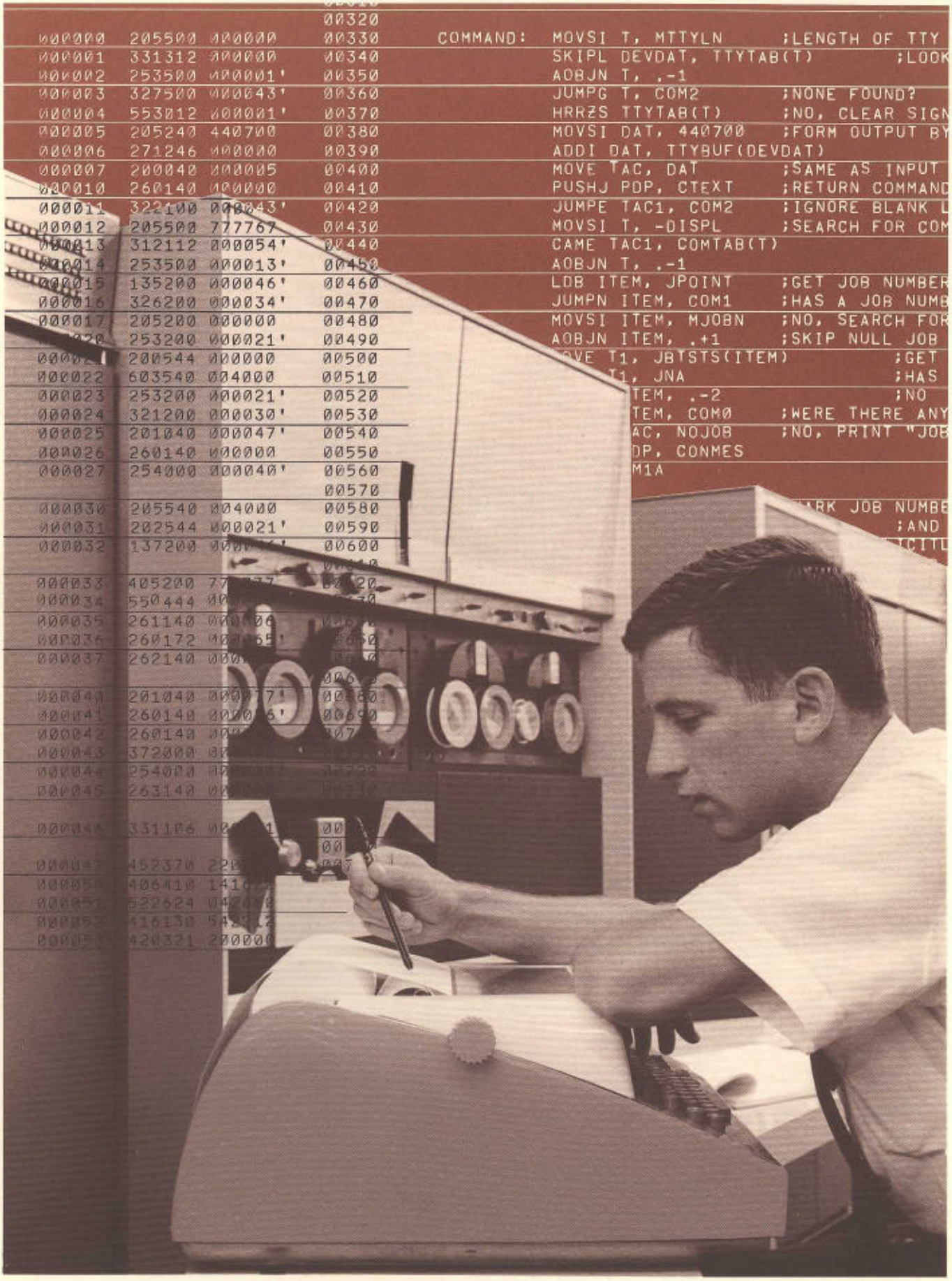
# PDP-6 TIME SHARING SOFTWARE




DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

# COMPREHENSIVE TIME-SHARING...

**The Programmed Data Processor-6 is an integrated hardware-software system which is capable of significantly lowering computation costs and improving program turn-around time. Efficient use of the central processor and peripheral equipment is made possible by concurrent operation of several user programs. The PDP-6 Monitor is a comprehensive control program which: 1) simplifies the use of the PDP-6's extensive asynchronous input-output capabilities, 2) maximizes the use made of the high-speed arithmetic processor, and 3) provides dynamic run control features necessary to allow concurrent use of the system by multiple users. This gives the user on-line interaction capabilities as well as efficient job-shop computation. These features lead to higher through-put rates at lower cost than is possible with conventional serially operated monitor systems.**

**The following sections are intended to familiarize the reader with the use of the Monitor and the salient features of the software system. Appropriate references at the end of each section should be used for a more detailed study of the PDP-6 time sharing system.**

### CONSOLE USE OF COMMON USER SERVICE PROGRAMS

Common user service programs are system library programs that perform file manipulation, editing, special desk calculating, and other functions of general interest. By adding programs to the system library, new facilities are immediately made available to all users.

