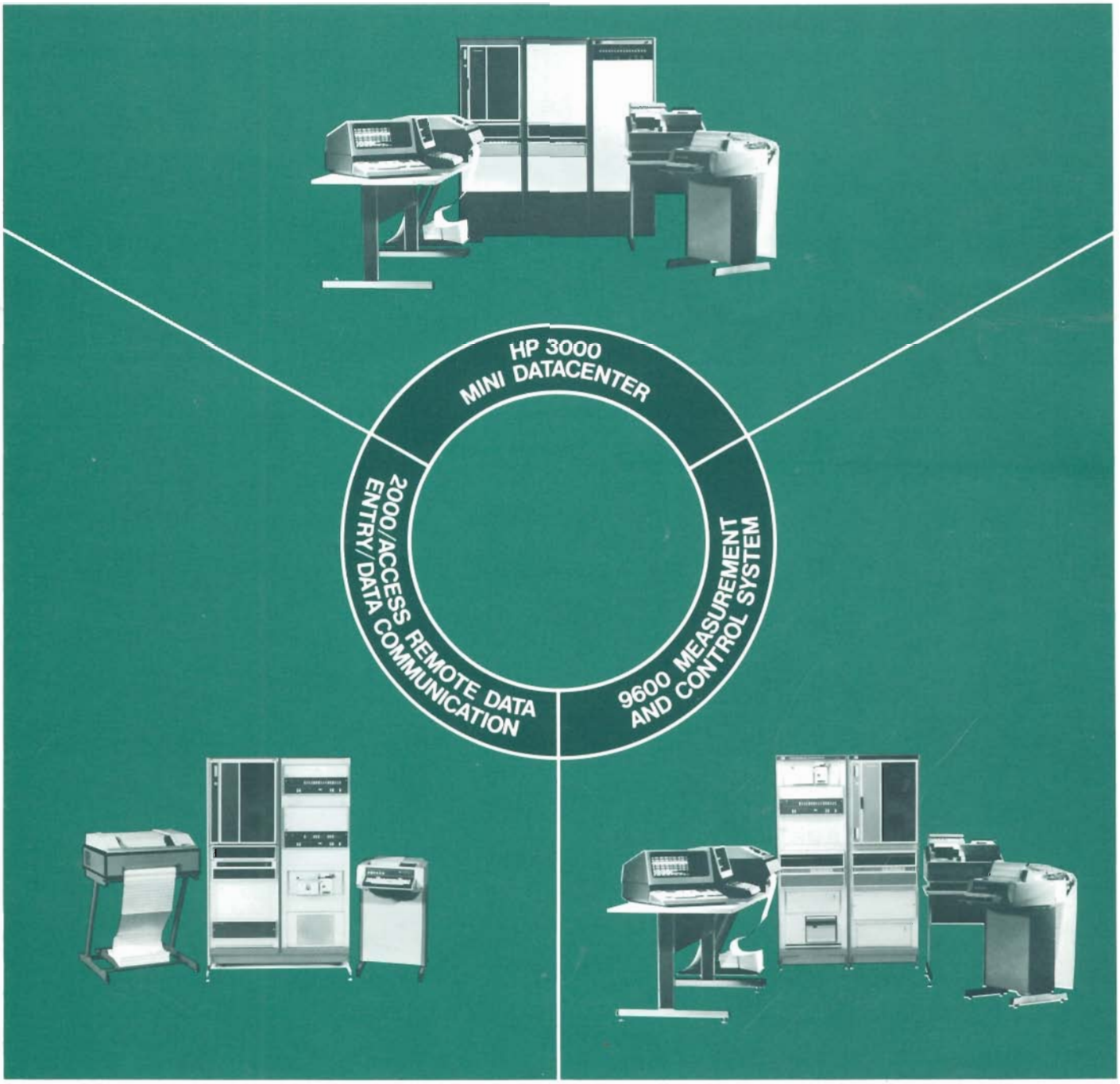
HEWLETT hp PACKARD

ISSUE 5
JANUARY 1, 1976

computer systems COMMUNICATOR



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

HEWLETT-PACKARD
COMPUTER SYSTEMS COMMUNICATOR ORDER FORM

Please Print:

Name _____ Title _____

Company _____

Street _____

City _____ State _____ Zip Code _____

Country _____

☐ MAIL ORDER SUBSCRIPTIONS

BASE SUBSCRIPTION $ _____

_____ ADDITIONAL SUBSCRIPTION(S) $ _____
No.

TOTAL AMOUNT ENCLOSED $ _____

☐ SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS

BASE SUBSCRIPTION (NO ADDITIONAL CHARGE) NAC

_____ ADDITIONAL SUBSCRIPTION(S) $ _____
No.

TOTAL AMOUNT YOU WILL BE BILLED $ _____

FOR HP USE ONLY

SUPPORT OFFICE NUMBER _____ ORDER DATE _____

APPROVED BY _____ C.E. NUMBER _____

SERVICE CONTRACT NUMBER _____ EXPIRATION DATE _____

AUTHORIZED TOTAL NUMBER OF SUBSCRIPTIONS _____

CUSTOMER'S HP OPERATING SYSTEM _____

Printed 10/75

TABLE OF PRORATED $ AMOUNT DUE PER ADDITIONAL SUBSCRIPTION

(Use only for ordering ADDITIONAL SUBSCRIPTION(S) against an existing Software Service Contract)

	Months Remaining in Service Contract											
	1	2	3	4	5	6	7	8	9	10	11	12
Cost of Each ADDITIONAL SUBSCRIPTION	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	11.00	12.00

INSTRUCTIONS FOR ORDERING COMMUNICATOR

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract.

Customers who do not have Software Service Contracts may purchase Mail-Order Subscriptions through HP's Direct Mail Order System.

A. MAIL-ORDER SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
2. Compute amount due:
   a) Annual Base Subscription (8 issues) $ 48.00
   b) _____ Additional Subscriptions* @ $12.00 ea. $ _____
   c) Total Order Amount (a + b) $ _____
   d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Mail check or bank draft with ORDER FORM to:

   HEWLETT-PACKARD CO.
   MAIL ORDER DEPARTMENT
   P.O. DRAWER #20
   MOUNTAIN VIEW, CA. 94043
   U.S.A.

B. SOFTWARE SERVICE CONTRACT SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
2. Compute amount due: (BASE SUBSCRIPTION is at no additional charge.)
   a) Annual Base Subscription (8 issues) $ 0.00
   b) _____ Additional Subscriptions* $ _____
   Prorate the dollar amount to make the ADDITIONAL SUBSCRIPTIONS EXPIRE WITH YOUR Software Service Contract. (SEE TABLE)
   c) Total Order Amount (a + b) $ _____
   d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Forward ORDER FORM to your local HP Customer Engineering Representative. Your order will be approved and forwarded to the appropriate department. You will be billed for any ADDITIONAL SUBSCRIPTIONS by your local HP office.

C. SPECIAL INSTRUCTIONS FOR INTERNATIONAL CUSTOMERS

1. International customers who do not have Software Service Contracts are encouraged to use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the ordering procedures outlined in Instruction A above. Optionally, international customers may purchase the **Communicator** through their local HP Sales and Service Office. The customer should contact his HP Office for the subscription prices in the currency of his country, then complete the Order Form and forward it together with payment to his local HP Customer Engineering Department.

1. International customers with Software Service Contracts should follow the ordering procedure outlined in Instruction B above. If the customer wishes to purchase ADDITIONAL SUBSCRIPTIONS, he should contact the local HP Office for the subscription price in the currency of his country, then submit the ORDER FORM. The customer will be billed for ADDITIONAL SUBSCRIPTIONS by his local HP Office.

*All ADDITIONAL SUBSCRIPTIONS will be sent to the same address as the BASE SUBSCRIPTION.

editor's note

Judging from the many positive comments and suggestions in the editor's mailbox this month, it appears that the **Communicator** is proving to be a very helpful document for HP computer systems users. We appreciate your taking the time to share your reactions with us. Many of you will find your suggestions on content and format implemented in future issues.

All letters referring to a particular product line have been forwarded to the technical editor for that product. Each of the **Communicator's** three technical editors is a member of the product support group for his/her system. You may forward your comments directly to the appropriate technical editor as follows:

HP 2000 Access and Timeshare Systems – Anna Holland
HP 3000 System – Pat Wilcox
9600/9700 Product Line (includes BCS, RTE) – Mike Manley

Anna and Pat are at the new General Systems Division, 5303 Stevens Creek Blvd., Santa Clara, Ca. 95050. Letters for Mike should be sent to Data Systems Division, 11000 Wolfe Road, Cupertino, Ca. 95014.

The Feature Article in this issue is "Three Modes of Data Communication." Several customers have expressed interest in tutorials on the subject of data communications; we're hopeful this and subsequent articles will meet some of your needs. The **Feature Article** in the next issue of the **Communicator** will discuss an application of data communications, distributed systems. Keep in mind that we also welcome applications articles submitted by our readers.

Address your correspondence to:

Editor
Computer Systems Communicator
HP General Systems Division
5303 Stevens Creek Blvd.
Santa Clara, Ca. 95050

OOPS!

The following program was accidentally left out of the article on page 175 of the November 15 **Communicator.** Apologies to author Mike and to any of you who may have been confused by the omission.

Editor

```
FTN4,L                  !
      PROGRAM TIME      !
      DIMENSION IDCB(144),NAME(3),IBUF(32),ISTIM(5),IETIM(5)
      DATA NAME/2HMY,2HF1,2HLE/
      CALL OPEN(IDCB,IERR,NAME,0,0,-2,128)
C GET STARTING TIME                    ↑
      CALL EXEC(11,ISTIM)              |
      DO 100 I = 1,6342
      CALL WRITF(IDCB,IERR,IBUF,1,I)
100   CONTINUE
C GET ENDING TIME
      CALL EXEC(11,IETIM)
      WRITE (1,200) (ISTIM(I),I = 1,5)
      WRITE(1,200) (IETIM(I),I = 1,5)
200   FORMAT(5I5)
      END
      END$
```

Only two numbers, indicated by arrows, were changed to generate execution time table.

contents

bulletins

2000 ACCESS SYSTEM SOFTWARE

If you own a 2000 Access System, it is important that you run current software. How do you know what is current? The best method of obtaining up-to-date software is by making sure your CE fills out and sends to the factory in your behalf a form titled SOFTWARE PRODUCT MAINTENANCE REQUEST (5952-4645). This form should be used by everyone whose system is still under warranty, who has a software Maintenance Agreement, or who uses the Software Subscription Service.

The purpose of completing the form is to let the factory know who to send software to when there are changes. It also is used to send you revised manuals.

The form should be filled out for you regardless of whether you have a new Access System or have upgraded from a previous system.

Bruce Templeton
HP General Systems

documentation

The following tables list all currently available software manuals for the HP 2000 Timeshare and 2000 Access systems. This list supersedes the previous list in the **Communicator**. The column labeled DATE specifies the date of the latest edition of the particular manual. The column labeled UPDATE specifies the date of any applicable update package. Copies of manuals and update packages can be obtained from your local Sales and Service Office. The address and telephone number of the office nearest to you are listed in the back of all reference manuals.

Customers in the U.S. may also order directly by mail. Simply list the name and part number of the manuals you need on the Corporate Parts Center order form supplied at the back of the **Communicator**. When using the Direct Mail order form, please remember to add to the list price any applicable local and state sales taxes, plus the additional $1.50 handling charge. Your check or money order must accompany each direct mail order. If you require an update package only, send your request to:

Software/Publications Distribution
11000 Wolfe Road
Cupertino, Ca. 95014

PART NUMBER	MANUAL TITLE	PRICE*	DATE	UPDATE
02000-90048	20856A Timeshared BASIC/2000, Level E, Reference Manual	$10.00	9/74	
02000-90049	20856A Timeshared BASIC/2000, Level E, System Operator's Manual	5.00	9/74	
02000-90055	2000C/2000F IDF Author's Manual	8.50	1/73	8/75
02000-90073	20854A Timeshared BASIC/2000, Level F, Reference Manual	7.50	10/74	8/75
02000-90074	20854A Timeshared BASIC/2000, Level F, System Operator's Manual	10.00	6/75	10/75
02000-90080	HP 2000E to HP 2000F Conversion Guide	1.00	12/73	
02116-9077	20392A HP BASIC Reference Manual	15.00	9/74	
19665-90001	2000/F to 2000/Access System Upgrade Kit and Conversion Program Manual	2.00	11/75	
20308-90001	Instructional Management Facility Proctor's Manual	7.00	9/74	
20308-90003	Instructional Management Facility System Manager's Reference Manual	5.00	10/74	
20309-90001	Instructional Dialogue Facility Proctor's Manual	10.00	9/74	
20309-90003	Instructional Dialogue Facility Course Developer's Manual	6.00	8/74	
20309-90005	Instructional Dialogue Facility Author's Pocket Guide	3.50	10/74	
20310-90001	HP MATH Teacher's Handbook	5.00	9/74	
20310-90005	HP MATH Proctor's Manual	5.00	9/74	
20310-90007	HP MATH Curriculum Guide	20.00	7/74	
20311-90001	Timeshared Graphics for Tektronix Terminals	7.00	8/74	
20311-90003	Timeshared Graphics Plotting Package	5.00	6/74	
20352-90001	Educational Budget and Accounting System — System Overview	5.00	6/74	
20352-90002	Educational Budget and Accounting System Reference Manual	10.00	3/75	10/75
20352-90003	Educational Budget and Accounting System — Technical Manual	75.00	3/75	
20353-90001	Educational Payroll System — System Overview	3.50	10/74	
22687-90001	HP 2000/Access BASIC Reference Manual	10.00	9/75	

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
22687-90005	HP 2000 Access Operator's Manual	10.00	9/75	10/75
22687-90009	Learning Timeshare BASIC	3.00	5/75	
22690-90001	Instructional Management Facility for HP 2000 Access Proctor's Manual	6.50	9/75	
22690-90002	Instructional Management Facility for HP 2000 Access System Manager's Reference Manual	4.50	9/75	
22691-90001	Instructional Dialogue Facility for HP 2000 Access Proctor's Manual	6.00	9/75	
22691-90002	Instructional Dialogue Facility for HP 2000 Access Course Developers' Manual	5.00	9/75	
22691-90003	Instructional Dialogue Facility for HP 2000 Access Author's Manual	13.00	9/75	
22691-90004	Instructional Dialogue Facility for HP 2000 Access Author's Pocket Guide	3.00	9/75	
22693-90001	HP MATH for HP 2000 Access Teacher's Handbook	5.50	9/75	
22693-90002	HP MATH for HP 2000 Access Proctor's Manual	6.50	9/75	
22693-90003	HP MATH for HP 2000 Access Curriculum Guide	17.50	9/75	
24383-90001	Course Writing Facility	15.00	5/74	
24384-90001	College Information System – System Overview	5.00	6/74	
24384-90003	College Information System Reference Manual	19.00	10/75	
24384-90005	College Information System Technical Manual	95.00	5/75	
24387-90001	Basic Analysis and Mapping Program Manual	18.00	6/74	5/75
24387-90002	Basic Analysis and Mapping Program Pocket Guide (10 copies)	10.00	6/74	
5951-1352	The Librarian	1.00	1/71	
5951-1353	Special Purpose Magnetic Tape Loader/CAI English	1.00	1/71	
5951-1381	DOS-M/2000C Timeshared BASIC File Handler	1.00	5/71	
5952-4490	20856A Timeshared BASIC/2000, Level E, Pocket Guide	0.15	10/74	
5952-4491	20854A Timeshared BASIC/2000, Level F, Pocket Guide	0.15	10/74	8/75

*Prices listed are subject to change without notice.

training schedule

The schedule for customer training courses on General Systems Division Products is outlined below and in the HP 3000 section of this publication. Included here are 2000 Access courses and special seminars for the 4 month period, January through April 1976.

GENERAL SYSTEMS DIVISION COURSE SCHEDULE

Jan. 1976 - Apr. 1976

Course Dates and Training Center Location

COURSE NUMBER	COURSE TITLE	LENGTH	PRICE	GENERAL SYSTEMS SANTA CLARA	EASTERN TRAINING CENTER – ROCKVILLE
22973A	2000 Access Data Entry, File Management and RJE	5 days	$500	2/2/76 3/22/76	2/23/76
22974A	Minicomputers in Manufacturing Seminar	2 days	$200	2/5/76 3/25/76	

Registration

Requests for enrollment in any of the above courses should be made through your local HP Sales Office. Your Sales Representative will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the training course, time of class, location and accommodations reserved.

Accommodations

Students provide their own transportation, meals, and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time your Sales Office requests a registration.

Cancellations

In the event you are unable to attend a class for which you are registered, please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than two weeks before the class begins.

