes ranging from 256k bytes to 2 megabytes
- Write: 700 nsec minimum cycle time
- Read: 350 nsec access, 700 nsec cycle time
- 45-90 minute rechargeable battery packs to maintain memory data during power failure

The HP 3000 uses high-speed, semiconductor, random access memory (16k MOS RAM) which provides single-bit fault correction. All double bit faults are detected on the Series III. The parity system is retained with this feature, thus maintaining integrity over the various buses.

Due to the modular design of the HP 3000, any system can be easily expanded from one memory size to another. When there is more than one megabyte of memory in the Series III, it is divided into two modules: one megabyte in the first module and the remainder in the second. These modules can operate concurrently which improves execution time.

The word length transmitted over the bus is 17 bits—16 bits of data (one word or two bytes) and one parity bit. In the memory modules the word length is expanded to 22 bits—16 bits of data and six bits for automatic fault detection and correction.

All detected memory faults are automatically logged to special (non-user) storage. MPE periodically reads this storage and writes the information to the disc file. This file is accessed by an HP Customer Engineer, from a terminal on the system, while performing preventative maintenance. If memory chips have a history of failures, they are replaced during maintenance.

Operating power for the memory modules is supplied by rechargeable battery packs in the semiconductor memory power supplies. When the power supply input voltage is removed, battery power is available for up to 90 minutes (depending on memory size and battery condition) to maintain memory data.

The memory modules interface with other system modules by way of the central data bus. The other system modules may request transfers of data to or from the memory modules on that bus. Operation of the memory modules with other system modules on the central data bus is controlled by module control units, one for each module.

Input/output

All access to input/output devices is by way of the device-independent MPE file system. All location of data, buffering, data transfers, and deblocking are handled automatically by MPE. When you ask to read a named file, you are only implicitly specifying the actual disc address of the file, the file system determines the explicit address and performs the read. At another level, when you ask the file system for a certain type of device by specifying a device class name (e.g., magnetic tape, line printer, etc.), the file system takes care of allocating an actual device. If you must have actual contact with specific devices, you may address them directly. Below this single, flexible interface is a powerful and carefully balanced hardware/software input/output system.

All devices can be operated concurrently (within system bandwidth). Peripherals that fail are taken off-line from the operating system by operator command.

There are two distinct means of implementing I/O: direct I/O (DIO) and programmed I/O (SIO).

Direct I/O (DIO): Direct I/O allows for single word transfers of data, status, or control information between a device connected to the IOP bus and the top of your data stack.

Programmed I/O (SIO): Programmed I/O can be used with devices on either the selector channel (high-speed devices) or the multiplexer channel (medium-to low-speed devices). With programmed I/O, the CPU simply issues a "Start I/O" instruction to the device controller. The device controller, in turn, initiates the SIO program for the particular device which then runs under the control of the selector channel or under the control of the multiplexer channel and the IOP. The SIO program uses a unique set of commands (optimized for I/O operations) to transfer information between main memory and the external device. Both the selector channel and the multiplexer channel (via the IOP) have direct access to main memory. The SIO program and CPU processing run concurrently until the appropriate I/O command terminates the device transfer. This I/O command can also cause an interrupt signal to be sent to the CPU, thus informing the CPU that the I/O task is complete.

Input/output processor (IOP): The input/output processor controls the IOP bus and interrupt lines, providing the communications and data path between the CPU and I/O devices for direct I/O operations and interrupt processing. It also offers a data path between memory and I/O devices for programmed I/O operations.

I/O operations are divided into three categories, direct I/O, programmed I/O, and interrupt processing. Direct I/O operations take place as a result of I/O instructions executed by the CPU; they result in transfer of a word of information between the CPU and an I/O device through the IOP or cause a control function to take place in the I/O system. Devices connected to a multiplexer (see Peripheral I/O Hardware) may use programmed I/O. Once started, programmed I/O operations can, (through the execution of I/O programs stored in memory) transfer block(s) of data between I/O devices and memory, and perform other device control functions without further CPU intervention or attention. The IOP also accepts interrupt requests from the device controllers, interrogates them by means of a poll line to find the highest priority request, and sends an interrupt signal together with the number of the interrupting device to the CPU.