### CONSOLE USE OF THE SYSTEM FOR PROGRAMMING

A user may create and edit a program (or text), translate the program to machine language (using FORTRAN, MACRO, or some other translator), load and run the program, and, if necessary, directly communicate with the program through the DDT debugging language.

### CONSOLE USE OF THE COMMON USER SERVICE PROGRAMS FOR DATA CONVERSION

The peripheral interchange programs accomplish general data conversion from medium to medium, e.g., card to tape, tape to printer, etc.

### UNATTENDED STACK OR BATCH PROCESSING OF JOBS

A job stack runs as though it were console controlled. Jobs are stacked in a card reader or other input device and processed in sequence as time-shared jobs.

### SPECIAL PURPOSE CONSOLE SERVICE

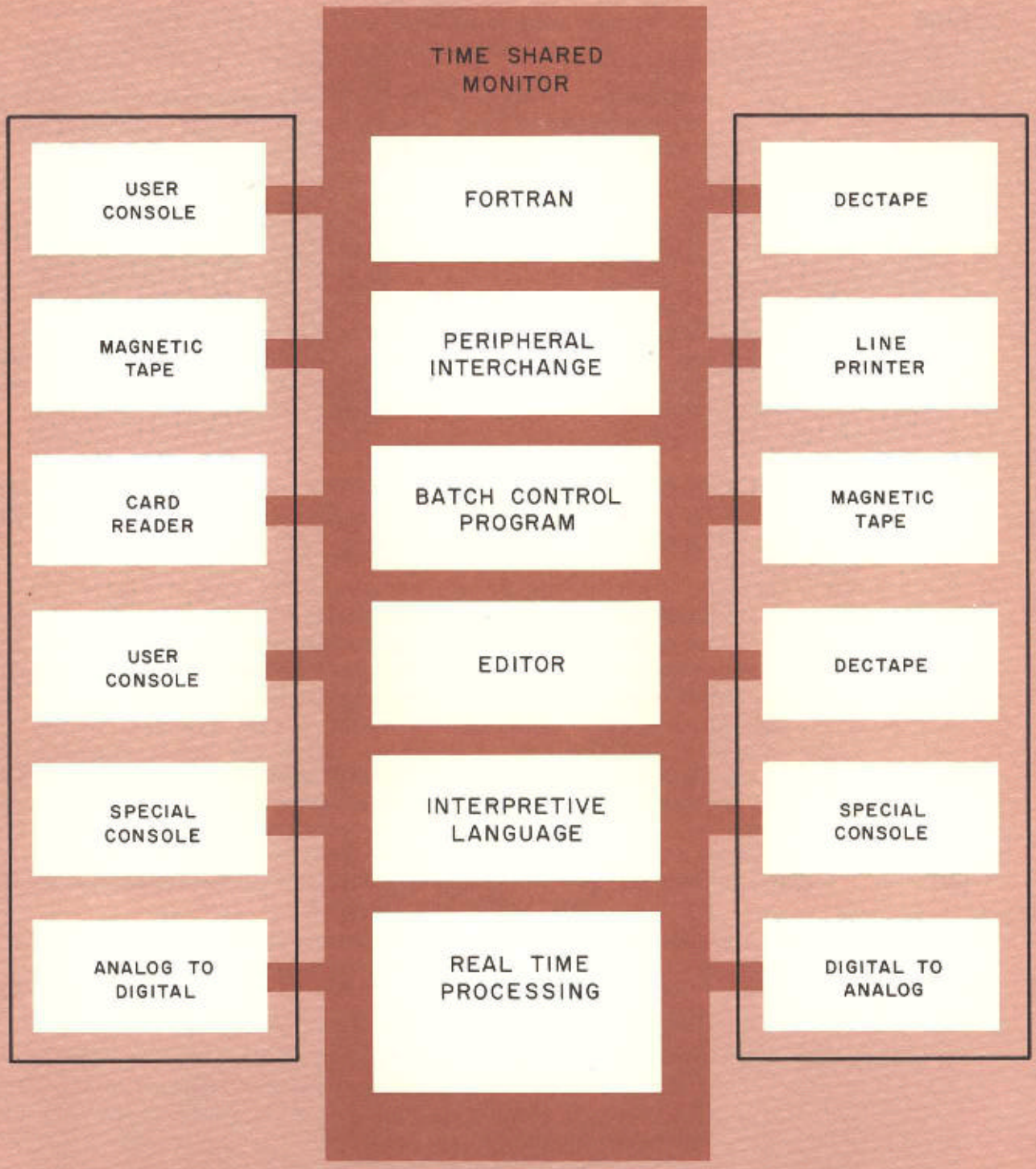
Monitor subroutines may be modified so that consoles requiring special monitor service may be included.