Eastern Training Center

Hewlett-Packard
4 Choke Cherry Road
Rockville, Maryland 20850
(301) 948-6370

General Systems Division Training Center

Hewlett-Packard
5303 Stevens Creek Blvd.
Santa Clara, Calif. 95050
(408) 249-7020

FCOPYing Files to Magnetic Tape

When FCOPYing a file to magnetic tape, the tape device does not rewind until the next FCOPY command is entered. If the next command does not append to the current tape file, FCOPY writes an EOF on the tape and rewinds it. Do not manually rewind or dismount the tape before entering another FCOPY command. If you do, the tape will not contain a proper EOF, and your Job/Session will wait for the tape drive to become ready so that FCOPY can write the EOF. While your Session is waiting, the terminal is locked out. If someone else mounts a tape with a write ring on your tape unit, they may find to their dismay that FCOPY has written an EOF on their good tape.

To free the terminal, mount a scratch tape with a write ring on the tape unit owned by FCOPY. If you have already entered another FCOPY command, or attempted to abort your Session, FCOPY will write an EOF on the scratch tape and rewind it. Your terminal should become available for further use. To obtain a tape with a valid EOF, re-do the previous FCOPY function(s) and allow FCOPY to rewind the tape for you.

Sam Boot
HP General Systems

SEGMENTATION FOR MAXIMUM EFFICIENCY OF SYSTEM-TYPE PROGRAMS

The purpose of this article is to describe, for the benefit of system programmers, some guidelines for the optimum design of programs for the 3000; in particular, attention will be given to the questions of segmentation.

The 3000 is a process oriented machine, incorporating the separation of code and data, and stack architecture. This permits easy design of re-entrant code. The purpose here is to discuss ways of making a particular process

a. Run as fast as possible

b. Have minimum effect on other processes in the system.

Process Environment

When you write a program, it is executed by MPE in the form shown in Figure 1. The process has a single data segment (or "stack") and a variable number of code segments of varying sizes. When you write your program you can control:

a. the size of the stack

b. the number of your code segments

c. the size of each segment

d. which code goes into which segment.

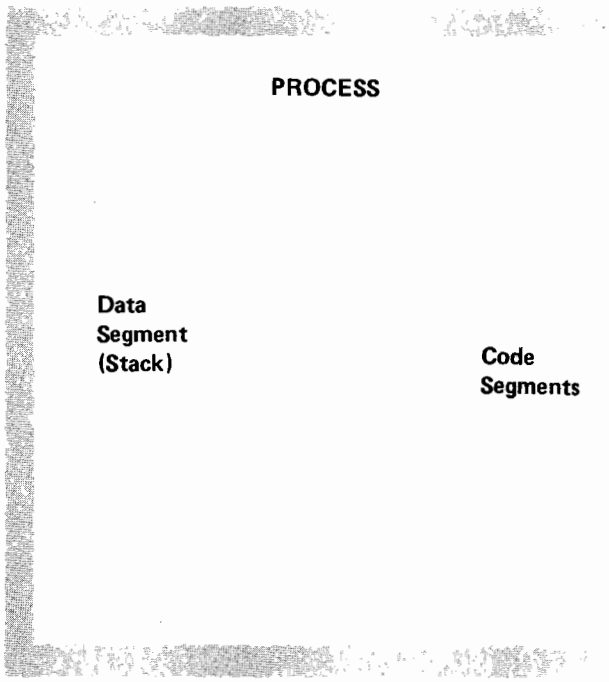


Figure 1.

The diagram shown above is actually a simplification since it does not show the externals referenced by your program (see Figure 2). If for example, your SPL-written program calls FOPEN, then a link will be created from your code to an MPE segment containing the FOPEN intrinsic code. Most of these intrinsics and all the Compiler Library routines are not in memory permanently, thus they are viewed by MPE as code segments identical to your own even though they were not written by you. For programs written in SPL, you are in control of which external procedures are called, since the calls are made explicitly. For other languages, the compiler will implicitly create in your program calls to external routines in order to perform, for example, a Fortran WRITE or a COBOL DISPLAY. The environment of a non-SPL program is harder to control because it requires a knowledge of when the compiler will emit those external calls. We will limit this discussion to those areas over which you have primary control: your own program code and data stack. Given any language, there are some fundamental principles to follow which will decrease the run-time of a process and its impact on system load. In a future issue of the **Communicator** we will include examples pertaining to COBOL and RPG.

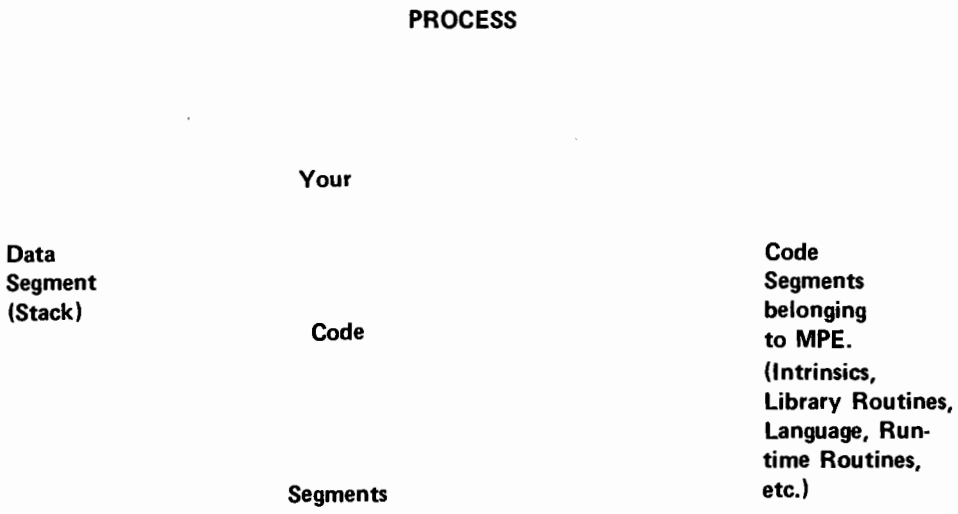


Figure 2.

How to Determine a Program Environment

When you prepare your program the PMAP option will show you the size of each segment, which procedures are in which segment, and the names of externals called by each segment. The MPE manual describes the format of the PMAP in detail.

How MPE Runs Your Program

There are two MPE modules concerned here — the dispatcher and the memory management system. The dispatcher is responsible for the allocation of CPU time to all the executing processes. The memory management system has the job of fitting code and data segments into memory as they are required, this operation often requiring the decision of which segment(s) to delete to make space. When your time-slice starts, the stack is made present in memory and control is passed to the program. As the program proceeds, it will call procedures which are not in the current segment. At this point your program is suspended while MPE arranges to make the required segment present. This can take from 20 to 100 milliseconds since a disc access is involved. While this is going on the dispatcher tries to run the process with the next highest priority which is already resident in memory. When the destination segment has been made present, control is passed to the procedure originally called.

The point to note here is that calling a procedure in an absent code segment is a time-consuming job.

How Do I Tell If A Segment Will Be Present?

You can't. The memory management system will simply attempt to keep the most popular segments in real memory. Since a typical code segment is about 4K decimal words in length and in a 64K system about 20K is used by MPE exclusively there is room for about (64-20)/4=11 segments. This assumes that all memory is occupied with code. If we assume a 12K stack then only 8 segments can be fitted in. Obviously, with these constraints and the fact that dozens of processes are competing for memory, it is important to consider how a program is segmented. The 3000 can deceive the unwary since calling a procedure is equally easy whether it is in the current segment or not. The writer must therefore be wary of calls referencing procedures outside the current segment (external calls) since they cause expenditures of time and considerable disruption to other processes while memory management re-organizes real memory.

Rules for Segmenting Your Program

Rule No. 1

Minimize the number of times the program crosses a segment boundary. In other words, stay within a segment for as long as possible. When you leave it, stay out for as long as possible.

Design of Programs is Important

Do not leave segmentation to the last minute. As will be shown below, it is possible to write programs that cannot be correctly segmented.

Any procedure or outer block Relocatable Binary Module (RBM) must reside wholly within a segment. Thus if it

proves necessary to move a block of code into a separate segment, it will only be possible if the code is a procedure. You cannot take an arbitrary set of instructions and place them into a named segment – the whole RBM must be moved. Therefore, the way you divide your program into procedures is vitally important in the design phase.

Concept of Locality

The locality of a program is the degree to which control remains in the same general area of code. A high locality means that control remains in the same area for a long period of time. Poor locality means the program branches wildly and rapidly all over the place. The 3000 needs programs that have good segment locality but does not care about the degree of locality within any given segment. That is to say, it does not want programs that jump from segment to segment continuously but once inside any given segment, it doesn't matter what the locality is like.

Functional vs. Temporal Segmentation

Intuitively, one segments according to the function of the procedures. That is, all the command decoding routines are put together, the command executors are put together, etc. This is wrong. Wrong. Segmentation is a speed-enhancing operation thus time, not function, is the critical dimension. Since Rule No. 1 says stay inside a segment for as long as you can, control must pass smoothly from segment to segment as the program progresses.

As an example, consider a small utility program which dumps a file to the line printer in some special format. Let us suppose that the operator can choose the name of the file and which of three possible formats to use. The program is written with four procedures: A, B, C, and D.

Let us further suppose that each dump routine has a procedure to fetch a record from its file and a procedure to format a print line:

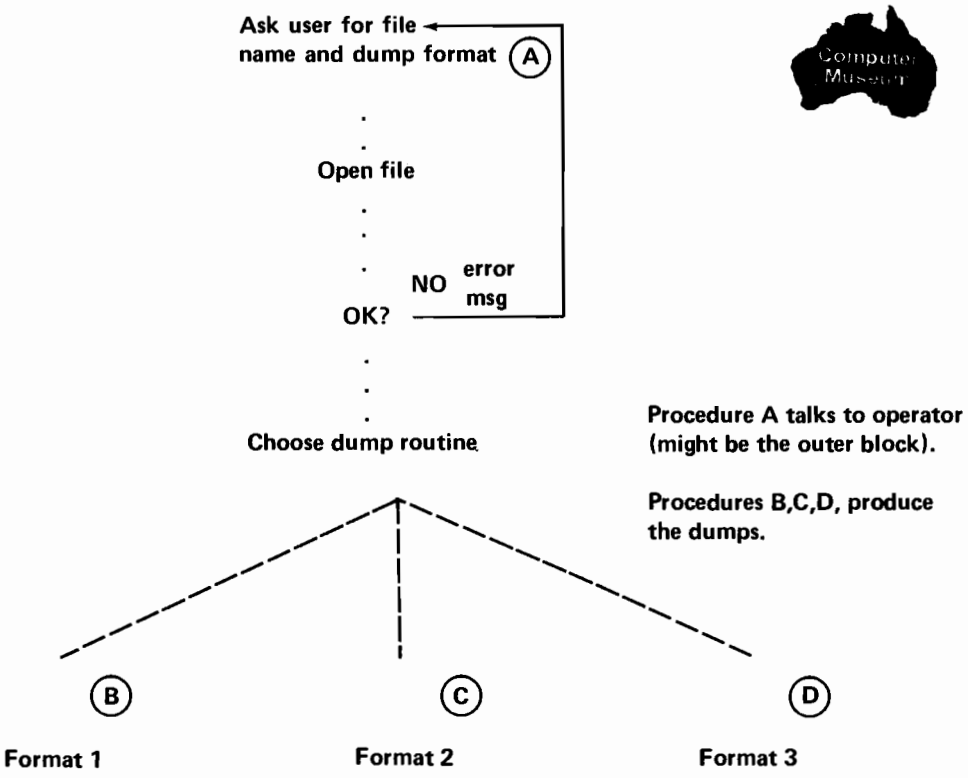


Figure 3.

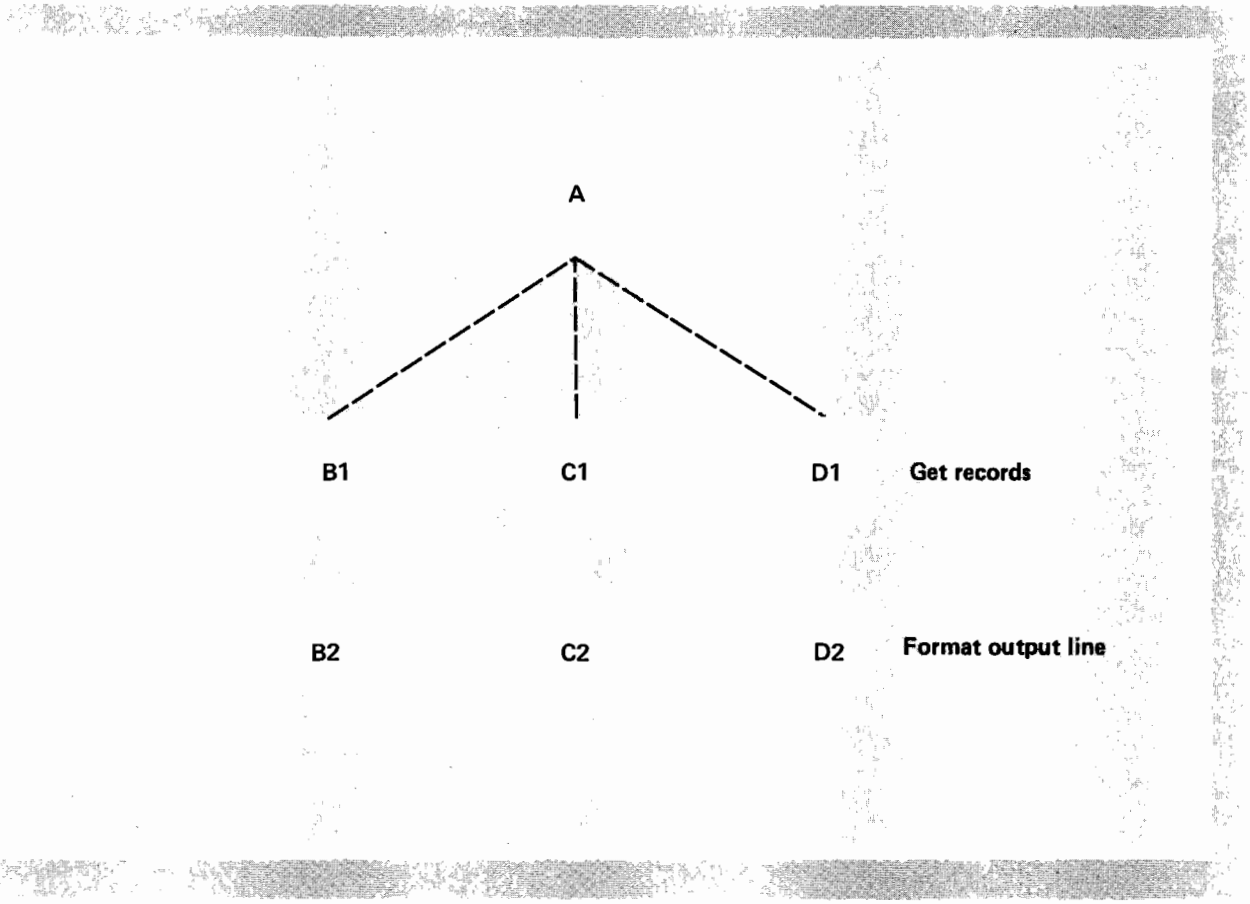


Figure 4.

It would be tempting to put all the formatting routines in one segment, and the record fetching routines in another. This would cause a segment boundary to be crossed twice for every record dumped – perhaps a thousand times. The correct way is to put B1B2 together, C1C2, etc. If A is in its own segment then only three segment boundaries are crossed for a whole dump. In a busy system this simple change could make large differences in the run time of your program.

To sum up, estimate the number of times a segment boundary is crossed in your program and multiply this by 40 milliseconds (12 msec if you have a swapping disc and your program resides on it). This is the time your program will be doing no useful work and other processes will be disrupted.

Rule No. 2

Do Not Burden Your Working Set With Infrequently Used Code

Let us suppose that you have arrived at some segmentation scheme using the above rule so that you have good segment locality. The next step is to reduce the size of the 'working set'.

Frequency of Code Use

The 'working set' of segments is the set that consumes most of the CPU time. For example in the program above the working set is the code that executes the main loop such as C1C2. Let us assume that C1C2 are in a segment of their own called CSEG. The system may spend minutes in this segment for a large dump. It is important therefore to minimize its size in order to reduce contention for the scarce memory resource.

To do this, examine the codes in the working set and remove any code that exectues infrequently. Very often, this applies to code which does error-handling. When your program detects an error, do not handle it in-line. Write an error-message generating procedure and call it with a parameter indicating which message to output. This can be put in a separate segment and thus not clutter up memory while doing normal error-free processing. As another example, suppose that in the program mentioned above, after doing an FWRITE, you check the condition code for end-of-file and, if required, execute a somewhat elaborate sequence to extend the file by building a new one and copying the old into it and then purging the old file. If this condition is likely to occur once in every 500 runs, why hold it in precious memory with the working set? Banish it to some auxiliary segment and let MPE bring it in only when needed. Remember that you can only move this code if it is a procedure.