Peripheral I/O hardware: HP 3000 Series III peripheral I/O hardware consists of the selector channel, IOP bus, interrupt lines, multiplexer channel, device controllers, and the peripheral units. Asynchronous terminal controllers interface log-on and data-entry terminals. Table F-4 lists the features which this peripheral I/O hardware offers.

TABLE F-4. HP 3000 Peripheral I/O Hardware Features

- 16 ports per terminal controller
- Up to 4 terminal controllers per system
- Support for 16 device controllers on the multiplexer channel
- Options for type 202S modems
- Type 103A, 113B, 202T and 212 type modem support

Asynchronous terminal controller: The asynchronous terminal controller (ATC) is designed to interface terminals to the HP 3000 via the IOP bus. Up to 16 terminals (including the system console) can be interfaced via one ATC, and up to 4 ATCs may be connected to the system. Terminals can be hardwired or connected through type 103A2, 103A3, 103J, 113B, 202S/T, 212 and Vadic VA 3400 modems. Terminals interfaced through the ATC can be configured to the Multiprogramming Executive as data entry terminals under user program control, or as log-on terminals accessing all the capabilities of the system. Since the system console occupies one terminal port, fifteen additional terminal ports are available before expansion to another controller is necessary.

Selector channel: The selector channel interfaces high speed peripheral devices to the HP 3000. Connecting to the central data bus through a port controller, it can accommodate up to eight disc drives connected to one device controller. The selector channel data transfers bypass the IOP completely to provide data transfer rates of up to 2.86 million bytes per second for a single device. Unlike the multiplexer channel (see next section) which switches between device controllers on demand, based on hardware priority, the selector channel maintains the connection for one device controller until it has completed the I/O program. Thus only one I/O program is current at a time for one selector channel. All selector channel data transfers are performed in block mode and the data is applied directly to main memory via the central data bus.

Multiplexer channel: The multiplexer channel acts as a switch to enable one of the 1 through 16 device controllers connected to it to transfer one word of data to or from memory via the IOP, then to allow another controller—based on priority—to perform its transfer. Operating in conjunction with the IOP, the multiplexer allows the device controllers connected to it to run concurrently, interleaving their transfers on a word-by-word basis. By multiplexing, asynchronous cumulative data rates of up to 1,038,000 bytes per second (inbound) and 952,000 bytes per second (outbound) are possible. Data from the multiplexer channel is supplied directly to the IOP for transfer to main memory via the central data bus.

A solid state memory in the multiplexer is divided into 16 sections, one for each device controller. Typically, this memory contains the current I/O program word and related information for each device. When a device is selected for service, the multiplexer executes the indicated operation (or portion thereof) in conjunction with the device controller.

Device controller: The device controller is the hardware linkage between a peripheral device and the computer system. Its primary function is to translate programmed I/O commands from a multiplexer channel or selector channel (or direct I/O commands from the I/O processor) to the unique signals required to control a particular device. When an I/O program is in execution, the device controller responds to and requests service from the multiplexer channel or selector channel. The device controller also generates interrupts when required by some device condition or by direct or programmed command.

Device reference table (DRT): Device controllers are identified by a device number which is used to access the device reference table (DRT). The DRT is known to both hardware and software containing, among other things, a pointer to the start of the SIO program for each device controller. Since there can be a maximum of 125 entries in this table, the HP 3000 logic design allows for up to 125 device controllers in its I/O system (the actual limitation is the 7-bit I/O address bus). Certain device controllers may control several devices. In such cases, each device attached to the controller is addressed separately using a unit number assigned when the device is installed.

Data service and interrupt priorities: In addition to a device number, there are two other characteristic numbers associated with each device. These are the data service priority and interrupt priority. Each of these values is completely independent of the other, and neither is related to the physical location of devices or controllers. This mutual independence of characteristics provides the following advantages:

1. Device numbers can be assigned consecutively, starting with number 3 and proceeding up to the last assigned device in the system. When a new device controller is added, it is merely assigned the next available number (or any vacant number).
2. A new device added to the system may have its controller connected anywhere in the interrupt or data service priority chains, independent of physical location within the cabinet.
3. Since data service priority and interrupt priority are independent of each other, a device which requires a high data transfer rate but interrupts infrequently (such as a disc) may be assigned a high data service priority and a low interrupt priority.