### REAL TIME PROCESS SERVICE

Input/output routines to connect a special device with the programming system are easily added. A job may issue system input/output commands for a special device in the same manner as for conventional devices.

***The first delivered***

# .....NOW

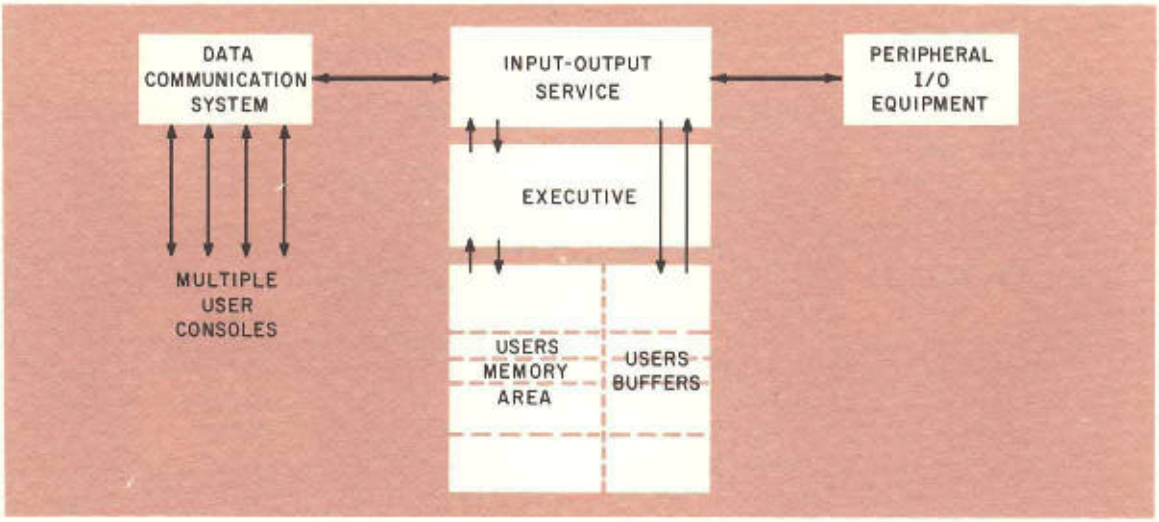



***software system written specifically for time-sharing.***

| | |
|---|---|
| IJOB | Initialize the job to which the console is attached. The job will have no core or devices assigned to it. |
| PJOB | Print the job number to which the console is currently attached. |
| KJOB | Kills the job and returns to the system all resources assigned to it. This command should be typed whenever a user is through with a job in order to allow someone else to use the job number. |
| CORE N | Sets the total number of 1024-word (decimal) blocks of core to N (decimal) for the job to which the console is currently attached. |
| GET DEVICE:FILE | Gets a previously saved program from device "DEVICE" and places it in core. This is the mechanism for loading system programs such as the assembler. It is also useful for loading user programs which have been previously saved by the "SAVE" command. The program is not started. |
| SAVE DEVICE:FILE | Saves the core image on device "DEVICE". All devices are released, and the program counter is set to the program's starting address before the file is written. |
| START LOC | Starts execution of the program at relative octal location "LOC" in the job area or at the program starting address if "LOC" is not specified. No check is made to see if a program has been loaded. All succeeding input will be directed to the user's program. |
| CONT | Continues execution from wherever the program was stopped by <control> C. All succeeding input will be directed to the user's program. |
| DDT | Starts execution of user DDT in the job area. All succeeding input will be directed to "DDT" and the user's program. |
| ASSIGN DEV:NAME | Assigns physical device DEV to the job. No other job may use a device once it has been assigned.<br><br>If DEV is the first three characters of one of the multiple devices (TTY, DTA, MTA) the Monitor will search for a free device. In any case, the Monitor will indicate the device assigned. |
| <CONTROL>C | Prints ↑C and returns control to Monitor command mode. |
| <RUBOUT> | Prints  and deletes the last character typed. |

## TIME-SHARING MONITOR SYSTEM

*Schedules multiple-user time-sharing of the system*
*Allocates facilities to particular users*
*Accepts input from and directs output to all system I/O devices*
*Relocates and protects user programs in available memory*




The Monitor system is a collection of programs remaining permanently in memory to provide overall coordination and control of the total operating system. It performs several functions. First, it permits several users' programs to be loaded into core memory simultaneously. The Monitor makes use of the PDP-6 time-sharing hardware to prevent one user's program from interfering with other users' programs. Each program is run for a certain length of time; then the Monitor switches control to another program in a rotating sequence. Switching is frequent enough so that all programs appear to run simultaneously.