WRONG

```
FWRITE(. . .);
IF>THEN
BEGIN
.
.
<<LENGTHEN FILE>>
.
.
END;
```

RIGHT

```
FWRITE(. . .);
IF>THEN EXTEND'
FILE;
```

Procedure EXTEND'FILE is put in another segment

Segment Sizes

This is a trade-off. If you have lots of small segments then a scarce resource is being used up in the form of Code Segment Table (CST) entries. One entry in the CST is needed for every loaded segment, and the table has a maximum of 256 entries of which about 80 are needed by the system itself. A program is getting large when it gets to around ten segments. A typical compiler for example has about 25, while a simple utility like SORT has 3.

Another problem is that if a working set has many segments then many disc accesses must be done to get them into memory soon after the time-slice starts. Not only does this take 40 msec per segment, but this is charged against your time slice.

At the opposite end of the spectrum, your program might have a few large segments. While this does minimize segment-boundary crossings, the effect on memory is devastating for other users. The optimum size is about 3K decimal words.

Rule No. 3

Make segment as small as possible with a maximum of about 3K decimal words.

Rule No. 4

If Rule 3 has to be violated in order to reduce the number of segments, keep principal working sets small and make infrequently used segments large.

If Your Code is Shared

If your program is going to be run from multiple terminals then the code segments will automatically be shared by the multiple processes. Each process will have its own stack of course. If your program design requires data which is never altered such as error messages, look-up tables, etc., then by placing them in the code rather than the stack, only one copy is required for all processes.

WRONG

```
BEGIN
BYTE ARRAY MESSG (0:22):="TOO
MANY TIMES ENTERED";          Global Declarations
.
.
.
.
PROCEDURE MESSOUT;            Procedure to print error
BEGIN                         message
PRINT (MESSG,-23,0);
:
END;
    :
END.
```

WHY WRONG? The array MESSG is present in the stack perpetually. Each process running this program carries the message string around in its stack.

RIGHT

```
BEGIN                 MESG only exists while MESSOUT
.                     executes. SPL will store the string
.                     in quotes in the code segment —
.                     effectively making it shared. The
.                     stack is now smaller.
.
.
PROCEDURE MESSOUT;
BEGIN
BYTE ARRAY MESG(0:22);
MOVE MESG:="TOO MANY VALUES ENTERED";
PRINT (MESG,-23,0);
END;
.
.
.
END.
```

Rule No. 5

In SPL, keep initialized variables, especially arrays, out of the GLOBAL DECLARATIONS.

In Fortran, infrequently used variables and arrays should not be initialized in DATA statements.

John Page/Madeline Lombaerde
HP General Systems

"FORMMSG" FROM FORTRAN

To use special forms from Fortran you must open the file yourself as FORMMSG is a parameter in FOPEN. Additionally you must call FSET from the Compiler library to translate the MPE file number returned from FOPEN into a

Fortran unit number. FOPEN must be declared as an integer. The forms message has a maximum length of 49 bytes including the period required as a terminator. The file name is not required to be FTNnn.

The sample program, console log and partial line printer output will serve as further illustrations. For additional information reference the Fortran manual Appendix A and the FOPEN intrinsic in the MPE manual.

Console Log

```
IO/15:56/28/FORMS:                          message at time
MOUNT PICK LIST FORMS <-------------device is FOPENed

?IO/15:56/28/SP#6/ IS #S126;          message generated by
PICKLIST ON LDEV#6 (Y/N) <---------- a system procedure

=REPLY 28,Y<-------------if "Y" alignment line is printed

?IO/15:57/28/LDEV#6                   message generated by
FORMS ALIGNED OK?(Y/N)<-------------a system procedure

=REPLY 28,Y <------------------- if "Y" output begins
```

Sample Program

```
   CHARACTER FORMMSG*30,PICKLIST*9,DEV*9
   INTEGER FOPEN
   FORMMSG="MOUNT PICK LIST FORMS."
   PICKLIST="PICKLIST"
   DEV="6"
   INUM=FOPEN(PICKLIST,\4\,\%301\,\0\,DEV,FORMMSG,
  1\0\,\0\,\0\,\0,0\,\0\,\0\,\0\,\%16600\)
   IF (.CC.)10,20,10
   DISPLAY "FILE ERROR IN OPENING LINE PRINTER"
   STOP
   CALL FSET(10,INUM,IOLD)
   DO 30 I=1,50
   WRITE(10,*) "TEST OF FORMS",I
   CONTINUE
   STOP
   END
```

Partial Line Printer Output

```
....!....1....!....2....!....3....!....4..
```

Operator should depress "TOP OF FORM"

```
TEST OF FORMS      1
TEST OF FORMS      2
TEST OF FORMS      3
TEST OF FORMS      4
TEST OF FORMS      5
TEST OF FORMS      6
TEST OF FORMS      7
TEST OF FORMS      8
TEST OF FORMS      9
TEST OF FORMS     10
TEST OF FORMS     11
TEST OF FORMS     12
```

Dick Sleght
HP General Systems

bulletins

STUDENT ASSIGNMENT SYSTEM RELEASE

The Student Assignment System (SAS), HP product number 32901A, is now available for HP 3000 customers. It is an administrative application software package for elementary, secondary and vocational schools that assigns students to classes. The system compares a previously developed master schedule with student course requests to produce class schedules.

It is written in COBOL and uses IMAGE for data base management. Both batch and terminal access are permitted. The package assists in the building of the master schedule, assigns students to classes, provides a set of comprehensive reports and maintains school and student files. The software can provide information for use by the Student Information System (SIS) (HP product number 32900A) mark reporting and attendance subsystems. Detailed information can be obtained from your field engineer.

Jean Danver
HP General Systems

REFERENCE MANUAL

The HP Student Assignment System (SAS) Reference Manual is now available and may be ordered using the Corporate Parts Center direct mail order form in the back of the **Communicator**. The HP part number is 32901-90001 and the price is $10.00.

Although SAS may be used as a stand-alone system, it is compatible with the HP Student Information System (SIS). SIS is used for recording, revising and reporting vital school related information such as student and family profiles, student academic statistics, emergency contacts, and bus route information. Using an SIS program, selected records from the SIS data base can be loaded into the SAS data base for scheduling; then, in turn, an SAS program will load the records back into the SIS data base for mark reporting.

The SIS manual HP part number is 32900-90001 and the price is $18.00.

Barbara Lewis
HP General Systems

CONTRIBUTED PACKAGES

The 3000 Contributed Library has three new Contributed Packages: ISAM, RPG COMPOSER, and IDEA/3000.

The ISAM package was designed for use by the HP 3000 RPG user. The ISAM package consists of a set of library procedures to access ISAM files. These routines are automatically invoked by the RPG program. The conversion from other systems should, in most cases, only require recompilation of the user program. These routines are also

callable from SPL, Fortran, and COBOL. Also included in the ISAM package is a utility program RSAMUTIL, for building and maintaining ISAM files.

The second contributed program is the RPG COMPOSER. This package runs only with the HP 2640 and 2644 terminals. The package provides a header on the screen for each type of RPG coding form. This header is an aid to the user in his development, entry and editing of his RPG program.

The IMAGE Data-Base Evaluative Analyzer (IDEA) was designed to be a measurement tool for rapid analysis of the system performance of a proposed IMAGE data base(s). This easily used package requires that the user provide the desired data relationships in the form of an empty IMAGE data base(s) and the user's transactions in the form of "Script Files". This will provide the following benefits:

a. Provide the data base designer quick assessment of the performance of his proposed data base in terms of response time and through-put;

b. Provide the System Manager with the capability to build a model(s) of his operational data base systems, so that proposed changes to the current operational performance can be predicted without jeopardizing the "Status Quo".

These packages will be catalogued in the second volume of the HP 3000 Contributed Library, Index and Catalog. Until Volume 2 is completed and orderable, ISAM and the COMPOSER can be obtained by ordering part number 36999-10902, priced at $20.00. This includes a magnetic tape containing the software for ***both*** packages and the appropriate user's guides. The IDEA package can be obtained by ordering part number 36999-10901; its price is $15.00. This includes a magnetic tape containing the software and a User's Guide. These part numbers are temporary and will be obsoleted when Volume 2 of the 3000 Contributed Library, Index and Catalog is available.

Bob Johnson
HP General Systems

3000 COBOL SELF STUDY COURSE

The second printing of the 3000 COBOL Self Study Course is now available from General Systems Division. It utilizes modularized audio cassettes that complement a student workbook and reference manual. The topics covered are: 1) Overview of COBOL, 2) Language Elements, 3) Identification, Environment and Data divisions, 4) Procedure division, 5) COBOL SORT verb, 6) Interprogram communications, 7) Library features, 8) Table handling, 9) Random access, 10) Segmentation. The course is designed for beginning and intermediate programmers who are already familiar with fundamental data processing concepts and have an understanding of program logic. Access to an HP 3000 is assumed.

The basic COBOL package (22957-90000) has ten audio cassette tapes, manuals, card decks, related listings and coding forms. The basic package is $325.00. The Advisor's Guide (22951-60005) contains the solutions to the lab exercises, answers to review questions and a pre/post test with answer key. The Advisor's Guide is $25.00. The course is designed for the student to work at his own pace. Approximately 30–40 hours should be allotted.

Orly Larson
HP General Systems

CONTRIBUTED LIBRARY

The following list gives the currently available order numbers and prices for the HP 3000 Contributed Library.

1. HP 3000 Contributed Library 5952-5564(22) N/C
   A Guide for Contributors
   Orders for the complimentary Guide for Contributors should be directed to:

   Hewlett-Packard Co.
   Inquiry Department
   1820 Embarcadero Rd.
   Palo Alto, Calif. 94303

2. HP 3000 Contributed Software 36995-90001 $ 7.50*
   Index and Catalog, Vol. 1

3. HP 3000 Contributed Library
   Magnetic Tape, Vol. 1
   (800 BPI) 36995-10001 $50.00*
   (1600 BPI) 36995-11001 $50.00*

*In using the Direct Mail order form, please remember to add to the list price any applicable local and state sales taxes, plus the additional $1.50 handling charge. Your check or money order must accompany each direct mail order. A direct mail order form appears at the end of this publication.

Customers outside the domestic U.S.A. should place their order through their local sales office.

Brenda Mapp
HP General Systems

software updates

Each issue of the **Communicator** provides you with information pertinent to the status of 3000 software products including the latest software changes and enhancements.

Software updates described in this issue relate to the following products:

MPE 32000C.00.10
HP 32216A Query/3000
HP 32104 RPG/3000

Where changes in documentation are indicated, updates to the appropriate manuals will be printed. This information is provided simply as a temporary measure.

Products described are available through your Customer Engineer, or can be ordered directly via Customer Parts Center in Mountain View, California.

MPE 32000.00.10

This article describes MPE 32000.00.10 as incorporated in the MIT date coded 1549.

Table of Contents

1. Module Changes C0.0X

MODULE		MPE FIX LEVEL 1	2	3	4	5	6	7	8	9	10
INITIAL	0	X		X	X			X	X	X	X
SYSDUMP	1	X	X	X			X		X	X	
SEGPROC	2	X	X				X			X	X
SEGDVR	3										
DISPATCH	4			X			X				X
LOAD	5		X								
MAPP	6					X					
UCOP	7	X									
DEVREC	8										
PROGEN	9	X							X	X	X
ININ	10					X		X			X
EXIN	11	X	X	X		X	X	X		X	
LOG	12	X									
IOPTRD0	13										
IOPTPN0	14						X		X		
IOPLOT0	15										
IOMDISK0	16			X				X	X	X	
IOFDISK0	17			X				X	X		
IOTAPE0	18				X				X		
IOLPRT0	19										
IOCDRD0	20		X				X				
IOCLTTY0	21										
IOTERM0	22										
IOCDPN0	23										
IOPRPN0	24					N	X				
IOREM0	25										
IOBSC0	26										
IOMDISK1	27	X		N				X	X	X	X
PFAIL	30			X	X	X					
FILESYS	50	X	X	X	X	X	X	X	X	X	X
COMM'INT	51	X		X			X				
STORE/RESTORE	52			X		X			X	X	X
DIRC	53									X	
ALLOCATE	54		X		X				X		
DISKSPC	55	X									
MMCORER	56						X		X		
MMDISKR	57										
ABORTRAP	58						X	X		X	
MESSAGE	59							X		X	X
CROUTINE	60			X	X						X
IOUTILITY	61	X		X	X			X	X	X	
TTYINT	62		X	X	X			X		X	
PCREATE	63	X									X
MORGUE	64			X						X	X
PROCMAIL	65										
PINT	66						X		X	X	
DATASEG	67	X								X	
IOPM	68		X			X					
CHECKER	69										
UTILITY	70	X	X	X		X					X
SEGUTIL	71	X		X				X		X	
LOADER1	72		X	X					X		X
RINS	73						X				
JOBTABLE	74	X									
DEBUG	75	X									
NURSERY	76			X							
SYSDPLY	77						X				
FIRMWARESIM	78	X								X	X
SPOOLING	79			X	X			X	X	X	X
SPOOLCOMS	80	X					X		X		X
MESSAGE CAT				X			X	X	X	X	

2. Problems Solved in MPE C0.10

   a. The resource contention problem in the dispatching algorithm has been corrected. The dispatcher would not attempt to adjust to situations where a resource (memory or internal resource) was owned on behalf of a process of low priority but required by a process of higher priority. The problem would manifest itself by the presence of an unproductive loop or the dispatcher pausing because of an incorrect assumption that it had nothing to do. The dispatcher will now adjust to this situation by only scheduling (allocating the CPU) processes capable of running (already having the necessary resources). This allows the freeing of resources necessary for the successful scheduling of higher priority processes.

b. A process synchronization problem has been fixed. If a son process terminated during the time the father process was loading another son process, the father process was mistakenly activated.

c. User label information in the file label, will now also be updated on an FCONTROL to rewind (control=5) disc files.

d. A process synchronization problem involving freeing RINs has been corrected.

e. Buffer synchronization was wrong when backspacing was performed immediately following spacing past a tape file mark.

f. The 7905 driver has been modified to correctly handle recoverable disc errors.

g. A correction has been made to allow the rescheduling of a JOB having been "waited" due to a spoolee's queue being SHUT because of insufficient spool disc space. Previously, a "waited" JOB would remain in this "waiting" state until the introduction of a new JOB into the JOB scheduling queue.

3. MPE Enhancements

a. A new command has been added to the set of console commands to recall pending messages that require a reply.

=RECALL

This command displays all console REPLY messages pending and the text of the original request(s) will appear on the console.

REPLY(S) PENDING:

?10/12:03/#S8/26/LDEV # FOR "ED" ON TAPE (NUM)

No bell will ring as is the case for the original message. If there are no pending messages, the following appears:

NO REPLIES PENDING

4. Known Problems in MPE

a. Closing a tape file with NO REWIND is not implemented.

b. FSPACE spaces tape files by blocks rather than by records.

c. Chained SIOs on magnetic tape do not perform correctly, causing transfer of blocks larger than 4096 words to fail if the record format is variable or undefined.

d. The character ":" is treated as an EOF on $STDINX.

e. The commands: LISTACCT, LISTGROUP, and LISTUSER can lock the directory indefinitely if the output is written to magnetic tape and the tape is not ready.

f. Input arguments to the intrinsic BINARY of 65536, 65537, 65538, and 65539 fail to return overflow.

g. If the FORMSG parameter of FOPEN begins on an odd byte boundary, the preceding byte is also printed.

h. Lower case :eod is not recognized as an end-of-file on data accepting devices.

i. Issuing a :DEALLOCATE command for a nonexistent program file returns an ERR 217. The error should be ERR 217,52. The 52 is the file system error number returned from FCHECK.

j. DEBUG break points cannot be set in dynamically loaded procedures except by specifying the physical CST numbers.

k. When DPAN finds that the PCB table has been filled, it prints the erroneous messages "INVALID FIRST UNASSIGNED ENTRY" and "INVALID BACKWARD SUBQUEUE POINTER" even though there is no error in the PCB.

l. When the maximum number of open spoofles is not sufficient to handle all spooling requirements, spooled JOBs may cause endless numbers of null list files to be generated. This bug manifests itself as multiple $STDLIST files for a single JOB, each producing only a header and trailer. If the line printer is spooled, this results in many null spoofles, each using four sectors of disc space. If the line printer is not spooled, these null spoofles will begin printing immediately unless the printer is not ready. In this case, the system will crash due to an IOQ overflow. If an open spoofle is closed during this resource allocation loop, the job may be launched normally. In this case, the last spoofle for $STDLIST will be the true job listing.

This bug can be overcome by increasing the maximum number of open spoofles. The recommended value is 20, but a more exact figure can be found by examining the usage of your system. Each initial allocation (FOPEN) of a spooled device uses one open spoofle. When the file is closed (FCLOSE), the spoofle becomes unopened.

For example:

A SESSIONS's single access to a spooled line printer requires one opened spoofle; a spooled

JOB requires at least two, one for $STDIN and one for $STDLIST. Each additional access to a file of device class LP requires an additional open spoofle.

One indication that the limit is being reached is allocation failures for spooled devices.