Interrupt system: The interrupt system provides for up to 125 external interrupt levels. When interrupts occur, the microprogrammed interrupt handler identifies each interrupt and grants control to the highest priority interrupt. Current operational status is saved by the microprogram, which then sets up the interrupt processing environment and transfers control to the interrupt routine.

Interrupt routines operate on a common stack (interrupt control stack) which is known to both hardware and software. This feature permits nesting of interrupt routines in the case of multiple interrupts, thus allowing higher priority devices to interrupt lower priority devices.

The interrupt system also provides for 22 internal interrupts (for user errors, system violations, hardware faults, and power fail/restart) plus fourteen traps for arithmetic errors and illegal use of instructions.

Peripherals: Peripheral devices receive output data for storage or display, or supply input data to the computer. Usually, one device controller controls one peripheral device; however, some device controllers are capable of controlling several devices.

Hewlett-Packard furnishes available peripherals as complete I/O subsystems (including the device, interface, cables, etc.) to facilitate system expansion.

For a complete configuration of the supported peripherals on the HP 3000 Series III, refer to the current HP 3000 Price/Configuration Guide.

Automatic restart after power failure

An integral part of the HP 3000 power supply is a power fail/automatic restart capability. When the system AC line voltage falls below 170 volts, the system initiates a power fail warning (PFW). During PFW the system (hardware and MPE) writes all register contents to a reserved section of main memory, activities in the system are successfully completed, and then the power down signal is generated and the system is shut down. The battery back-up power supply refreshes main memory and ensures its validity for 45 to 90 minutes, depending on memory size and battery condition.

The system is automatically restarted when all power supply voltages reach 90% of their normal values. There is a minimum restart delay of 0.6 seconds following the power on signal before the system is restarted, during which another power failure is possible. After the delay all register values are automatically restored and processing resumes.

Diagnostics

Several levels of diagnostic software help identify and diagnose hardware problems in the HP 3000 computer system. The levels of diagnostics are:

- Verification programs for peripherals run under MPE
- Stand-alone diagnostics to verify all system modules
- Panel microdiagnostics in PROM for CPU, memory, and I/O

Appendix G: HP 3000 Series III Machine Instructions

HEWLETT hp PACKARD

STACK OP INSTRUCTIONS

ADAX	Add A to X
ADBX	Add B to X
ADD	Add A to B
ADXA	Add X to A
ADXB	Add X to B
AND	Logical AND of A and B
BTST	Test byte on TOS and set CC
CAB	Rotate A-B-C
CMP	Integer compare B, A and set CC
DADD	Double integer add D, C + B, A
DCMP	Double integer compare and set CC
DDEL	Double delete TOS
DDIV	Double integer divide
DDUP	Double duplicate TOS
DECA	Decrement A
DECB	Decrement B
DECX	Decrement X
DEL	Delete TOS
DELB	Delete B
DFLT	Float a double integer
DIV	Integer divide B by A
DIVL	Divide long integer C, B ÷ A
DMUL	Double integer multiply
DNEG	Double integer negate
DSUB	Double integer subtract D, C − B, A
DTST	Test double word on TOS and set CC
DUP	Duplicate TOS
DXCH	Double exchange
DZRO	Push double zero onto stack
FADD	Floating point add D, C + B, A
FCMP	Floating point compare and set CC
FDIV	Floating point divide D, C ÷ B, A
FIXR	Fix and round
FIXT	Fix and truncate
FLT	Float an integer
FMPY	Floating point multiply
FNEG	Floating point negate
FSUB	Floating point subtract D, C—B, A
INCA	Increment A
INCB	Increment B
INCX	Increment X
LADD	Logical add A + B
LCMP	Logical compare B, A and set CC
LDIV	Logical divide C, B ÷ A
LDXA	Load X into A
LDXB	Load X into B
LMPY	Logical multiply B × A
LSUB	Logical subtract B − A
MPY	Multiply integers, integer product
MPYL	Multiply integers, long integer product
NEG	Integer negate
NOP	No operation
NOT	Logical complement TOS
OR	Logical OR of A, B
STAX	Store A into X
STBX	Store B into X
SUB	Integer subtract B − A
TEST	Test TOS and set CC
XAX	Exchange A and X
XBX	Exchange B and X
XCH	Exchange A and B
XOR	Logical exclusive OR of A, B
ZERO	Push integer zero onto stack
ZROB	Zero B
ZROX	Zero X