Another function of the time-sharing Monitor is to process input/output commands. Only one user at a time is permitted to operate each particular device. The input/output service routines preprocess data so that all devices appear identical to the user's program, thus simplifying coding. The Monitor makes use of the PDP-6 program interrupt system to overlap input/output operations with computation. If a user's program must wait for completion of an input or output operation, the Monitor automatically switches to another user's program. A program may be terminated temporarily by user intervention, or it may suspend its own operation. Temporary termination does not remove the program from memory. A program may be dumped on backing storage and discontinued under user control.

The facing page illustrates some of the commands defined in the Monitor system. Their specific use is illustrated for each of the common user service programs in the sections following:

**SIZE:** Permanently loaded into the first 5 x 1024 memory locations.

**INPUT:** Input service routines are provided for:

| Device | System Mnemonic |
|---|---|
| card reader | CDR: |
| paper tape reader | PTR: |
| magnetic tape x | MTA x: |
| DECtape x | DTA x: |
| user Teletype | TTY: |

**OUTPUT:** Output service routines are provided for:

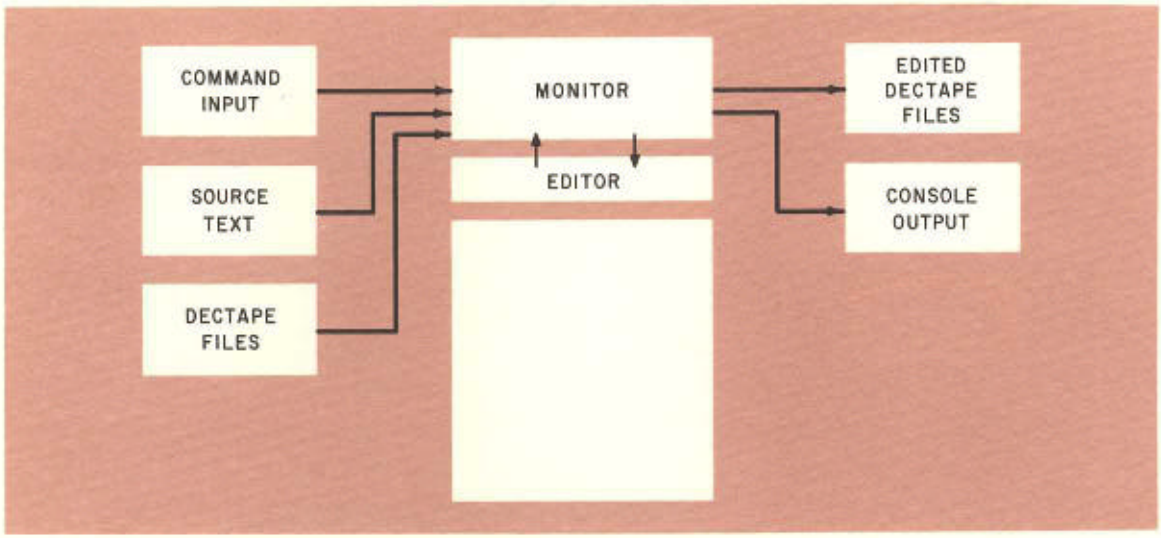
| Device | System Mnemonic |
|---|---|
| magnetic tape x | MTA x: |
| DECtape x | DTA x: |
| punched paper tape | PTP: |
| line printer | LPT: |
| user Teletype | TTY: |

**REFERENCE:** Multiprogramming System Manual (DEC-6-0-EX-SYS-UM-1P- PRE00).

```
IJOB                                    Initialize job

ASSIGN DTA                              Assign a DECtape
DEVICE DTA5 ASSIGNED

CORE 1                                  Assign 1,024 words of core

GET DTA0:EDITOR                         Load the editor
JOB SETUP

START                                   Start the job

S5,MATRIX                               Create a file called MATRIX on DECtape 5.

I10,10                                  Initialize sequencing at 10 and increment by 10.
00010     TITLE MATRIX
00020     X=A+B*C)
00030     DO 1 I=1,8
00040     X=(A+B)*C
.
.
.
.
00250     END                           Leave incrementing mode.
00260
I11                                     Insert line 11
00011     DIMENSION A(8)
D30                                     Delete line 30

P10,260                                 Print lines 10-250
00010     TITLE MATRIX
00011     DIMENSION A(8)
00020     DO 1 I=1,8
00040     X=(A+B)*C
.
.
.
.
00250     END
E                                       End file

↑C                                      Transfer control to monitor

KJOB                                    End the job
```

## DECTAPE EDITOR

*Provides a convenient means to prepare and edit any form of text*
*Eliminates the need for preparing punched cards off-line*
*Programs and text are kept on convenient DECtape reels*
*Takes virtually no processor time away from other programs*




The DECtape Editor provides a means of creating, adding to, or deleting from sequence-numbered lines in files on DECtape. This text may be input for the FORTRAN compiler, the MACRO-6 assembler, or simply a convenient means of handling textual information.