5. Documentation Change

page 7-1

After the description of the console operator messages, insert:

=RECALL This command displays the REPLY messages pending:

REPLY(s) PENDING
IO/12:03/#S8/26/LDEV # for "ED" on tape (num)

If there are no pending messages, the following will appear:

NO REPLIES PENDING

HP 32216A QUERY/3000

This article describes HP 32216A.02.01 (QUERY/3000) as incorporated in the MIT date coded 1549.

The following error conditions have been corrected:

1. Stack overflow during a serial read of a data set.

   Changes have been made to the 'J00J216A' and 'I00I216A' Files.

HP 32104 RPG/3000

This article describes the release of HP 32104.02 as included in the MIT date coded 1549.

Problems corrected:

1. A field used in a chain operation and defined on a subsequent line was assigned an incorrect run time address.

2. Z-ADD of a constant of value (-1 < = constant < = +1) to an array only initialized the first element.

3. Record identification of beyond column 512 did not work properly all the time.

The following features have been added to RPG/3000:

1. Variable length output record support is now available.

2. The SETLL command has been added and can be used for IMAGE files or for user written ISAM files (see 3).

   The format for the set lower limit (SETLL) command is:

   The conditioning indicators and control level are optional.

   Column 6 must contain a C (calculation record).

   Factor 1 must contain a field name or literal with the lower limit being set. This field or literal must have the same length as the length of the key field specified on the file record (file name in factor 2).

   Factor 2 contains the name of the file the lower limit is being set for. The file must be an indexed or IMAGE demand file.

   Result field and resulting indicators are not used for this operation.

3. RPG support of user written ISAM.

   RPG will support a user written ISAM by accepting standard ISAM file definitions in RPG and calling a procedure (R'ISAM).

   Editor's note: A user written ISAM package is announced in the 3000 Bulletin section of this issue.

   The file definitions supported are the same as for IMAGE files (See Appendix H of RPG manual) although the specialized IMAGE continuation records are not used.

The RPG compiler recognizes an IMAGE file by the presence of IMAGE continuation records (which are required for IMAGE). If there are no IMAGE continuation records then RPG assumes a user ISAM procedure exists.

A user who wishes to code his own ISAM procedure must name the procedure R'ISAM (The parameters for this are described on the following pages).

The RPG Library will have a procedure of this name which, if called will print out an error message (USER ISAM PROCEDURE NOT SUPPLIED).

The user ISAM procedure may be incorporated with the generated RPG program in one of the following ways:

- Put it in a group or Public Library and use LIB=G or LIB=P parameter when running the generated program.
- Put it in an RL accessed at prepare time.
- Put it in the same USL file as the RPG generated program
- Replace the segment RPG'ISAM in the system library.

R'ISAM Procedure definition:

INTEGER PROCEDURE R'ISAM (***FNUM,TYPE, FARRAY***);

VALUE ***TYPE;***
INTEGER ***TYPE, FNUM;***
INTEGER ARRAY ***FARRAY;***
OPTION EXTERNAL;

The return values from R'ISAM are as follows:

0= Normal processing, no error occurred
+1= End of file encountered
+2= Record not found
+3= Duplicate Record
+4= No ISAM file processor present
<0= Fatal error occurred (RPG will print the error number and give an error dump.)

FNUM is the file number and must be returned by open

TYPE HAS THE FOLLOWING VALUES:

0 = Open an R'ISAM file
1 = Close an R'ISAM file
2 = Read the next record in key sequence
3 = Read the next physical record
4 = Read by relative record number
5 = Read by record key
6 = Write a record to the file, key is contained in record
7 = Not used
8 = Write a record to the file, key is passed as parameter
9 = Update previously read record, key field cannot be modified
10 = Seek relative record number
11 = Seek record by key
12 = Get current record number
13 = Get value of current key
14 = Delete current record

The contents of ***FARRAY*** for each type are as follows:

0 = Open
Word Contents (***FARRAY***)
0 Byte pointer to formal designator (RPG file name)
1 Foptions
2 Aoptions
3 Record size (negative=Byte size)
4 Byte Pointer to device name (as specified in RPG file card)
5 Formsmessage pointer (not used)
6 Number of user labels
7 Blocking factor
8 Number of buffers
9-10 File size
11 Number of extents maximum
12 Initial number of extents
13 File code
14 Key size (Byte length)
15 Key type (0=Byte key, 1=Packed decimal key)

Note: Words 0-13 correspond to the FOPEN parameters, words 14-15 are special R'ISAM parameters.

1 = Close
Word contents (***FARRAY***)
0 File number
1 Disposition (always zero)
2 Security code (always zero)

2 = Read serial (sorted order)
Word Contents (***FARRAY***)
0 File number
1 Target address (word)
2 Record length (Bytes, negative, maximum)

3 = Read consecutive (physical order)
Word contents (***FARRAY***)
0 File number
1 Target address (word)
2 Record length (Bytes, negative, maximum)

4 = Read relative record number
Word contents (***FARRAY***)
0 File number
1 Target address (Word)
2 Record length (Bytes, negative, maximum)
3-4 Record number

5 = Read by key
Word contents (***FARRAY***)
0 File number
1 Target address (word)
2 Record length (Bytes, negative, maximum)
3 Key pointer (Byte pointer)
4 Generic key length (should be zero or same as in open)

6 = Write record (Key in record)
Word contents (***FARRAY***)
0 File number
1 Target address (word)
2 Record length (Byte, negative)
3 Control (always zero)

7 = Not used

8 = Write record by key
Word contents (***FARRAY***)
0 File number
1 Target address (Word)
2 Record length (Byte, negative)
3 Pointer to key (Byte)

9 = Update previously read record
Word contents (***FARRAY***)
0 File number
1 Target address (word)
2 Record length (Byte, negative)

10 = Seek Relative record number
Word contents (*FARRAY*)
0 File number
1-2 Record number

11 = Seek Key
Word contents (*FARRAY*)
0 File number
1 Key pointer (Byte)
2 Key length (zero, not currently used)

12 = Get current record number
Word contents (*FARRAY*)
0 File number
1-2 Returned record number (used by RPG to save the current location of a sequential update file that a record is being added to)

13 = Get current key
Word contents (*FARRAY*)
0 File number
1 Hold area Byte pointer, the key of the current record should be moved to this location

14 = Delete current record
Word contents (*FARRAY*)
0 File number

documentation

The following tables list all currently available HP 3000 software manuals. The column labeled DATE specifies the date of the latest edition of the particular manual. The column labeled UPDATE specifies the date(s) of the applicable update package(s). This list supersedes the previous list in the **Communicator.** Copies of manuals and update packages can be obtained from your local Sales and Service Office. The address and telephone number of the office nearest to you are listed in the back of all reference manuals.

Customers in the U.S. may also order directly by mail. Simply list the name and part number of the manuals you need on the Corporate Parts Center form supplied at the back of the **Communicator** when using the Direct Mail order form, please remember to add to the list price any applicable local and state sales taxes, plus the additional $1.50 handling charge. Your check or money order must accompany each direct mail order. If you require an update package (the items under UPDATE in the tables) send your request to:

Software/Publications Distribution
11000 Wolfe Road
Cupertino, Ca. 95014

MPE/3000 MANUALS

PART NUMBER	MANUAL TITLE	PRICE*	DATE	UPDATE
03000-90096	Multiprogramming Executive General Information Manual	$ 4.00	11/73	
03000-90126	HP 3000 Software Pocket Guide	3.50	7/75	
32000-90002	32000C MPE/3000 Reference Manual	19.50	1/75	
32000-90004	32000C MPE/3000 Console Operator's Guide	7.00	1/75	
32000-90006	32000C MPE/3000 System Manager/System Supervisor Manual	13.00	1/75	

LANGUAGE MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
03000-90002	SPL/3000 Reference Manual	$ 7.50	11/73	
03000-90003	SPL/3000 Textbook	13.00	11/73	3/75
03000-90008	BASIC/3000 Interpreter Reference Manual	10.00	7/75	
03000-90025	BASIC for Beginners	5.50	11/72	
03000-90047	Cross Assembler for 2100 Computers Reference and Application Manual	17.00	3/75	
03000-90050	BASIC/3000 Interpreter Pocket Guide	2.50	9/74	
32102-90001	FORTRAN/3000 Reference Manual	13.50	6/75	
32103-90001	BASIC/3000 Compiler Reference Manual	3.50	11/74	
32104-90001	RPG/3000 Compiler Reference and Application Manual	22.00	2/75	
32104-90003	RPG Listing Analyzer	0.50	4/75	
32213-90001	COBOL/3000 Reference Manual	12.50	11/75	

ADDITIONAL MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
03000-90009	HP 3000 Compiler Library Reference Manual	$10.00	10/75	
03000-90010	HP 3000 Scientific Library Reference Manual	7.00	7/75	
03000-90011	STAR/3000 (Statistical Analysis Routines) Reference Manual	5.50	11/72	
03000-90012	EDIT/3000 Reference Manual	7.50	8/75	
03000-90015	HP 3000 Symbol Trace Reference Manual	4.00	2/74	
03000-90019	HP 3000 Computer Systems Reference Manual	14.00	9/73	
03000-90064	FCOPY/3000 Reference Manual	6.00	7/75	
03000-90107	HP 3000 Cross Loader for HP 2100 Computers	11.00	10/74	
03000-90121	A Guide for the Terminal User	7.50	6/75	
30130-90001	2780/3780 Emulator Subsystem Reference and Application Manual	10.00	12/74	
30300-90002	Programmable Controller Reference and Application Manual	12.00	2/75	
32215-90001	IMAGE/3000 Reference Manual	7.00	4/75	
32216-90001	QUERY/3000 Reference Manual	7.00	8/75	
32900-90001	Student Information System Reference Manual	18.00	3/75	
32900-90002	Student Information System – System Overview	7.00	9/74	
32900-90005	Student Information System – Technical Manual	18.50	9/74	
36995-90013	IBM 1130/1800 to HP 3000 FORTRAN Conversion Guide	6.00	2/75	5/75
32214-90001	Sort/3000 Reference Manual	6.50	4/75	
30301-90002	Real-Time Programmable Controller Reference Manual	9.50	2/75	
32901-90001	Student Assignment System Reference Manual	10.00	7/75	

*Prices listed are subject to change without notice.

training schedule

The schedule for customer training courses on General Systems Division products is outlined below and in the 2000 Access section of this publication. Included here are HP 3000 software courses and special seminars for the 4 month period January through April 1976.

GENERAL SYSTEMS DIVISION COURSE SCHEDULE

Jan. – Apr. 1976

Course Dates and Training Center Location

COURSE NUMBER	COURSE TITLE	LENGTH	PRICE	GENERAL SYSTEMS SANTA CLARA	EASTERN TRAINING CENTER – ROCKVILLE
22962A	3000 Commercial/Business User	5 days	$500	1/5/76	1/19/75
					2/2/76
				2/23/76	
					3/22/76
					4/5/76
				4/5/76	
				4/19/76	
22963A	3000 Scientific/Engineering User	5 days	500		1/5/76
				1/26/76	
					3/8/76
				3/15/76	
					4/19/76

Course Dates and Training Center Location

COURSE NUMBER	COURSE TITLE	LENGTH	PRICE	GENERAL SYSTEMS SANTA CLARA	EASTERN TRAINING CENTER – ROCKVILLE
22964A	3000 System Management	3 days	300	1/12/76 2/2/76 3/1/76 3/22/76 4/12/76	1/12/76 1/26/76 2/9/76 3/15/76 3/29/76 4/12/76 4/26/76
22956A	3000 Image	5 days	500	1/19/76 3/8/76	2/23/76
22974A	Minicomputers in Manufacturing Seminar	2 days	$200	2/5/76 3/25/76	
22975A	System 3 Conversion Seminar	2 days	200	1/15/76 3/4/76 4/29/76	

Registration

Requests for enrollment in any of the above courses should be made through your local HP Sales Office. Your Sales Representative will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the training course, time of class, location and accommodations reserved.

Accommodations

Students provide their own transportation, meals, and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time your Sales Office requests a registration.

Cancellations

In the event you are unable to attend a class for which you are registrered, please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than two weeks before the class begins.

Eastern Training Center

Hewlett-Packard
4 Choke Cherry Road
Rockville, Maryland 20850
(301) 948-6370

General Systems Division Training Center

Hewlett-Packard
5303 Stevens Creek Blvd.
Santa Clara, Calif. 95050
(408) 249-7020

featuring -

three modes of data communications

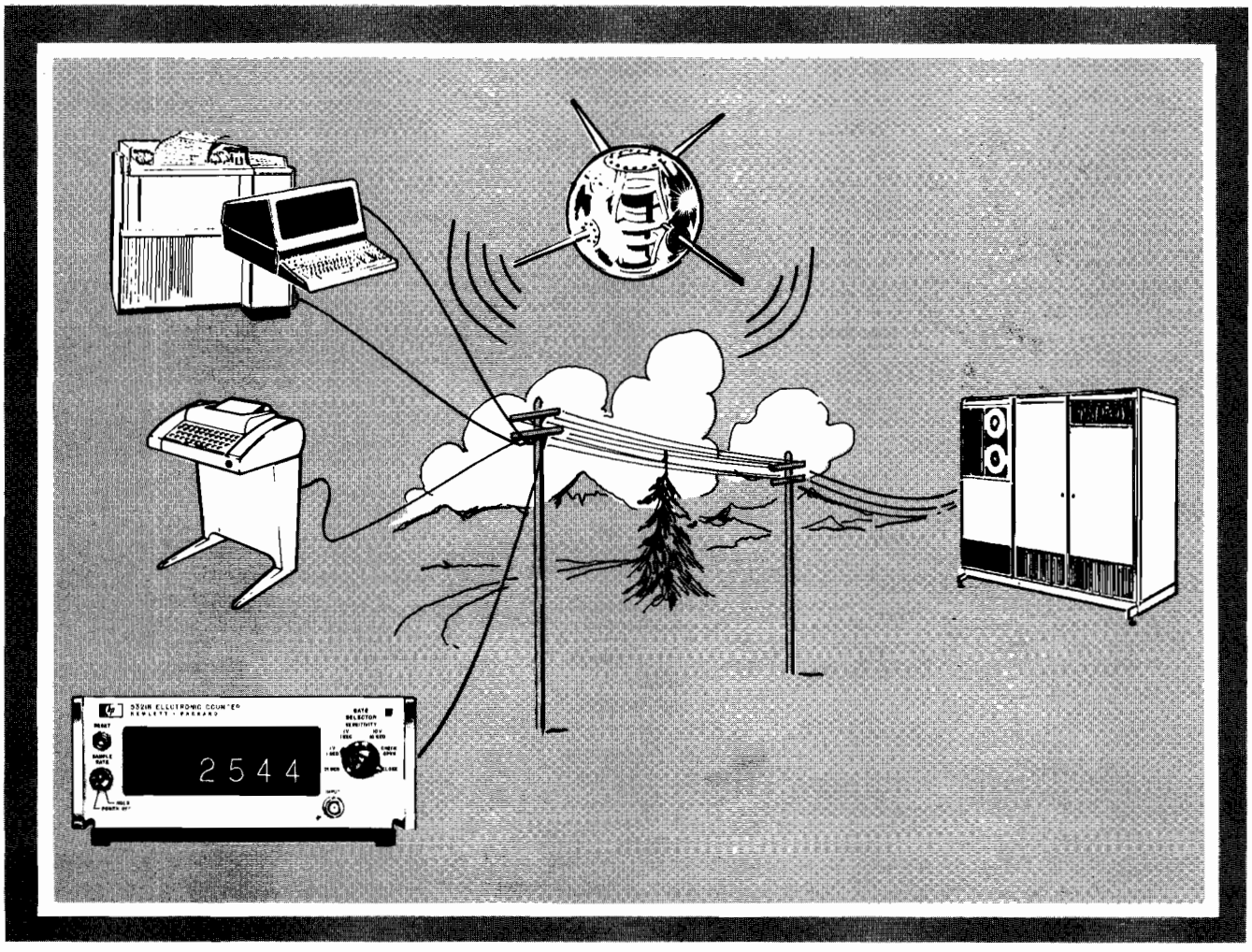
When we use a telephone to convey information in a verbal format it's called ***voice*** communication. When remote terminals and computers converse over a telephone line, it's called ***data*** communication. At some time in the not-to-distant future, the volume of data communications will equal that of voice communications.

Data communications is the transmission of information between a point of origin and a destination without altering the information content or its sequence. It requires four elements:

1. A transmitter or data source,
2. The message or information,
3. A channel or data link,
4. A receiver.

The transmitter and receiver can be either remote terminals, such as the HP 2640, or computers. The information is transmitted in one of three modes of operation:

- Asynchronous (Serial)
- Synchronous (Serial)
- Parallel



ASYNCHRONOUS

Asynchronous communication is designed to handle characters generated at random speed, such as data from a keyboard terminal. It has the advantage of low cost. Error checking is minimal, with the major portion of verification done by the operator visually checking the characters as they are printed or as they appear on a video display. Each character transmitted requires three bits of control information, two of which are the start and stop bits that come before and after each character (figure 1). This causes additional overhead and a low transfer rate (up to 1800 bits per second on a remote connection).