SHIFT INSTRUCTIONS

ASL	Arithmetic shift left
ASR	Arithmetic shift right
CSL	Circular shift left
CSR	Circular shift right
DASL	Double arithmetic shift left
DASR	Double arithmetic shift right
DCSL	Double circular shift left
DCSR	Double circular shift right
DLSL	Double logical shift left
DLSR	Double logical shift right
LSL	Logical shift left
LSR	Logical shift right
QASL	Quadruple arithmetic shift left
QASR	Quadruple arithmetic shift right
TASL	Triple arithmetic shift left
TASR	Triple arithmetic shift right
TNSL	Triple normalizing shift left

LEGEND

TOS	Top of stack	A	Top of stack	D	Location below C
CC	Condition Code	B	Location below A	DB	Data Base
X	Index Register	C	Location below B	DL	Data Limit

FIELD AND BIT INSTRUCTIONS

DPF	Deposit field, A bits to B	**TCBC**	Test and complement bit, set CC
EXF	Extract specified field, right-justify	**TRBC**	Test and reset bit, set CC
SCAN	Scan bits	**TSBC**	Test and set bit, set CC
TBC	Test specified bit and set CC		

BRANCH INSTRUCTIONS

BCC	Branch on specified CC	**BRO**	Branch on TOS odd (bit 15 = 1)
BCY	Branch on carry	**CPRB**	Compare range and branch
BNCY	Branch on no carry	**DABZ**	Decrement A, branch if zero
BNOV	Branch on no overflow	**DXBZ**	Decrement X, branch if zero
BOV	Branch on overflow	**IABZ**	Increment A, branch if zero
BR	Branch unconditionally	**IXBZ**	Increment X, branch if zero
BRE	Branch on TOS even (bit 15 = 0)		

MOVE INSTRUCTIONS

CMPB	Compare bytes in two memory blocks	**MVB**	Move bytes in memory, addresses +/−
MABS	Move using absolute addresses	**MVBL**	Move words from DB+ to DL+ area
MDS	Move using data segments	**MVBW**	Move bytes while of specified type
MFDS	Move from data segment	**MVLB**	Move words from DL+ to DB+ area
MOVE	Move words in memory, addresses +/−	**SCU**	Scan bytes until test or terminal byte
MTDS	Move to data segment	**SCW**	Scan bytes while equal to test byte

PRIVILEGED MEMORY REFERENCE INSTRUCTIONS

LDEA	Load double word from extended address	**PSTA**	Privileged store into absolute address
LSEA	Load single word from extended address	**SDEA**	Store double word into extended address
LST	Load from system table	**SSEA**	Store single word into extended address
PLDA	Privileged load from absolute address	**SST**	Store into system table

IMMEDIATE INSTRUCTIONS

ADDI	Add immediate to integer in A	**LDXI**	Load X immediate
ADXI	Add immediate to X	**LDXN**	Load X negative immediate
ANDI	Logical AND immediate with A	**MPYI**	Multiply immediate with A
CMPI	Compare A with immediate, set CC	**ORI**	Logical OR immediate with A
CMPN	Compare A with negative immediate	**SBXI**	Subtract immediate from X
DIVI	Divide immediate into A	**SUBI**	Subtract immediate from A
LDI	Load immediate to TOS	**XORI**	Logical exclusive OR immediate
LDNI	Load negative immediate to TOS		

REGISTER CONTROL INSTRUCTIONS

ADDS	Add operand to stack pointer	**SETR**	Set specified registers from stack
PSHR	Push specified registers onto stack	**SUBS**	Subtract operand from stack pointer
RCLK	Read clock	**XCHD**	Exchange DB and TOS
SCLK	Store clock		