The Editor provides means for selecting a tape unit, clearing the directory, adding a new file to the directory, or selecting a file currently in the directory.

When a file has been selected, the user may resequence it, print a line or many lines, enter new lines, delete existing lines, or replace existing lines.

A few of the specific Editor commands are:

| | |
|---|---|
| Sx, name | Allow the user to access file "name" on DECtape unit "x" |
| In | Insert a line at n |
| Dn | Delete a line at n |
| Pn | Print line n |
| E | End the current file |

The procedure for composing and editing a FORTRAN II program is illustrated on the facing page.

**SIZE:** 1 x 1024 memory locations
**INPUT:** Source text from user console
**OUTPUT:** Symbolically named files on DECtape

**REFERENCE:** Multiprogramming System Manual (DEC-6-0-EX-SYS-UM-1P-PRE00)
Chapter VII, The Editor

```
IJOB
CORE 22
GET DTA0:F2
JOB SETUP
START
DTA4:INVERS,LPT:←DTA5:MATRIX(T)
PROGRAM BREAK IS 00051
NO ASSEMBLY ERROR(S)
NO SOURCE ERROR(S)
EXIT
KJOB
```

```
                                ;TITLE DEMO


                                ;X=0
                                     EXTERNAL FORSE.
                                %%%:

                                %:
000000 015000000000                  RESET.
000001 403140000000 L                SETZB 3,X#

                                ;DO 1 I=1,10
000002 201140000001                  MOVEI 3,1
000003 202140000000 L                MOVEM 3,I#
                                %.1000:

                                ;TYPE 10,SINF(X)
000004 201140000000 L                MOVEI 3,%10
000005 017140100004                  OUT. 3,32772
                                     EXTERNAL SIN
000006 200740000001'L                MOVE 15,X#
000007 260040000000 L                PUSHJ 1,SIN
000010 200140000017                  MOVE 3,15
000011 020000000003                  DATA. 3
000012 021000000000                  FIN.

                                ;1:X=X+.1

                                %1:
000013 200140000000 L                MOVE 3,[0.1]
000014 147140000006'L                FADRB 3,X#
000015 350140000003'L                AOS 3,I#
000016 307140000012                  CAGI 3,10
000017 254000000004'                 JRST %.1000

                                ;10:FORMAT(F15.7)

                                %10:
000020 254000000000 L                JRST %.1001
000021 242146132534                  ASCII ?(F15.7)?
000022 335220000000
                                %.1001:

                                ;END

                                %%%%:
000023 040000000000 L                CALL [SIXBIT "EXIT"]
                                     END %
    LITERALS

000024 457051640000
000025 175631463146

PROGRAM BREAK IS 000030
NO  ASSEMBLY ERROR(S)
NO SOURCE ERROR(S)
```

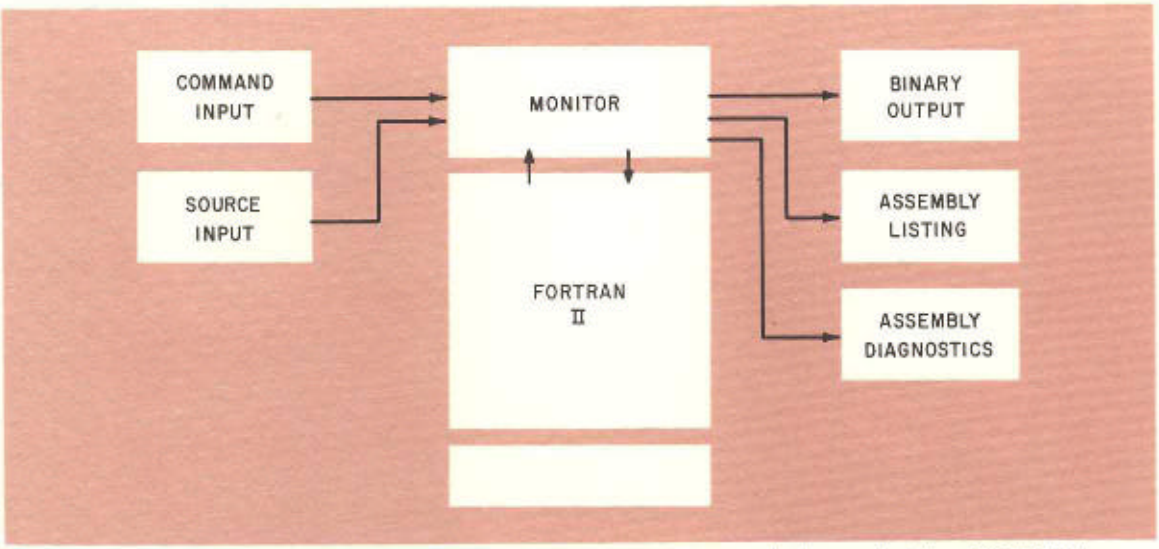
## FORTRAN II COMPILER

*One pass, syntax-directed compiler*

*Accepts input from any system input device including user console*

*Extended FORTRAN II language features include: Boolean operations, shifting capability, general format I/O conversions, N-dimensional arrays, multiple-equals statements*