Hewlett-Packard pioneered asynchronous computer timesharing with the HP 2000 timeshare system and continues to use it for terminal communication.

SYNCHRONOUS

Synchronous transmission requires no start and stop bits for characters, generates less overhead, and lends itself to the transmission of large blocks of contiguous data. Synchronization of the line is accomplished by sending a special character, referred to as a SYN character, prior to the data. Transmission rates are faster than asynchronous.

Links

Synchronous communication can be categorized as either hardwired or telephone line. The hardwired link, like the parallel link, allows for direct digital transmission and total control of the media, but has a limited cable length (up to 4000 feet for the HP 12889 Hardwired Serial Interface).

Data can be transmitted over unlimited distances on switched lines (e.g. public telephone lines) or leased lines (private telephone lines). The equipment required is more complex and costly than a hardwired link, and the digital information must be converted to analog signals before transmission. The information is converted to the analog signal and readied for transmission by a device called a modem.

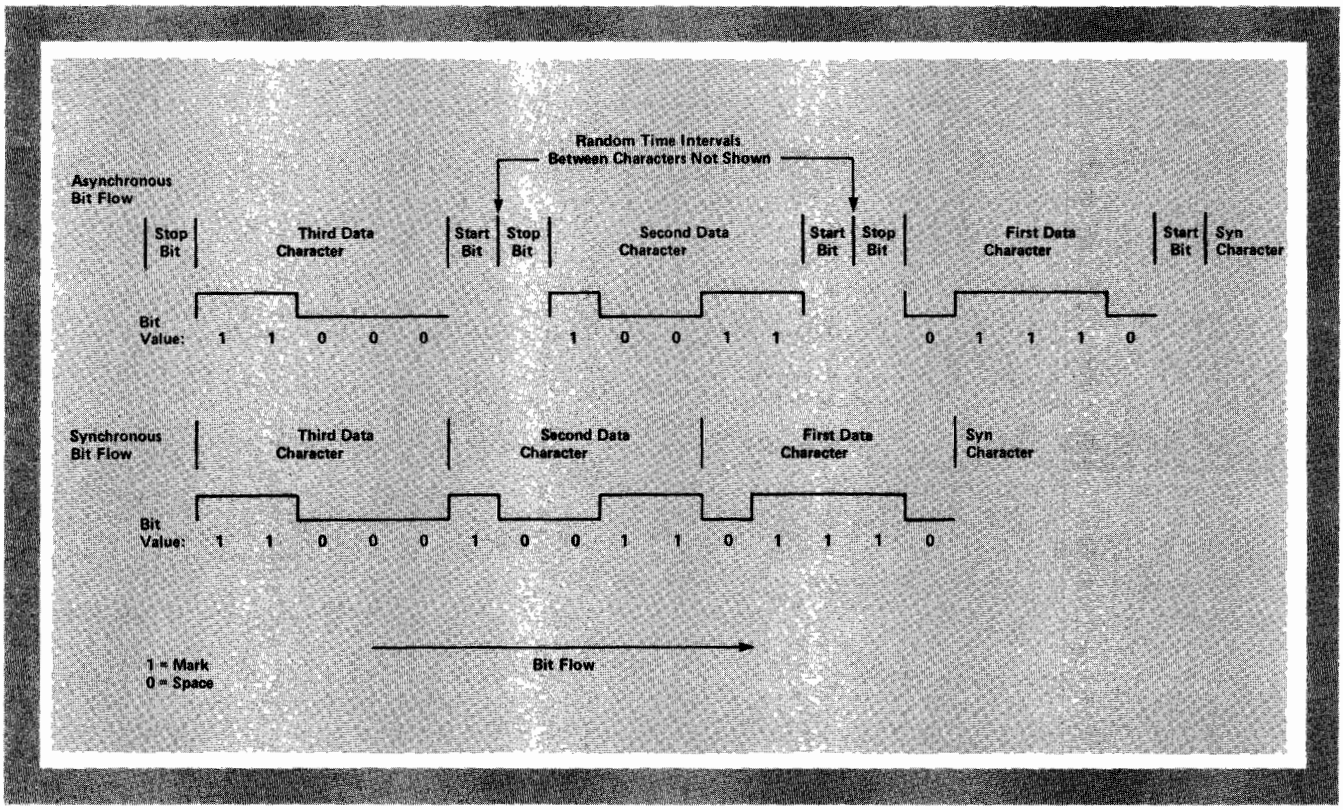


Figure 1. Asynchronous and Synchronous Bit Flows

Modems

The modem converts digital signals to analog signals and vice versa, returns status information to the computer interface, and accepts control signals from the computer interface (Figure 2). The conversion is required because telephone lines cannot carry digital signals beyond a few miles. These modems are rated in terms of their capability to answer calls automatically, and to communicate at various speeds. The most commonly used synchronous modems are the Bell 201 and 208.

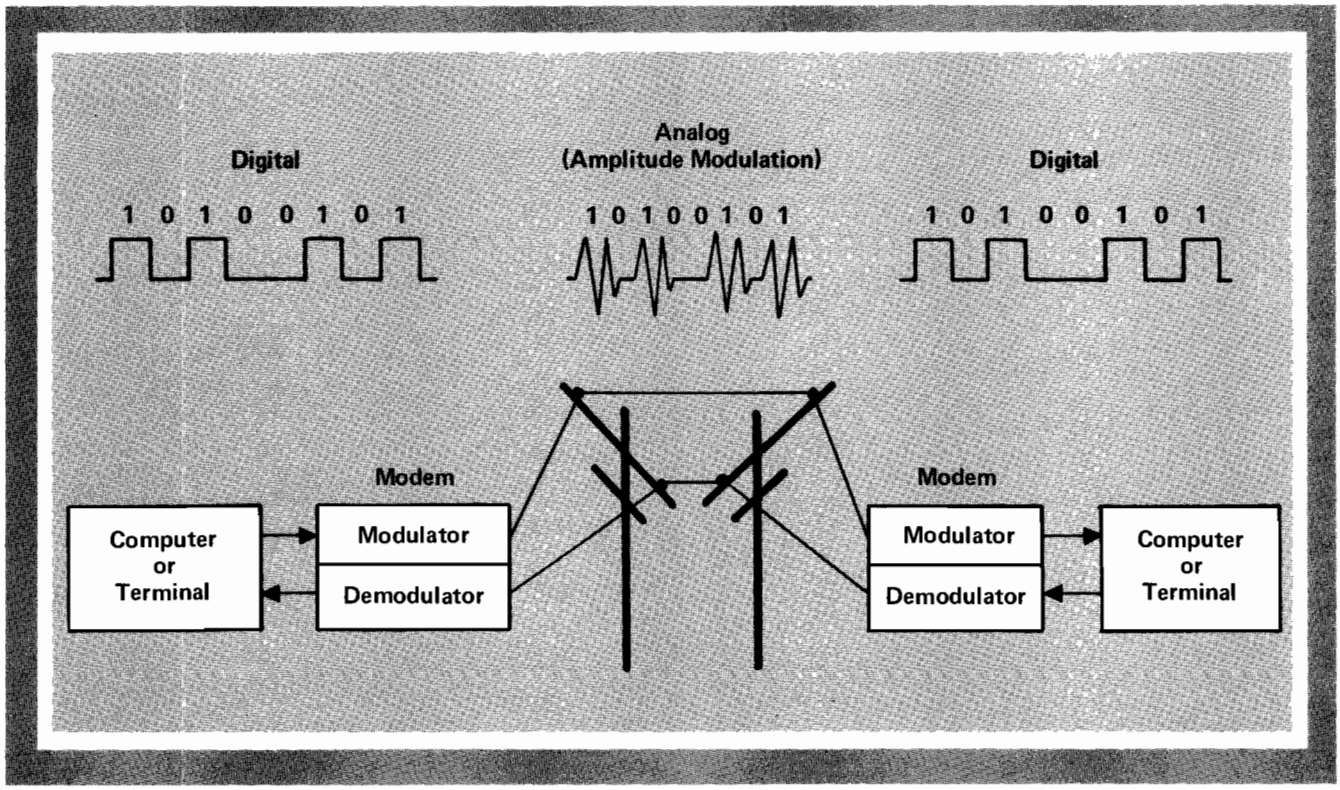


Figure 2. Modem Role in Data Communications

Line Turnaround

The transmission speed achieved by voice grade links is greatly affected by line turnaround, that is, the time required to reverse the direction of transmission from send to receive or vice versa. Turnaround time varies from 50-250 milliseconds for half-duplex (one way at a time) and can account for a large portion of the transmission time for short blocks of data. For example, the turnaround time to transmit 1000 blocks of 40 characters each is 300 seconds, while the time required to actually transmit the data is 160 seconds. There is no turnaround time for full-duplex (two way simultaneous) operations, since the direction of transmission does not have to be reversed.

Line Protocol

The control that designates who is using the line and when the line is to be turned around is determined by line protocol. The protocol distinguishes the sender from the receiver, allows for an orderly transfer of data, and permits error detection and retransmission of messages. It is implemented by using control characters and a rigorous set of rules. The most widely known and well established protocol is IBM's Binary Synchronous Communication (Bisync). Bisync defines a set of rules for synchronous transmission control by assigning specific meaning to control characters which are designed for half duplex (transmission in one direction at a time). It allows for message blocks and synchronization, control characters, transparent-text (binary data), and error detection and retransmission. Bisync is used in controlling the HP 2780/3780 terminals available on the RTE and 3000 systems and an extended version is employed in a HASP work station.

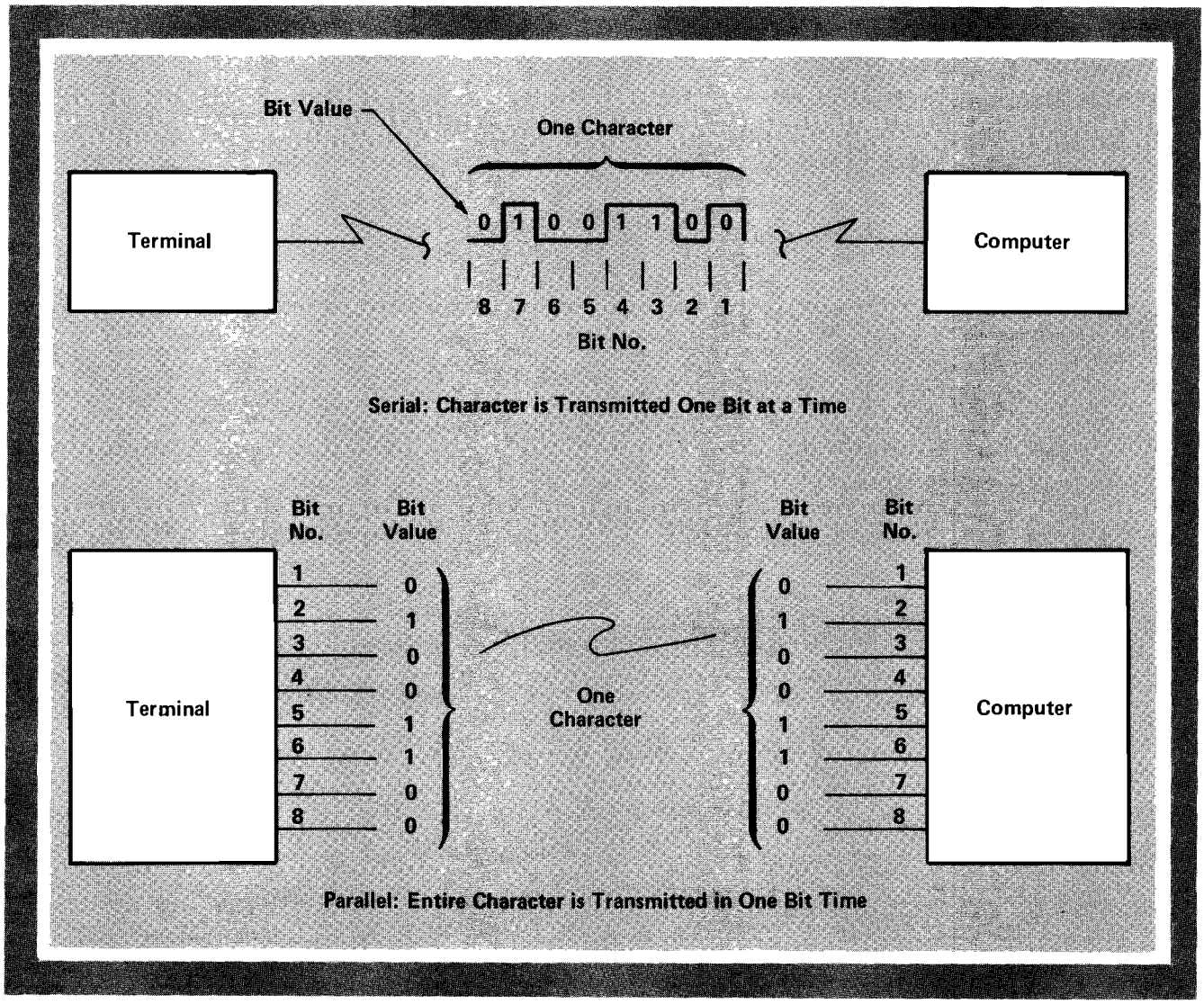


Figure 3. Serial and Parallel Bit Transmission

Serial

HP terminals and computers operating in synchronous and asynchronous modes usually transmit bit patterns in a serial stream. Where any sizeable distance is involved it is the most economical method of transmission since only one communications line is required. As shown in Figure 3, one bit is transmitted at each clock cycle.

PARALLEL

Parallel transmission allows for direct digital transmission of data at the maximum data rate and at a relatively low cost for local or in-plant use. This is accomplished by using a separate line for each bit of information comprising the character being transmitted. The drawback to parallel transmission is that typically it employs a hardwired link with a very limited range (200 to 300 feet).

Hank Ćureton/Colman McDonough
HP General Systems

software tips

KNOW YOUR ASSEMBLER

The topic chosen for this issue comes from a suggestion by Betsy Leight in the Santa Clara Office, concerning an area of data processing that is commonly recognized and generally understood, but probably lacking in knowledge of application:

Cross Assemblers

The process of developing software packages in assembly language for the purpose of simulating another manufacturer's assembly language could be called Cross Assembly or Emulation. This article will attempt to explain the purpose and use of Cross Assemblers and differentiate between other similar and conflicting terms. Some practical reasons for designing a Cross Assembler or Emulator could be:

- Limited Available Resources
- Time Constraints
- Software Conversion
- System Extension and/or Enhancement
- Sales

Although the purpose of most emulation is to give the user additional software power on his computer, others have been specifically developed to fill in where there is a lack of time and/or resources needed to convert existing routines in the native assembly language to the current routines. Regardless of the reason for developing a Cross Assembler, it has been found that most are designed and written by a user rather than by persons involved in customer support activities.

In order to get a better understanding of emulation, we must first differentiate between other conflicting terms such as "Simulation". Simulation is the technique of observing (manipulating, measuring, etc.) an artificial mechanism that represents a real-world system of activity which is not suitable for direct experimentation. The artificial mechanism is termed the "Simulation Model" which should ideally represent the essential characteristics and behavior of the actual system under study. The model is usually used to derive data about system performance by following the change of state of the system resulting from the succession of events affecting it. This can be accomplished under control of a program (perhaps a benchmark) using numerical techniques that describe the corresponding changes in the models.

When designing a simulation model, two basic objectives should always be considered:

1. It must be possible to predict performance to a certain degree before the actual system is designed.

2. It must be insured that the system design selected is optimal in terms of the stated design objectives.

Finally, during construction of the model, two key system problems must be considered:

1. The efficient use of available computer storage.

2. Minimum processing time, i.e., a logical arrangement of data that minimizes the amount of time required to access files, extract logical records, and decode pointers and other linkage information.

Some advantages of simulation are:

1. Realistic representation of complex systems that would normally lead to ill-behaved mathematical functions.

2. Determination of optimality without being concerned about stating each step in explicit form.

3. Pseudo-random events can be produced which exhibit every key property of random events.

4. Formulation of multi-variable constants except the one(s) under consideration.

Looking on the other side of the fence, about the only disadvantage of digital simulation is its relative cost in terms of design, programming, data preparation, implementation, and actual computer time required to "run the simulation model." Some application areas for simulation models are:

1. System Design – using a finite number of components of known characteristics.

2. Systems Analysis – gain understanding of how a system behaves under predefined simulated conditions.

3. Demonstration and Training – offers realistic substitutes for demonstration or training purposes that could be costly or unsafe.

4. System Formulation – matching the known behavior of the system with the processes that produced the observed behaviors that are not fully understood.

Several digital simulation languages have been developed over the years, but two languages, GPSS (General Purpose System Simulator) and CSMP (Continuous System Modeling Program), have been widely employed with considerable success in the area of simulating mechanical engineering systems. There are many more. As a matter of fact, just about any computer manufacurer in the past has had the need to construct a simulation model at one time or another. Today, it is common place for a manufacturer to have several simulation languages extending from conventional languages such as Fortran and COBOL to special purpose languages.