PROGRAM CONTROL AND SPECIAL INSTRUCTIONS

DISP	Dispatch
EXIT	Exit from procedure
HALT	Halt
IXIT	Interrupt exit
LLBL	Load label
LLSH	Linked list search
LOCK	Lock resource
PAUS	Pause, interruptable
PCAL	Procedure call
PCN	Push CPU code number
PSDB	Pseudo interrupt disable
PSEB	Pseudo interrupt enable
RSW	Read switch register
SCAL	Subroutine call
SXIT	Exit from subroutine
UNLK	Unlock resource
XEQ	Execute stack word

I/O INSTRUCTIONS

CIO	Control I/O, direct
CMD	Send command to module, direct
RIO	Read I/O, direct
RMSK	Read device mask
SED	Set enable/disable external interrupts
SIN	Set interrupt
SIO	Start I/O, block transfer
SMSK	Set device mask
TIO	Test I/O, direct
WIO	Write I/O, direct

LOOP CONTROL INSTRUCTIONS

MTBA	Modify variable test against limit, branch
MTBX	Modify X, test against limit, branch
TBA	Test variable against limit, branch
TBX	Test X against limit, branch

MEMORY ADDRESS INSTRUCTIONS

ADDM	Add memory to TOS
CMPM	Compare TOS with memory
DECM	Decrement memory
INCM	Increment memory
LDB	Load byte onto stack
LDD	Load double word onto stack
LDPN	Load double from program, negative
LDPP	Load double from program, positive
LDX	Load X
LOAD	Load word onto stack
LRA	Load relative address onto stack
MPYM	Multiply TOS by memory
STB	Store byte on TOS into memory
STD	Store double on TOS into memory
STOR	Store TOS into memory
SUBM	Subtract memory from TOS

EXTENDED INSTRUCTION SET

Extended-Precision Floating Point

EADD	Add
ECMP	Compare
EDIV	Divide
EMPY	Multiply
ENEG	Negate
ESUB	Subtract

Decimal Arithmetic

ADDD	Decimal add
CMPD	Decimal compare
CVAD	ASCII to decimal conversion
CVBD	Binary to decimal conversion
CVDA	Decimal to ASCII conversion
CVDB	Decimal to binary conversion
DMPY	Double logical multiply
MPYD	Decimal multiply
NSLD	Decimal normalizing left shift
SLD	Decimal left shift
SRD	Decimal right shift
SUBD	Decimal subtract

Appendix H: HP 3000 Guide to Synchronous Modems

Selection of a modem is a critical factor in planning a geographically dispersed network. A wide variety of modems is available, so this guide has been prepared to aid in choosing the proper unit. Hewlett-Packard recommends that the modems listed below be used. If other modems are selected, they must be functionally and electrically identical (at the system interface) with the recommended modems.

Modem selection criteria

HP synchronous modems are fully compatible with the synchronous communication subsystems available on the HP 3000 computers. They are not available in all countries, and your local sales office should be consulted. The selection, installation, and proper functioning of common carrier or third party modems is your responsibility. Hewlett-Packard accepts responsibility for maintaining hardware and software compatibility only with recommended modems. To determine the optimum modem for your system, the following parameters must be considered.

Operating mode: Switched network (dial-up) or leased (private) line.

Type of circuit: Two-wire (half duplex) or four wire (full duplex).

Transmission speed: 1200, 2400, 4800, or 9600 bps.

Line conditioning: Unconditioned or conditioned.

A communications network's performance, in terms of throughput and responsiveness, is heavily influenced by the decisions made concerning these parameters. For low volume networks, low speed communications may be appropriate. Most users normally will select the highest performance service that meets their budgetary requirements. Since modem selection is also based on software related considerations, you should consult your Hewlett-Packard representative regarding the HP communications software and applications which you will be using.

Operating mode: Switched network (dial-up) operation normally is advisable when communications occur only periodically (once or twice per day) and the volume of data to be transmitted is low to moderate. Generally, only two-wire circuits are available for switched networks, with operation limited to 4800 bps or below.