*Optimized object code due to powerful PDP-6 instruction set*




PDP-6 FORTRAN II operates in the time-sharing mode or independently. FORTRAN core requirements range from 10K to 22K depending on the degree of optimization of the object code desired as a result of compilation. This gives the user the choice of a smaller compiler to allow many users to time-share or a larger one to obtain a minimized object program. PDP-6 FORTRAN II language is compatible with most FORTRAN II systems. The source language provides substantial power and flexibility through a wide variety of arithmetic, control, Boolean, function (internal and external), subroutine call, and I/O statements.

Several unique features have been implemented to make FORTRAN more flexible for the time-sharing user. TYPE and ACCEPT statements offer a method whereby FORTRAN programs communicate with the user Teletype. Source formats are not limited to 80-column punched card fields, allowing FORTRAN source code to be prepared at the user Teletype console. Devices may be dynamically assigned at run time, thereby making maximum use of all peripheral equipment available in the system.

The example at the left illustrates a method of calling in a FORTRAN program. (The source FORTRAN program was previously prepared at a user's console and stored on DECtape.)

These commands take the source program from file MATRIX on DECtape 5 and compile the object code on DECtape 4 with the file name of INVERS. The listing of the compiled FORTRAN program is directed to the line printer, and is shown at the left.

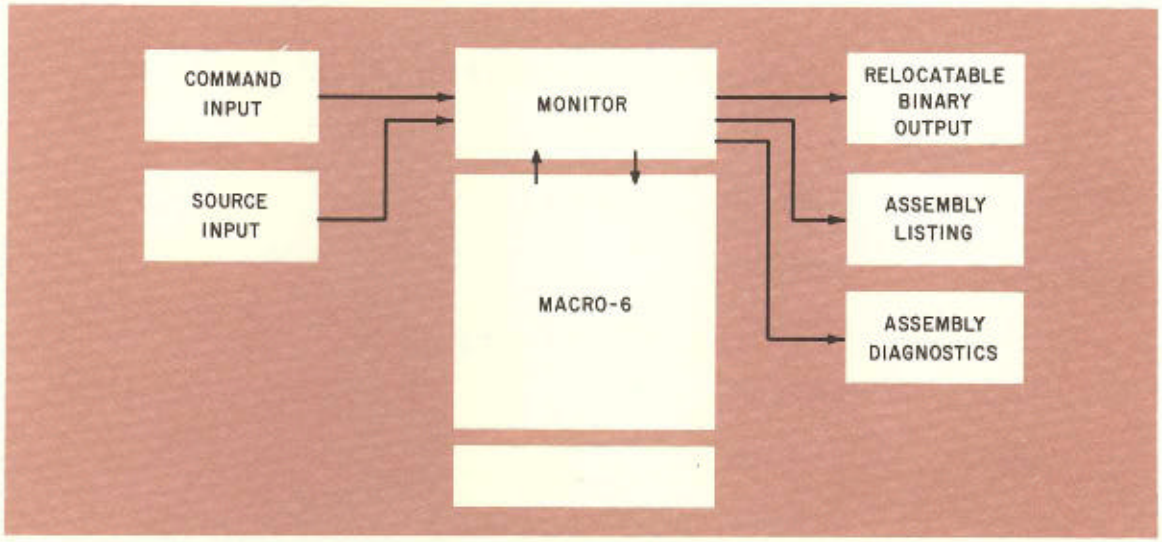
| | | | |
|---|---|---|---|
| **CORE SIZE:** | 10K to 22K | **OUTPUT:** | Relocatable binary code on DECtape, magnetic tape, or paper tape. Symbolic listing and symbol table on any system defined output device. Diagnostics on listing. |
| **COMMAND FORMAT:** | Binary file, listing ←Source file (input switch) | | |
| **INPUT:** | Any defined system input device | **REFERENCE:** | FORTRAN II Language (DEC-6-0-TP-FII-LM-FP-PRE00) |

```
IJOB

CORE 9

GET DTA0:MACRO
JOB SETUP

START

MTA4:OBJ,LPT:=DTA3:SOURCE

END OF PASS 2

↑C

KJOB
```

```
000000  212000 000017    SQR1:   MOVMS   A
000001  023740 000024'           JUMPLE  A,SQ2
000002  244740 777745            ASHC    A,-33
000003  275740 000201            SUBI    A,201
000004  241740 777777            RUT     A,-1
000005  542740 000023'           HRRM    A,SQ1
000006  242740 777735            LSH     A,-43
000007  240000 777770            ASH     B,-10
000010  132017 000177            FSC     B,177(A)
000011  202000 000031'           MOVEM   B,ST
000012  160017 000025'           FMP     B,S1(A)
000013  140017 000027'           FAD     B,S2(A)
000014  200740 000031'           MOVE    A,ST
000015  170740 000000            FUV     A,B
000016  140000 000017            FAD     B,A
000017  132000 777777            FSC     B,-1
000020  200740 000031'           MOVE    A,ST
000021  170740 000000            FUV     A,B
000022  144740 000000            FADR    A,B
000023  132740 000000    SQ1:    FSC     A,0
000024  263040 000000    SQ2:    POPJ    P,

000025  200640 000000    S1:     0.8125
000026  200450 000000            0.578125
000027  177465 777633    S2:     0.3b2734
000030  177660 000000            0.421875

000031  000000 000000    ST:     0

                                 A=17
                                 P=1
                                 B=0

                                 ENTRY   SQRT

                                 END

THERE ARE NO ERRORS

  PROGRAM BREAK IS 000032

    SYMBOL TABLE

A               000017
B               000000
P               000001
S1              000025'
S2              000027'
SQ1             000023'
SQ2             000024'
SQRT            000000' (NT
ST              000031'

END OF ASSEMBLY
```