A term that could accurately describe a Cross Assembler would be "Emulator." Emulation is a process which establishes complete compatibility by linking basic operations of an assembly language of another computer (native) to the host computer. For example: a control program written in 2100 assembly language, which emulates (operation by operation) the assembly language of the HP 3000 could be called a Cross Assembler or Emulator.

When designing an emulator, some considerations should be given to:

1. The amount of core storage required on an emulating machine (host machine) is always larger than the storage capacity of the native machine. For example, it might take four instructions in the assembly language of the host machine to completely emulate one operation of the native machine.

2. Device and operation code (word size) restrictions in the original program which may render certain components of the host computing system inoperable.

A great deal of controversy has arisen from programs which emulate another manufacturer's assembly language. For example, the announcement of an emulator developed by Burroughs for the IBM 1401 assembly language (called Autocoder) was widespread as "a challenge to the profession as well as a scandal." Control Data Institutes throughout the country have a 360 Cross Assembler they call the IBM Simulator, which performs reasonable simulation of a limited number of instructions in the IBM Assembler. Although heavy criticism might be drawn by others, there is really no substantial means for constructive evaluation nor criticism. After all, software development is one of the most viable entities ever introduced into business. Another example of an emulator program developed on a 2100 at Stanford University emulates the IBM Assembler. This particular program is listed in the program catalog and is called "360 Simulator Program" (22046B)

Another is a program written in IBM Assembler which emulates the HP 2100 Basic Instruction Set and certain pseudo operations and is called "An Assembler for the IBM 360" (22396A).

Here is an interesting Assembler Extension problem you might find challenging:

```
        ORG 0
NEXT    NOP
        CLA
        CLB
        HLT 77B
        DEC 31740, 3840, -31680, -31291
START   LDA *-2
        LDB *-2
        JMP NEXT
        END
```

Assume your computer has a standard absolute binary loader. Simulate the process of assembling this routine and loading it with the ABL. Start execution at the label START and when the routine halts find the following:

(A) =
(B) =
(P) =

Indicate why this routine would not load correctly with the ABL if the instruction at NEXT is anything different than zero. Hint: You might have to examine the ABL routine itself!

If you have any ideas for a topic for next issue, send them to:

Communicator 9600/9700 Group
HP Data Systems Division
11000 Wolfe Road
Cupertino, Ca. 95014

REPLACING THE ON-LINE LOADR IN AN RTE II/III SYSTEM

Did you know that you can replace the on-line LOADR in an RTE II/III system? A memory patch is required, but it is a relatively simple procedure.

LOADR occupies a special niche in the heart of the system in that it is allowed to write on system tracks. When any program makes the request to write on the system tracks, its name is compared with "LOADR", which is stored as ASCII code in RTIOC, a module of the core resident operating system. By changing this ASCII name stored in memory, the system may be tricked into allowing the privileges of the LOADR to any other program. Later, we will help you find this location in memory. The basic procedure for replacing the LOADR is as follows:

1. Load the "New" LOADR with the "Old" LOADR (i.e. the one now in the system). Because the LOADR will not allow two programs with the same name, the new LOADR will be named . .ADR.

   Give . .ADR Temporary status in the system, i.e.
   *On, LOADR, < Logical Unit >

2. Patch memory so that the system treats . .ADR as its sweetheart.

3. Use . .ADR to Purge the LOADR from the system, e.g.

   ```
   *ON, . .ADR,,,4
   /LOADR: PNAME? LOADR
   /LOADR:$END
   ```

4. Use . .ADR to reload the new LOADR a second time. This time its name will be LOADR. Be sure to make it a permanent load, e.g.

   *ON, . .ADR, < LU,6,28 >

5. Reboot the system, . .ADR will no longer be a part of the system and the new LOADR will be a permanent part of the system.

The ASCII code for LOADR is stored in memory 520 octal locations from the entry point $CIC in revision C of the module RTIOC. In other revisions it may be displaced somewhat. The contents of the locations in that area in Rev C are:

LOC.	CONTENTS	COMMENTS
520	046117	LO(PATCH TO 027056 = . .)
521	040504	AD
522	051000	R -ZERO-
523	000177	
524	074000	

The procedure is identical for RTE II and RTE III. Since the system is always mapped into user partitions, you do not have to be concerned about dynamic mapping.

Jim Bridges
HP Data Systems

A PRIMER ON USING SPOOLING

Spooling capability is included in RTE III (92060A) and is optionally available in RTE II (92001A, option Y13).

The function of spooling is to increase throughput by having the programs read and/or write to disc files instead of physical devices. The system then manages the interface (read/write) between the files and the physical devices at speeds nearly independent of the running of the programs themselves.

Spooling may be used to advantage by user application programs as well as system utilities such as the Assembler, Compiler, etc. We will cover here only those applications where the effect of spooling is transparent to the trograms themselves (i.e. requires no change in the way the program is written). In these applications possible reasons for using spooling would be:

1. To speed up batch processing by the system. In the examples presented, the "batch" will be simply a series of programs to be processed sequentially by the assembler.

2. To permit user programs to read or write to/from disc files as if they were standard I/O devices (e.g. mag tape).

Number 2 is really a fallout of the spool system itself and is achieved simply by defining the files to be "standard" format. This will be explained later. There are other "fallout features" which provide convenience to the user: A full discussion would tend to obscure the purpose of this article.

This article will assume that the reader is already oriented to RTE as a user but would like a simple introduction to spooling as implemented in RTE II and RTE III, with very minimal help from the Batch Spool Monitor Reference Manual (92002-93001). The reader should be able to make immediate use of spool features from the information presented here.

To provide spooling, the batch must be entered under the system program named "JOB". In addition, spool files must be made available. Spool files come from a "spool pool" set up by the system program "GASP" or they are created by the user. When using the spool pool, operation of spooling is nearly transparent, i.e. very little knowledge is required. Files in the spool pool are named "SPOLXX", where XX is a number from 01 to 80. The system assumes that no user will access these files by name. Therefore, the system treats the contents of spool pool files as temporary and available for whatever needs may arise. The spool system may also use specific files created and named by the user (not "SPOLXX"). These files are used by the spool system only as directed by the user commands.

A user created file or a spool pool file will be automatically extended by the spool system. Each time a spool file is extended, an area of the disc equal to the original size of the file is added to the file. This addition is called an "extent". A separate directory entry is made for each extent. If a file is created with a large number of blocks and extents are added, the space in the last extent may not be filled. The opposite approach is to create a small file. This will result in small (equal size) extents and so the unused space in the last extent will be of less consequence. However, the number of extents may become very large and may overrun the available space in the directory. Hence the two factors must be balanced according to the individuals' needs.

To start the user off, a sample dialogue with the system program GASP (Great Automatic Spool Program) is presented below. User responses to the questions are underlined. A question asked repeatedly is terminated with "E" (for end).


```
*ON,GASP

MAX NUMBER OF JOBS, JOB FILE DISC? 126,13
NUMBER OF SPOOL FILES (5 TO 80)? 5
SIZE OF SPOOL FILES (IN BLOCKS)? 24
NUMBER, LOCATION OF SPOOL FILES? 5,13
NUMBER, LOCATION OF SPOOL FILES? E
MAXIMUM NUMBER OF ACTIVE AND PENDING
SPOOL FILES? 5
ENTER OUTSPOOL DESTINATION LU 6
ENTER OUTSPOOL DESTINATION LU 8
ENTER OUTSPOOL DESTINATION LU 4
ENTER OUTSPOOL DESTINATION LU E
END GASP
```

In addition to spool pool files, GASP also sets up "JOBFIL" and "SPLCON". These two files are used for bookkeeping operations relating to inspooling and out-spooling.

Note that the spool pool files are placed on cartridge "13". It is a good idea to assign a separate cartridge for spool pool files (created by GASP). When the system is finished with a file from the spool pool, it purges the extents created on the file. If the extents lie between the original file and any user file, the disc space released is not available until the cartridge is repacked (":PK" command).

The outspool destination LU's should include all the output devices which the spool system will use.

A job (submitted to the program "JOB") is a set of instructions delineated by a JOB command and an EOJ command. In the example below, only spool pool files (created by GASP) are used. The system assigns the files as needed and gives them up when it is done using them.

:JOB,JOBA	Start of Job, called "JOBA"
:LS	Reset LS area
:LG,5	Give 5 tracks for relocatable binary
:MS,&PROGA,ASMB	Move source to LS and give to Assembler
:RU,ASMB,2,99	Input from LS, output to LG
:MS,&PROGB,ASMB	Second program to LS
:RU,ASMB,2,99	
:SA,LG,%RELO	Save Relocatable binary in file %RELO
:EOJ,JOBA	Marks end of JOB

If JOBA is in a file named "&JOBA", then it may be run as follows:

*ON,JOB,&J,OB,A

If JOBA is on cards, place the cards in the reader (logical unit 9) and enter the job with the command:

*ON,JOB,9

Note that in either case &PROGA and &PROGB are assumed to be already on the disc. But if JOBA is in the card reader, it must first be read onto the disc before it can be processed. In this case, the job is read into a spool pool file (created by GASP) as fast as it can. A quick check of the :JOB and :EOJ commands is made to determine name, priority, etc. to be stored in JOBFIL and to find out where the job begins and ends. A logical error (such as failing to assign tracks in the LG area) would not be detected in this phase. Logical errors are detected at the time the job is actually run. This reading in of the job "instructions" is known as "inspooling" (versus outspooling). Inspooling and outspooling are completely separate operations, as is the running of the job itself. In the case where the JOBA is already on the disc, no inspooling need take place. This does not mean that it is better since the file is not necessarily of long term value.

Let's discuss how outspooling is used to advantage in the above example (JOBA). During the first pass of the Assembler, no output takes place, so the line printer is idle during pass one of &PROGA. During the second pass, the Assembler writes the listing to a file in the spool pool. Since it can write to a file on the disc much faster than to the line printer, it completes pass two more quickly. The Assembler then goes to pass one for &PROGB while the spool system is printing the listing for &PROGA. If it should finish pass one before the previous listing is printed, it will not have to wait. The next listing will be appended to the spool file, thus the speed of processing the source is "decoupled" from the speed of the line printer, which increases throughput.

A second example will show how (with a little more complexity) the speed of JOBA can be improved. See JOBB below. In JOBB, the use of the LS area is omitted, eliminating the need for the system to move the source before the assembler can use it. For a big file the time savings can be appreciable.

```
:JOB,JOBB
:LG,5
:LU,8,&PROGA
:RU,ASMB,8,99
:LU,8,&PROGB
:RU,ASMB,8,99
:SA,LG,%RELO
:EOJ,JOBB
```

In JOBB, logical unit 8 is "set up" by the LU command to be equivalent to file &PROGA. The actual logical unit need not be "8" but can be any logical unit recognized by the system (i.e. declared at generation) and subject to the rules discussed in the reference manual.

The file &PROGA is not "inspooled" in any way by this equivalence. A side effect of this equivalence is that more than one source program may appear in the same file. If the source is in the LS area, only the source last loaded will be used: thus &PROGB cannot be concatenated to &PROGA. But using the LU equivalence, source for &PROGA and &PROGB may be in the same file and the Assembler merely run twice to accomplish both assemblies. In this way, the source file looks like a physical device (e.g. mag tape) to the Assembler.

The next logical step is to eliminate the need for the LG tracks by spooling the binary to user file. See JOBC below. The reader is cautioned against thinking that the total need for the LG area is eliminated, however. The loader cannot input directly from a file, as we shall see later. Note that the file "%RELO" must be created by direct action; it is not automatically created by the spooling system. For example, using the File Manager, give the command:

:CR,%RELO:::5:24

The file "%RELO" is thus created as type 5 with 24 blocks. No cartridge or security code is specified: these parameters are optional.

```
:JOB,JOBC
:LU,8,&PROGA
:LU,4,%RELO,WRST
:RU,ASMB,8
:EOJ,JOBC
```

In JOBC, it is assumed that the control statement in &PROGA has the optional parameter "B", e.g. "ASMB,B,L". Otherwise, no binary will be available. Note that "%RELO" is a standard file (i.e. "ordinary" type 5) and also has the attribute of being a "write only" file. This does not mean that the file cannot be read later. The default condition is a read only file. The "WR" tells the system that it may write in the file but not read from it (we don't plan to have the spool system read the file). The file must be declared "standard" (ST) in order to prevent the spool system from writing control characters into the file which are specified in the original I/O call.

The next step in complexity might be to outspool the listing to a file which would not be printed until the end of the day (or maybe the night shift). To do this we must create a file for listing so that a spool pool file is not used. To create a list file, for example:

:CR,&LIST:::4:48 (type 4, 48 blocks long)

The job is given below:

```
:JOB,JOBD
:LU,8,&PROGA
:LU,4,%RELO,WRST
:LU,6,&LIST,WR
:RU,ASMB,8
:EOJ,JOBD
```

The file "&LIST" will not be actually printed until another job (see JOBE) is run:

```
:JOB,JOBE
:LU,6,&LIST,,6
:EOJ,JOBE
```

The second "6" in the LU command above tells the system to outspool the file to logical unit 6.

If we wished to have two copies of the listing and purge the file afterwards, we could use JOBF (see below), where the "PU" purges the file after outspooling.

```
:JOB,JOBF
:LU,6,&LIST,,6
:LU,6,&LIST,PU,6
:EDJ,JOBF
```

To this point, the only use we have made of inspooling is to (perhaps) read (and inspool) the job commands from a card deck. For convenience, a card deck can be made to contain the job control commands and the assembly language source, in which case the entire deck is inspooled. See JOBG below. The deck is read very rapidly, not limited by the processing speed of the Assembler.

```
*ON,JOB,9  (Logical unit 9 assumed to be a card reader)
:JOB,JOBG
:LU,4,%RELO,WRST
:RU,ASMB
ASMB,B,L
        NAM IRAL,7  Fortran callable routine to rotate left
        ENT IRAL    Call IRAL (word, nbits)
        EXT .ENTR
WORD    NOP         Word to rotate
BITS    NOP         Number of bits to rotate left
IRAL    NOP         Entry
        JSB .ENTR
        DEF WORD
        LDA WORD,I
        LDB BITS, I
        CMB,INB,SZB Set negative, exit if zero count
        RSS         Skip next instr unless zero count
        JSB IRAL, I
        RAL
        ISZ 1       Skip if count up
        JMP *-2     Else continue to rotate bits
        JMP IRAL, I
        END
:EOJ,JOBG
```

A job deck may be assembled in this manner because the logical unit from which the command ":JOB" is received is equated to logical unit 5. Logical unit 5 is the default input for the Assembler. The "turn on" unit (or NAMR) is stored in global parameter 0G. Thus, in the above example, ":RU,ASMB,0G" would have worked equally well.

If we wished, we could store the entire card deck for JOBG (above) in a disc file and run with the command:

```
*ON,JOB,&J,OB,G
```

If the program is very large, we might not wish to outspool the listing (because it would take up a lot of disc space and might overrun). In that case, the job file would precede the program source and, in addition, the job command would use the "NS" parameter. See JOBH below. The "NS" parameter inhibits spooling of output to logical unit 6. Under the NS option, output to logical unit 6 is done directly: if the device is busy the job waits.

```
:JOB,JOBH,NS
:LU,4,%RELO,WRST
:RU,ASMB,9
:EOJ,JOBH
ASMB,B,L
        NAM IRAL,7  Fortran callable routine to rotate left
        ENT IRAL    Call IRAL (word, nbits)
        EXT .ENTR
WORD    NOP         Word to rotate
BITS    NOP         Number of bits to rotate left
IRAL    NOP         Entry
        JSB .ENTR
        DEF WORD
        LDA WORD, I
        LDB BITS, I
        CMB,INB,SZB Set negative, exit if zero count
        RSS         Skip next instr unless zero count
        JSB IRAL,I
        RAL
        ISZ 1       Skip if count up
        JMP *-2     Else continue to rotate bits
        JMP IRAL,I
        END
:EOJ,JOBH
```

We conclude with two cautionary examples. The first (as mentioned earlier) is that the loader cannot input directly from a file. For example:

```
:JOB,JOBI
:LU,4,%RELO
:RU,LOADR,4
:EDJ,JOBI
```

The above job cannot be successfully run because whenever LOADR inputs from a logical unit not equal to 99 (the LG area), it suspends itself for operator intervention (*GO,LOADR,<some option>). This is currently not permitted in a job and thus will cause an abort.

The second note of caution is that the operator should not run the job from within the File Manager (FMGR) by transferring control to a logical unit (e.g. :TR,8). The reason is that FMGR resets the global parameter 0G to the file NAMR or LU from which it received the ":JOB" command. If a program is run from FMGR using default parameters, FMGR will pass 0G as the first parameter. The example below illustrates the problem:

```
*ON,FMGR
:TR,8
:JOB,ERROR     (THIS IS A SAMPLE)
<JOB STREAM ECHOES ON TERMINAL>
:EOJ,ERROR
:RU,EDITR
```

What happens in this case is that EDITR is not told by the operator what terminal to use for operator input. Therefore, the FMGR tells EDITR to use 0G. Initially, 0G was set to the terminal from which FMGR was run. But it was reset by the unit which issued the job command (i.e. 8). Therefore, EDITR goes to logical unit 8 to receive input.