Leased line service is appropriate for higher volumes of data and continuous on-line service.

Type of circuit: Two-wire communications circuits send/receive in one direction at a time, and to reverse the direction of transmission, a "line turnaround" must be performed. Since frequent, time-consuming turnarounds are necessary, two-wire circuits should be selected only for low volume/response applications.

Four-wire circuits can send and receive simultaneously, which eliminates turnaround delays. Most leased line service is four-wire. To maintain responsiveness and the best transmission throughput, four-wire circuits are suggested.

Transmission speed: Transmission speed is the most critical decision to be made in selecting communications facilities for distributed systems networks. Usually the speed of the service, including circuits and modem, is directly associated with the cost; higher speed means higher cost. Availability of service from the common carrier, the volume of traffic, and the need for responsiveness during interactive access are key factors in selecting the transmission speed.

In many areas, only leased-line service is available to 9600 bps on voice-grade circuits.

- In most countries 4800 bps leased-line service is offered.
- Some countries also allow 4800 bps service on a switched circuit (dial-up) basis, although usually with only two-wire circuits.
- Nearly all countries provide both switched and non-switched (leased line) service at 2400 bps.

Line conditioning: By applying certain internal compensation to a communications circuit, the common carrier can remove noise and other degrading characteristics to improve the quality of the circuit. The level of conditioning recommended for the modem should always be used to ensure reliable service.

Recommended modems

Hewlett-Packard modems operate with DS/3000, RJE/3000, MRJE/3000, and MTS/3000. The modems listed below may be used:

FOR CANADA AND THE UNITED STATES

Switched service (dial up)

HP 37210T	(4800 bps)
Bell 201C	(2400 bps)
Bell 208B	(4800 bps)

Leased line service

HP 37210T	(4800 bps)
HP 37220T	(9600 bps)
Bell 201C	(2400 bps)
Bell 208A	(4800 bps)
Bell 209A	(9600 bps)

If you choose other modems you are responsible for working with their supplier to ensure that the units are fully equivalent to the HP recommended equipment.

For international areas

In many countries, the provision of telecommunications services is controlled by a local transmission authority. Check with your local Hewlett-Packard representative regarding the availability of Hewlett-Packard modems. With other modems, the user or Hewlett-Packard must demonstrate satisfactory operation with the locally supplied modem prior to HP's acceptance of responsibility for compatible operation with the supplied communications software.

Appendix I: HP 3000 Communication Subsystem/Synchronous Interface Reference Charts

Communication subsystems

DS/3000—Distributed systems

RJE/3000—Remote Job Entry (2780/3780)

MRJE/3000—Multileaving Remote Job Entry

MTS/3000—Multipoint Terminal Software

Synchronous hardware interfaces

HSI—Hardwired Serial Interface (30360A)

SSLC—Synchronous Single Line Controller (30055A)

INP—Intelligent Network Processor
(30010A, Series II or III; 30020A, Series 30 or 33)

Communications subsystems supported by each hardware interface

	HSI	SSLC	INP	INP
Subsystem	(Series II/III)	(Series II/III)	(Series II/III)	(Series 30/33)
DS	X	X	X	X
RJE		X	X	X
MRJE		X		
MTS		X		

Possible hardware interface links for DS/3000

HSI to HSI (HP 3000 HSI to either HP 3000 HSI or HP 1000 HSI (12889A))

SSLC to SSLC

SSLC to INP

INP to INP

Links may be used in any combination in a DS/3000 Network.

Note. Links using an SSLC are limited to 9600 bps transfer rates.

Maximum data transfer rates supported by synchronous hardware interfaces

Interface	Data Rate (Maximum)
HSI	2.5 Mbps up to 1000′; 1.25 Mbps up to 2000′ (hardwired only)
SSLC	9600 bps over modems
INP	19.2 Kbps over modems; 56 Kbps hardwired or with CCITT V.35 standard interface

Maximum data transfer rates supported by communications subsystems

Subsystem	Data Rate (Maximum)
DS	See hardware interfaces
RJE	19.2 Kbps
MRJE	9600 bps (SSLC only)
MTS	9600 bps (SSLC only)

Index