## MACRO-6 ASSEMBLER

*Sophisticated 2-pass assembly program*
*Accepts input from and directs output to any I/O device*
*Complete MACRO facilities*
*Produces machine language compatible with Linking Loader and DDT-6*
*Address arithmetic, automatic assignment of literals, text and byte manipulation*




The primary function of MACRO-6 is to allow mnemonic instruction codes and programmer-created symbolic locations to be used in place of direct machine language. Features include address arithmetic, automatic assignment of program constants and temporary storage, and input of alphanumeric data. Macro instructions may be used as abbreviations for common sequences of code or for assembling complex word formats.

In addition, numbers may be expressed as binary, octal, decimal, or floating point. Text may be placed in a binary program by the use of the ASCII data generating statement. BYTE will cause a string of bytes to be assigned and packed into a word.

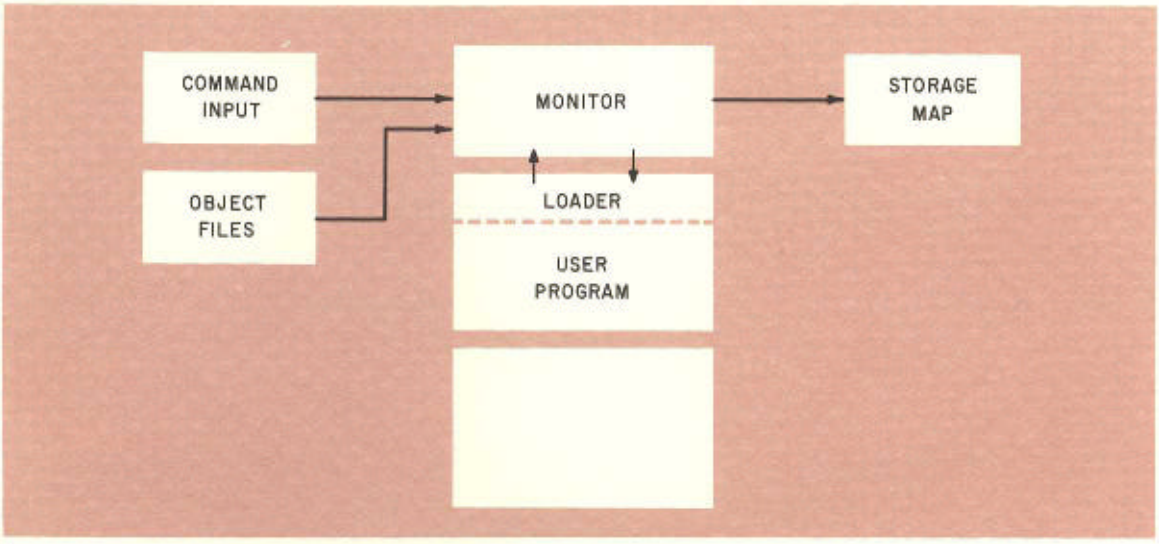
The command sequence at left illustrates an assembly from an edited file from DECtape with binary output on magnetic tape and listing on the line printer. The printout at left illustrates the format of the assembler listing and symbol table printed by the line printer.