A simple expedient is (1) don't run JOB from FMGR or (2) always pass the terminal logical unit to EDITR (e.g. :RU,EDITR,1).

Jim Bridges
HP Data Systems

A NOTE ON INITIALIZING CERTAIN 21MX EIG INSTRUCTIONS

Several EIG instructions have temporary fields included within them that are created during assembly. These instructions may require initializing by the programmer in real time situations. The instructions are:

MVW	move words
MBT	move bytes
CMW	compare words
CBT	compare bytes

When assembled each instruction generates a DEF to its operand and a NOP. Thus, when assembled:

```
MVW     COUNT   =    MVW  COUNT
                     DEF  COUNT
                     NOP
```

An example of this with a listing and an assembly of the listing is shown below.

```
0001  ASMB,L
0002        NAM PROG
0003        ENT PROG
0004  PROG  NOP
0005        MVW COUNT
0006  COUNT DEC 10
0007        END
0008        END$
```

```
0001                ASMB,L
0002  00000                 NAM PROG
0003                        ENT PROG
0004  00000 000000  PROG    NOP
0005  00001 105777          MVW COUNT
      00002 000004R
      00003 000000
0006  00004 000012  COUNT   DEC 10
0007                        END
**  NO ERRORS *TOTAL **RTE ASMB 750420**
```

The NOP temporary field is used by the micro-code for retaining a counter when the instruction is not allowed to complete due to the CPU recognizing an interrupt. During normal operation a count is placed in the NOP location so that if the micro-code is interrupted it may be re-entered.

It is possible that while the CPU is handling an interrupt, the process executing one of the four mentioned instructions may be aborted. If this situation should occur with a core resident program, the next time the process is scheduled avoid the possibility of the instruction executing improperly.

```
START  CLA
       STA LABL1
       STA LABL2
        .
        .
        .
       STA LABLN
        .
        .
        .
       MVW CNT1
LABL1  EQU *-1
        .
        .
        .
       MBT CNT2
LABL2  EQU *-1
        .
        .
        .
       CBT CNT3
LABLN  EQU *-1
```

The assembler will set the LABLN's to point to the temporary field of the preceding instruction. Now upon entry to the process the programmer is assured his code will always execute properly.

Earl Stutes
HP Data Systems

Software Sam

Dear SAM:

Enjoyed last issued article on the program IGET. I have used IGET during the past month to create a number of programs which wander through the communication area of the system and print out status information on equipment, programs and the system itself.

I tried to improve upon IGET with a program IPUT which places entered values into locations specified. However, the program only works part of the time. Whenever I try to place a word on base page, my program is aborted with a memory protect. Can you tell me why?

NAME WITHHELD BY REQUEST

Dear ANONYMOUS

SAM has looked over your program and finds that you forgot one crucial factor. That is, RTE II & RTE III allow you to read but not write past the memory protect fence. Since program IGET is a read only operation, no memory protect is generated. However, if the same technique is used to write, RTE in an effort to protect itself, aborts the offending program. An assembly language program containing a CLF 0 will also be aborted by RTE for the same reason.

Do not despair! SAM has a simple interactive program (called IPUT in honor of our first customer letter) which will allow the user to perform system surgery on RTE.

The program is broken up into a Fortran main program and an assembly subroutine. Line 24 of the main program calls the subroutine, SUB, and passes it the new value and location. The rest of the Fortran main program is simple interactive dialogue which SAM leaves as an exercise for the reader.

The assembly subroutine really does most of the work by calling system routines $LIBR to turn off the interrupt system and $LIBX to turn the interrupt system back on. The only other external to the assembly subroutine is .ENTP. This external is merely a different entry point into the system routine .ENTR. Thus .ENTP places the address of IVALUE and LOCATN into PRAM1 and PRAM2 respectively. Lines 15, 16, and 17 place the value passed into the location specified.

Many examples of how to use IPUT should come to mind. For example, if you have made a generation error requiring only a few words to be changed during boot up, then IPUT is an excellent solution to patching the system. This would save hours or even days of trouble shooting and regeneration. However, one word of caution is in order. Indiscriminant use of IPUT can cause disastrous results. Use IPUT sparingly and with discretion.

Main Program

```
0001  FTN4,L
0002        PROGRAM IPUT
0003  C
0004  C ASK FOR REPLACEMENT VALUE
0005  C
0006  10    WRITE(1,25)
0007  25    FORMAT("ENTER VALUE")
0008        READ(1,100) IVALUE
0009  100   FORMAT(I6)
0010  C
0011  C GET LOCATION FOR REPLACEMENT VALUE
0012  C
0013        WRITE(1,105)
0014  105   FORMAT("ENTER LOCATION")
0015        READ(1,100) LOCATN
0016  C
0017  C ECHO CHECK TO MAKE SURE PROPER VALUES ENTERED.
0018  C
0019  110   WRITE(1,120) IVALUE,LOCATN
0020  120   FORMAT("VALUE = ",I6,"   LOCATION = ",I6)
0021  C
0022  C CALL THE IPUT SUBROUTINE
0023  C
0024        CALL SUB(IVALUE,LOCATN)
0025  C
0026  C SEE IF THERE IS ANOTHER VALUE TO PUT.
0027  C
0028        WRITE(1,150)
0029  150   FORMAT("ANOTHER LOCATION AND VALUE ?")
0030        READ(1,200) ITRY
0031  200   FORMAT(A2)
0032        IF(ITRY .EQ. 2HYE) GO TO 10
0033        END
0034        END$
```

Subroutine

```
0001  ASMB,L
0002        NAM IPUT,6
0003        ENT SUB
0004        EXT $LIBR,.ENTP,$LIBX
0005  PRAM1 NOP                 THIS IS THE VALUE
0006  PRAM2 NOP                 THIS IS THE LOCATION
0007  SUB   NOP                 SUBROUTINE ENTRY POINT
0008  *
0009  *                         TURN OFF INTERUPT SYSTEM
0010  *
0011        JSB $LIBR
0012        NOP
0013        JSB .ENTP           GET LOCATIONS OF PARAMETERS
0014        DEF PRAM1                 IN MAIN PROGRAM
0015        LDA PRAM1,I         PUT VALUE INTO A REG
0016        LDB PRAM2,I         PUT LOCATION INTO B REG
0017        STA 1,I             STORE VALUE INTO LOCATION
0018  *
0019  *                         TURN INTERUPT SYSTEM BACK
0020  *                         ON
0021  EXIT  JSB $LIBX
0022        DEF SUB
0023        END
0024        END$
```

SAM would suggest that the reader keep this issue and the last issue handy for future reference. Many system problems can be solved by judicious use of IPUT and IGET. These routines will probably be used again and again.

If you have questions, suggestions, or comments about your 9600 system, let SAM help. Write to:

SOFTWARE SAM
c/o **Communicator** 9600/9700 Group
HP Data Systems Division
11000 Wolfe Road
Cupertino, Ca. 95014

bulletins

RTE-III: A GUIDE FOR NEW USERS

Here is a manual for new users of the RTE-III Operating System who want to use the system primarily for program development. It contains step-by-step instructions for turning on the equipment, loading the system, entering and editing a program, and then compiling and running it. Each step is illustrated with sample procedures that you can follow easily as you sit at the console.

Possible errors are described in the context of where they might occur and recovery procedures are explained. In addition, there is a special troubleshooting section that shows you how to diagnose problems using tools such as the WHZAT program.

The RTE-III Operating System provides a variety of ways to prepare and run programs. After describing a typical way to prepare a Fortran program, the manual describes alternative methods including batch job submittal, loading subprograms and segments, using the Multi-Terminal Monitor, and any differences in procedure if your program is written in a language other than Fortran.

Written specifically for the RTE-III Operating System, this manual will be just as useful for anyone beginning to use RTE-II. It does not cover the many intricacies of the very flexible RTE system; for these, you should turn to the Reference Manuals that describe the operating system and Batch-Spool Monitor.

The HP part number is 92060-90012 and the price is $6.50.

Joan Martin
HP General Systems

RTE INTERACTIVE EDITOR MANUAL

A separate RTE Interactive Editor Reference Manual is now available. The original description of EDITR in the RTE-II/III Reference Manuals has been expanded.

The first two sections of the new manual tell you all you need to know to do simple editing tasks. Initiating EDITR, line edits, character, edits, and pattern edits are introduced simply with extensive examples.

The command section of the manual is designed for your use as you enlarge your knowledge of how EDITR works and want to make more efficient use of its features. Each command is described in depth with examples and special features.

The RTE Interactive Editor Reference Manual also includes instructions for initiating EDITR either interactively or in batch mode, with or without spooling, and in a multi-terminal environment.

The price of the new book is $6.00. HP part number is 92060-90014.

Carol Guddal
HP Data Systems

software updates

The material lists below show all software parts and manuals which are shipped with RTE II and RTE III. Note that the batch spool monitor is option Y13 on RTE II.

SOFTWARE	RTE II		RTE III
Core Res Sys	92001-16012		92060-12003
Loader	92001-16002		92060-16004
Multi Term Monitor	92001-16003		Same
Sys Library	92001-16005		Same
DVP43 (Power Fail)	92001-16004		92060-16001
Autor (Auto Restart)	92001-16014		Same
Autor Source	92001-18014		Same
Spool Monitor	92002-12001		Same
Spool Program	92002-12002	Option	92060-12001
Batch Monitor Libr	92002-16006	Y13	Same
EDITR	92002-16010		Same
EDITOR	20805-60001		Not included

DVR00	29029-60001	Same
DVR31 (7900 Disc)	29013-60001	Same
DVR32 (7905 Disc)	92060-16031	Same
7900 System Generator	92001-16013	92060-16029
Fixed Head Generator	92001-16018	Not Supported
7905 System Generator	92001-16026	92060-16032
WHZAT	Not Available	92060-16006
Assembler	92060-12004	Same
XREF	92060-16028	Same
$PVMP	Not applicable	92060-16035
FORTRAN II	20875-60001	Same
	20875-60002	Same
	20875-60003	Same
	20875-60004	Same
	20875-60005	Same
FORTRAN IV	24170-60001	Same
	24170-60002	
	24170-60003	Same
FORTRAN IV (10K Area)	24177-60001	Same
	24177-60002	
ALGOL	24129-60001	Same
	24129-60002	Same
RELO SUBR LIBR-EAU	24151-60001	Same
RELO SUBR LIBR-FP	24248-60001	Same
FFP SUBR LIBR	12977-16001	Same
FTN4 SUBR LIBRARY	24152-60001	Same
FTN II FORMATTER	24153-60001	Same
16K SIO MAG TAPE 9 TR	13022-60001	Same
16K SIO MAG TAPE 7 TR	13030-60001	Same
16K SIO PAPER TAPE RDR	20319-60001	Same
16K SIO PAPER TAPE PUNCH	20320-60001	Same
16K SIO SYSTEM DUMP	20335-60001	Same
16K SIO TTY (LP COMPAT)	24127-60001	Same
16K SIO 2767 LP	24166-60001	Same
16K SIO 2762/2615	24329-60001	Same
16K SIO 2607 LP	24347-16001	Same
PUNCH/VERIFY	20312-60001	Same
24K SIO TTY (LP COMPAT)	29100-60017	Same
24K SIO SYSTEM DUMP	29100-60018	Same
24K SIO PAPER TAPE RDR	29100-60019	Same
24K SIO PAPER TAPE PUNCH	29100-60020	Same
24K SIO 2767 LP	29100-60022	Same
24K SIO MAG TAPE 9 TR	29100-60023	Same
24K SIO MAG TAPE 7 TR	29100-60049	Same
24K SIO 2762/2615	29100-60050	Same

SOFTWARE	RTE II		RTE III
Manuals			
RTE REFERENCE	92001-93001		92060-90004
BATCH/SPOOL MONITOR	92002-93001	Option Y13	Same
ASSEMBLER	92060-90005		Same
FORTRAN II	02116-9015		Same
FORTRAN IV	5951-1321		Same
ALGOL	02116-9072		Same
RELO SUBR	02116-91780		Same
SIO DVR 2762/2615	02762-90002		Same
SIO DVR 2767 LP	12653-90004		Same
SIO DVR 2607 LP	12987-90006		Same
SIO DVR 7970 B/E 9 TR	13022-90010		Same
SIO DVR 7970B 7 TR	13029-90010		Same
SIO INTRO	5951-1369		Same
SIO SYS CONFIG.	5951-1374		Same
SIO SUBSYS	5951-1390		Same
ERROR MESSAGES	5951-1377		Same
DVR00	29029-95001		Same

documentation

The following tables list currently-available customer manuals for Data Systems Division products. This list supersedes the list in the last issue of the **Communicator**.

We have added two tables: "Recent Publications," which contains documentation we recently have added (new manuals) or changed (revised or updated manuals); and "Obsoleted Publications," which lists documentation no longer supported by Data Systems Division (but which still can be ordered from CPC).

The manuals in these tables will change with each issue of the **Communicator**. "Recent Publications" from the previous issue will be incorporated into the appropriate list, while "Obsoleted Publications" (if any) from the previous issue will no longer be reported.

Copies of manuals and updates can be obtained from your local Sales and Service Office. The address and telephone number of the office nearest to you are listed in the back of all customer manuals.

Customers in the U.S. may also order directly by mail. Simply list the name and part number of the manual(s) you need on the Corporate Parts Center form supplied at the back of the **Communicator**. If you require an update package only, send your request to:

Software/Publications Distribution
11000 Wolfe Road
Cupertino, Ca. 95014

A few words about documentation terms:

New — A new manual refers only to the first printing of a manual. When first printed, a manual is assigned a part number.

Revised — A revised manual is a printing of an existing manual which incorporates new and/or changed information in its contents. For example, a manual is revised when an update package is incorporated into the manual: the manual gets a new print date and the update package disappears. Note that a revision to a manual effectively obsoletes the previous version of the manual.

Update — An update package is a supplement to an existing manual which contains new and/or changed information. Updates are issued when information must get to customers, yet it is inappropriate to issue a revised manual. An update has no part number; it is automatically included when you order the manual with which it is associated.

RECENT PUBLICATIONS

PART NUMBER	TITLE	REVISED NEW UPDATE	DATE	PRICE*
29029-95001	RTE Driver DVR00 for Multiple-Device System Control	R	11/75	$ 1.50
92001-93001	RTE-II Software System Programming and Operating Manual	R	8/75	10.00
24376-90001	IMAGE/2000 Data Base Management System Reference Manual	R	8/75	11.00
02116-9014	HP Assembler	R	8/75	6.50
29103-93001	RTE System Cross Loader Programming and Operating Manual	U	11/75	—
24307-90012	DOS-III Data Communications Drivers Reference Manual	U	11/75	—
92060-90010	RTE, Batch/Spool Monitor and Operating System Pocket Guide	N	10/75	3.00
92060-90012	RTE-III: A Guide for New Users	N	10/75	6.50
92060-90014	RTE Interactive Editor Reference Manual	N	10/75	6.00
92060-90016	Multi-User Real-Time BASIC Reference Manual	N	10/75	12.00
29102-93001	RTE BASIC Software System Programming and Operating Manual	U	8/75	—
09610-93003	ISA FORTRAN Extension Package Reference Manual	R	10/75	2.50
12560-90023	DOS, RTE, and BCS Calcomp Plotter Drivers	R	10/75	1.50
92060-90004	RTE-III Software System Programming and Operating Manual	R	10/75	12.00

OBSOLETED PUBLICATIONS

PART NUMBER	TITLE	DATE OBSOLETED
5951-1377	FORTRAN, ALGOL, and Assembler Error Messages	30 Oct 75
24380-90001	HPRJE HP 2100 Remote Job Entry Processor	31 Oct 75

*Prices subject to change without notice

9600/9700 SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02005-90001	Real-Time Executive Software System	$12.00	10/71	
02313-93002	RTE 2313B Analog-Digital Interface Subsystem Operating and Service Manual	12.50	2/75	
02320-93002	RTE System Driver DVR76 for HP 2320A Low Speed Data Acquisition Subsystem Programming and Operating Manual	1.00	8/74	
02321-93001	RTE System Driver DVR74 for HP 2321A Low Speed Data Acquisition Subsystem Programming and Operating Manual	1.00	8/74	
09600-93010	RTE System DVR11 for HP 2892A Card Reader Programming and Operating Manual	1.00	8/74	
09600-93015	91200A TV Interface Kit; Programming and Operating Manual	4.50	07/75	
09601-93007	RTE Device Subroutine for HP 5327A/B-H48 Counter	2.50	12/74	
09601-93009	RTE Device Subroutine for HP 5326A-H18 Counter	2.50	12/74	
09601-93014	RTE System Driver DVR15 Mark Sense Card Reader Programming and Operating Manual	1.00	08/74	1/75
09601-93015	RTE for 40-bit Output Register #12556B	1.00	10/74	
09603-93001	9603A/9604A Control System and Scientific Measurement Operating and Service Manual	7.50	06/75	
09610-93003	ISA FORTRAN Extension Package Reference Manual	2.50	10/75	
09611-90009	9611A Operating 406 Industrial Measurement and Control System	.25	4/75	
09611-90010	HP 6940A/B Multiprogrammer Verification Manual	4.50	8/75	
12604-93002	RTE DVR40 for 12604B Data Source Interface	1.00	8/74	
12665-93001	RTE System Driver DVR65 for HP 12771A Computer Serial Interface Kit	1.00	8/74	

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
12989-99001	RTE System Driver DVA 15 for Card Reader Punch Subsystem 2894	1.00	1/75	
25117-93003	RTE System Driver DVR24 for HP 7970 Series Digital Magnetic Tape Unit	1.00	8/74	
29003-93001	RTE System Driver DVR66 for HP 12772A Coupler Modem Interface Kit Programming and Operating Manual	1.00	8/74	
29003-93003	RTE System Driver DVR66 for HP 12770A Coupler Serial Interface Kit Programming and Operating Manual	1.00	8/74	
29009-93001	RTE System Driver DVR62 for HP 23138 Subsystem	2.50	8/74	
29013-90001	DVR31 RTE Moving Head Driver	10.00	2/73	
29014-90001	Moving Head Real-Time System Generator	20.00	4/72	
29015-90001	Fixed Head Real-Time System Generator	15.00	4/72	
29016-90002	RTE Scheduler	50.00	9/72	
29016-90003	Real-Time Input/Output Control	50.00	2/73	
29022-90001	Real-Time Relocating Loader	10.00	6/73	
29028-95001	RTE HP 2610A/2614A Line Printer Driver	1.50	8/73	
29029-91001	Real-Time Executive Multiple-Device System Control Device (DVR00) Program Listing	10.00	9/72	
29029-95001	Real-Time Executive System Driver DVR00 for Multiple Device System Control Small Programs Manual	1.00	11/75	
29033-98000	Real-Time Executive-File Manager System	10.00	3/73	
29100-93001	RTE System Driver DVR40 (29100-60041) for HP 12604B Data Source Interface Programming and Operating Manual	1.00	8/74	10/74
29100-93003	RTE System Driver DVR61 for HP 6940A, 6941A Bidirectional Multiprogrammer Programming and Operating Manual	3.00	8/74	
29101-93001	RTE Core-Based Software System Users Manual	5.00	8/73	
29102-93001	RTE BASIC Software System Programming and Operating Manual	10.00	3/74	8/75
29103-93001	RTE System Cross Loader; Programming and Operating Manual	2.50	3/75	11/75
91060-93005	RTE Driver for X-Y Display Storage Subsystem (HP Model 1331C-016) Programming and Operating Manual	1.00	8/74	
91062-93003	Real-Time Executive System Driver for DVM/Scanner Subsystem	9.00	8/74	
92001-93001	Real-Time Executive II Software System Programming and Operating Manual	10.00	1/75	
92002-93001	RTE Batch-Spool Monitor Programming and Operating Manual	$10.00	2/75	05/75
92060-90004	RTE-III Software System Programming and Operating Manual	12.00	10/75	
92060-90005	RTE Assembler Reference Manual	7.00	5/75	
92060-90010	RTE Batch/Spool Monitor and Operating System Pocket Guide	3.00	10/75	
92060-90012	RTE-III: A Guide for New Users	6.50	10/75	
92060-90014	RTE Interactive Editor Reference Manual	6.00	10/75	
92060-90016	Multi-User Real-Time BASIC Reference Manual	12.00	10/75	
92200-93001	RTE System Driver DVR12 for HP 2607A Line Printer Programming and Operating Manual	1.00	8/74	
92200-93005	Real-Time Executive Operating System Drivers and Device Subroutine Manual	5.00	11/74	
92202-93001	RTE System Driver DVR23 for HP 7970 Series Digital Mag Tape Units Programming and Operating Manual	1.00	8/74	
93005-93005	Thermal Line Printer Subsystem for Driver DVR00 (RTE)	2.50	12/74	
93513-90002	RTE System Driver DVA 76-DVR40 for 2801 Quartz Thermometer System	1.50	4/75	

SOFTWARE INPUT/OUTPUT SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02100-90072	HP 2605A Console Printer Driver	$ 1.00	3/72	
02116-91760	Teleprinter Driver (LP Compatible) Manual	1.00	8/73	
02762-90002	HP 2762A Terminal Printer Driver	1.00	5/73	
02892-90003	HP 2892A Card Reader Driver	1.50	6/72	
12602-90022	Mark Sense Card Reader Drivers	1.00	6/70	
12653-90004	HP 2767 Line Printer Driver	1.00	9/70	1/73
12845-90005	HP 2610A/2614A Line Printer Driver	1.00	2/74	
12987-90006	HP 2607 Line Printer Driver	5.00	1/73	
13022-90010	HP 7970 Magnetic Tape Unit Driver	1.00	2/72	
13029-90010	Magnetic Tape Driver (7-Track)	1.00	2/72	
5950-9276	SIO Drum-Disc	1.00	2/70	
5951-1374	Software Input/Output System Configuration	1.00	7/74	
5951-1390	Subsystem Operation	2.00	10/74	

BASIC CONTROL SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02022-90014	Magnetic Tape Reformatting System Support Utilities	$ 1.50	1/74	
02100-90073	HP 2605A Console Printer Driver	1.00	3/72	6/72
02100-90129	HP 2100 Microassembler Coding Form	5.00		
02100-90140	Decimal String Arithmetic Routines	6.50	10/73	
02108-90008	Microprogramming 21MX Computers Reference Manual	5.00	8/74	10/74
02116-9017	Basic Control System Manual	8.50	12/71	
02116-91751	Prepare Tape System	2.50	8/74	
02116-91752	Magnetic Tape System	6.00	6/71	
02116-91780	2100 Series Relocatable Subroutines	11.00	12/74	
02762-90003	HP 2762A Terminal Printer Driver	1.00	5/73	
02892-90004	HP 2892A Card Reader Driver	1.50	6/72	
12602-90021	Mark Sense Drivers	1.00	6/70	
12653-90005	HP 2767 Line Printer Driver	1.00	10/70	
12845-90004	HP 2610A/2614A Line Printer Driver	1.00	6/72	
12987-90008	HP 2607 Line Printer Driver	5.00	12/73	
13023-90010	HP 7970 Magnetic Tape Unit Driver	1.00	5/74	
13026-90010	Magnetic Tape Driver (7-Track without DMA)	1.00	5/71	6/72
13027-90010	Magnetic Tape Driver (7-Track with DMA)	1.00	5/71	6/72
5951-1371	HP 2100 Front Panel Procedures	1.00	8/73	
5951-1376	Basic Binary Loader/Disc Loader, Basic Moving-Head Disc Loader	1.00	4/74	
5951-1391	Basic Control System	1.50	10/74	
5951-1392	Magnetic Tape System	1.00	7/71	

DISC OPERATING SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02100-90074	HP 2605A Console Printer Driver	$ 1.00	3/72	
02767-90007	DOS/RTE 2767 Line Printer Driver	1.00	12/70	
12560-90023	DOS RTE and BCS Calcomp Plotter Drivers	1.50	10/75	
12587-90011	HP 12587B Asynchronous Data Set Interface Driver Reference Manual	5.00	5/74	
12602-90023	DOS/RTE Mark Sense Drivers Kit 12602B	1.00	8/70	
12908-90004	HP 12908 Writable Control Store Driver	1.00	2/75	
12920-90004	HP 12920B Asynchronous Multiplexer Interface Driver Reference Manual	5.00	5/74	

DISC OPERATING SYSTEM MANUALS (Continued)

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
24307-90006	DOS-III Reference Manual	20.00	2/75	6/75
24307-90012	DOS-III Data Communications Drivers	7.50	11/74	1/75
24307-90018	DOS-III Pocket Guide	3.50	4/75	
24307-90022	DOS-III Terminal Printer Driver	1.00	1/75	
24307-90073	DOS-III Standard Drivers	6.00	1/75	
24376-90001	IMAGE/2000 Data Base Management System Reference Manual	11.00	11/74	
5951-1366	Cross Reference Table Generator	1.00	8/74	
5951-1381	DOS-M/2000C Timeshared BASIC File Handler	1.00	5/71	
5951-1394	2000C File Interface for DOS-M	1.00	6/71	

LANGUAGE MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02116-9014	HP Assembler Manual	$ 6.50	8/75	
02116-9015	HP FORTRAN Manual	5.00	3/74	
02116-9016	Symbolic Editor	4.50	2/74	
02116-9072	ALGOL Reference Manual	10.00	10/74	
12907-90010	Implementing the HP 2100 Fast FORTRAN Processor	5.00	11/74	
24307-90014	DOS III Assembler Reference Manual	8.00	7/74	
92060-90005	RTE Assembler Reference Manual	7.00	5/75	
5951-1321	HP FORTRAN IV Reference Manual	6.00	8/75	

training schedule

The schedule for customer training courses on Data Systems Division products is outlined below. Included here are courses for the 4 month period January through April, 1976.

DATA SYSTEMS DIVISION CUSTOMERS COURSE SCHEDULE
Nov. 1975 – Apr. 1976

Training Course Dates and Center Location

COURSE NUMBER	COURSE TITLE	LENGTH	PRICE	DATA SYSTEMS CUPERTINO	EASTERN TRAINING CENTER – ROCKVILLE
22940A	2100 MAINT.	10 days	$1000	1/19/76 3/22/76	1/12/76 3/1/76
22941A	21MX MAINT.	5 days	$ 500	1/26/76 3/8/76 4/5/76	2/9/76 4/5/76
22942A	7900 MAINT.	5 days	$ 500	2/2/76 3/15/76 4/5/76	
22950A	2100 SER. ASSM.	5 days	$ 500	1/12/76 2/9/76 3/8/76 3/29/76 4/26/76	1/5/76 1/19/76 2/2/76 3/15/76 3/29/76

COURSE NUMBER	COURSE TITLE	LENGTH	PRICE	DATA SYSTEMS CUPERTINO	EASTERN TRAINING CENTER – ROCKVILLE
22952A	DOS-III B	5 days	$ 500	3/1/76	
22953A	2100 TCS/IMAGE	5 days	$ 500	3/8/76	
22960A	21MX MIC. PROG.	5 days	$ 500	1/19/76 2/23/76 4/5/76	
22965	REAL TIME MEASUREMENT AND CONTROL [Course includes: RTE Oper. Sys., 5 days, B.S. Monitor, 2 days, and Meas. and Cont., 3 days.]	10 days	$1000	1/5/76 1/19/76 2/2/76 2/23/76 3/15/76 3/29/76 4/19/76	1/5/76 1/26/76 3/1/76 3/22/76 4/19/76
22969A	DISTB. SYS.	5 days	$ 500	2/2/76 4/12/76	3/15/76

Registration

Requests for enrollment in any of the above courses should be made through your local HP representative. He will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the Training Course, time of class, location and accommodations reserved.

Accommodations

Students provide their own transportation, meals and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time of registration.

In the event you are unable to attend a class for which you are registered please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than two weeks before the class begins.

Cancellations

In the event you are unable to attend a class for which you are registered, please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than two weeks before the class begins.

Eastern Training Center

Hewlett-Packard
4 Choke Cherry Road
Rockville, Maryland 20850
(301) 948-6370

Data Systems Division Training Center

Hewlett-Packard
11000 Wolfe Road
Cupertino, California 95014
(408) 257-7000

subscription information

Annual subscriptions consisting of 8 issues are available as outlined below.

I. CUSTOMERS WITH SOFTWARE MAINTENANCE AGREEMENTS OR SOFTWARE SUBSCRIPTION SERVICE AGREEMENTS (SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS)

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract. A proration table appears on the ORDER FORM which is bound into this issue.

To receive a BASE SUBSCRIPTION at no charge as well as to purchase ADDITIONAL SUBSCRIPTIONS under the provisions of the Software Service Contract Program, complete the ORDER FORM and forward it to your local HP Sales and Service Office. Your local Customer Engineer will validate your order and mail it to the appropriate HP department.

Rates:	U.S.A.	NON-U.S.A.
BASE SUBSCRIPTION	NAC*	NAC*
ADDITIONAL SUBSCRIPTIONS (ea.)	$12/yr.	**

1) ADDITIONAL SUBSCRIPTIONS must go to the same address as the BASE SUBSCRIPTION to qualify for the reduced rates.

2) ADDITIONAL SUBSCRIPTIONS ordered at a later date than the BASE SUBSCRIPTION must include, with the order form, a copy of the address label for proper identification.

3) Charges for ADDITIONAL SUBSCRIPTIONS will be prorated to expire with your Software Service Contract.

4) Orders for ADDITIONAL SUBSCRIPTIONS from a customer with a Software Service Contract will be verified by the Customer Engineer who will complete the "FOR HP USE ONLY" portion of the subscription form and direct the order to the appropriate HP department. The customer will be billed by his local HP Customer Engineering Department.

*No Additional Charge (NAC)

**Contact your local HP Customer Engineer for the price in the currency of your country.

II. CUSTOMERS WITHOUT SOFTWARE MAINTENANCE AGREEMENTS OR SOFTWARE SUBSCRIPTION SERVICE AGREEMENTS (MAIL ORDER SUBSCRIPTIONS)

Rates:	U.S.A.	NON-U.S.A.
BASE SUBSCRIPTION	$48/yr.	***
ADDITIONAL SUBSCRIPTIONS (ea.)	$12/yr.	***

1) ADDITIONAL SUBSCRIPTIONS must be ordered at the same time as the BASE SUBSCRIPTION and go to the same address as the BASE SUBSCRIPTION to qualify for the reduced rate.

2) The customer is to include payment (check, bank draft, money order, etc.) with the order. This is a Direct Mail Order procedure; please do not send a purchase order to HP.

3) Complete the ORDER FORM as directed and mail together with your payment to:

Hewlett-Packard Co.
Mail Order Dept.
P. O. Drawer # 20
Mountain View, California 94043
U.S.A.

SUBSCRIPTION CORRESPONDENCE

Address all correspondence relating to **COMMUNICATOR** subscriptions to:

Subscription Service Manager
Hewlett-Packard Company
Corporate Parts Center
333 Logue Avenue
Mountain View, California 94043
U.S.A.

***The international customer is encouraged to also use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the order procedure outlines above. If the currency regulations in the customer's country disallow the purchase of bank drafts in American dollars, or if the customer does not have ready access to the required banking services, the customer may order subscriptions from the local HP Sales and Service Office through his Customer Engineer. The customer should contact his HP Office for the price of the subscription in the currency of his country then complete the ORDER FORM and forward it together with payment to his local HP Office.

HEWLETT hp PACKARD

CORPORATE PARTS CENTER

Direct Mail
Parts and Supplies Order Form

SHIP TO:

NAME _____

COMPANY _____ CUSTOMER REFERENCE # _____

STREET _____ TAXABLE*? _____

CITY _____ STATE _____ ZIP CODE _____

Item No.	Check Digit	Part No.	Qty.	Description	List Price Each		Extended Total	

Special Instructions

Sub-total		
Your State & Local Sales Taxes*		
Handling Charge	1	50
TOTAL		

*Tax is verified by computer according to your ZIP CODE. If no sales tax is added, your state exemption number must be provided: #_____. If not, your order may have to be returned.

Check or Money Order, made payable to Hewlett-Packard Company, must accompany order.

When completed, please mail this form with payment to:

HEWLETT-PACKARD COMPANY
Mail Order Department
P.O. Drawer #20
Mountain View, CA 94043

Phone: (415) 968-9200

Most orders are shipped within 24 hours of receipt. Shipments to California, Oregon and Washington will be made via UPS. Other shipments will be sent Air Parcel Post, with the exception that shipments over 25 pounds will be made via truck. No Direct Mail Order can be shipped outside the U.S.

Although every effort is made to insure the accuracy of the data presented in the **Communicator**, Hewlett-Packard cannot assume liability for the information contained herein.

Prices quoted apply only in U.S.A. If outside the U.S., contact your local sales and service office for prices in your country.

direct mail order form

To expedite your order for software, manuals, or other materials described in this publication, use this form to order directly from the Corporate Parts Center in Mountain View, California.

1. Enter your name, address, customer reference number, and tax exemption information.
2. List the item or items you want by part number and description.
3. Compute the amount due and enclose a check or money order payable to Hewlett-Packard.

If you need assistance in placing your order, contact your local HP Sales Office.

Note: Update packages and manual supplements cannot be ordered from the Corporate Parts Center. Direct your request to:

Manual Distribution Center
Hewlett-Packard
11000 Wolfe Road
Cupertino, Ca. 95014

Computer Systems **Communicator**
Published by:
HP General Systems Division
5303 Stevens Creek Blvd.
Santa Clara, CA. 95050

January 1976

Address Correction Requested