| | | | |
|---|---|---|---|
| **SIZE:** | 9 x 1024 memory locations | **OUTPUT:** | Relocatable machine language on DECtape, magnetic tape, or punched paper tape. Symbolic listing, symbol table, and diagnostics on any output device |
| **COMMAND FORMAT:** | Binary output file, listing ←Source file 1, . . . . Source file N | | |
| **INPUT:** | Source data from any system-defined input device | **REFERENCE:** | MACRO-6 Assembly Language Manual (DEC-6-0-TP-MAC-LM-FP-ACT01) |

```
IJOB
CORE 4
GET DTA0:LOADER
JOB SETUP
START
TTY:←DTA4:(S)PG1,PG2,(WL)LB1,ARTLIB,(N)TSTDAT,(M)
STORAGE MAP
LOG         016602    000040
LOG         016602
SIN         016642    000063
COS         016645
COSD        016642
SIN         016646
SIND        016643
ATAN        016725    000061
ATAN        016725
FORSE.      017006    002031
CHINN.      017676
DEPOT.      020003
FORSE.      017006
IIB.        020012
LOADER FINISHED
START
↑C
KJOB
```

## THE LINKING LOADER

*Automatically loads and relocates programs produced by MACRO-6 or FORTRAN II*
*Produces storage map for user if desired*
*Creates symbol table in core for DDT if desired*
*Upon request will do library search*
*Storage used by loader recovered after loading*




The Linking Loader accepts programs in a form produced by MACRO-6, FORTRAN II, and other system translators, and loads them in the user's memory area to be run. Through the use of control mode characters, the loader performs several special functions. Some of the control modes available are:

| | |
|---|---|
| S | load local symbols for symbolic debugging with DDT |
| L | search files in library search mode |
| M | print a storage map |
| W | stop loading symbols |
| N | stop library search mode |

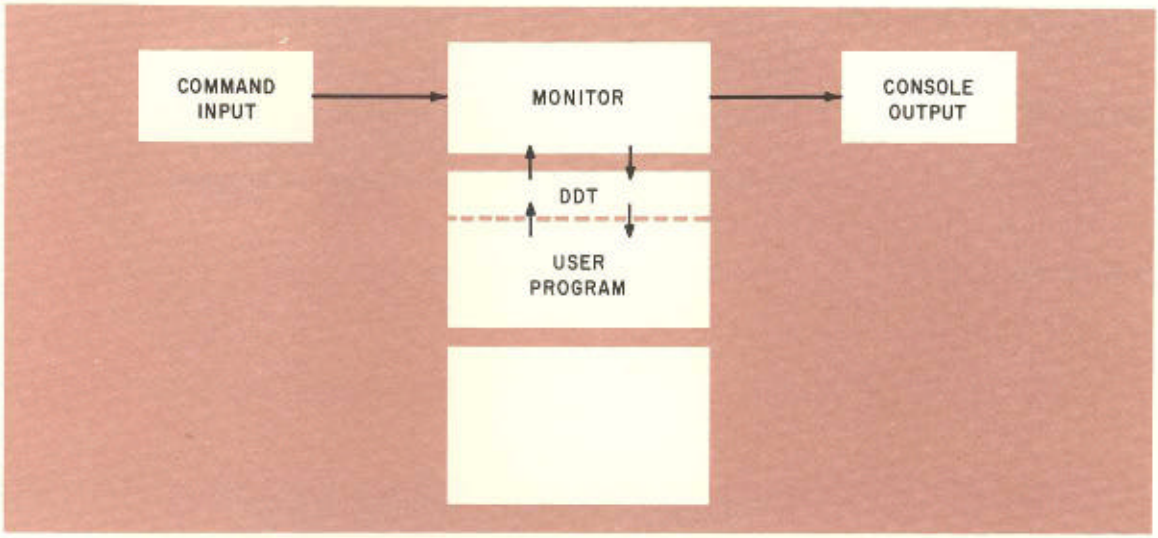
The printout on the facing page shows the commands necessary to load and run an object program. This sequence of commands loads from DTA4 files PG1 and PG2 with local symbols, LB1 and ARTLIB without locals as library files, and TSTDAT without local symbols. The user Teletype is specified for the storage map. The names with two numbers opposite are program names. The first number is the first location in the program, while the second number is the length of that program. Both numbers are octal. The names with one number opposite are global symbols and the number is the octal definition of the symbol.

| | | | |
|---|---|---|---|
| **SIZE:** | 1 x 1024 memory locations | **OUTPUT:** | Object files loaded into core memory. Storage map on the user-specified output device (if requested). |
| **COMMAND FORMAT:** | Storage map file ←(Mode) Source file 1, . . . . (Mode) Source file N. | **REFERENCE:** | Multiprogramming System Manual (DEC-6-0-EX-SYS-UM-IP-PRE00) Chapter VI, The Loader. |
| **INPUT:** | Object files from any system-defined input device. | | |

| | |
|---|---|
| `IJOB` | Initialize the Monitor. |
| `CORE 4` | Command to Monitor to assign 4K of core memory. |
| `GET DTA0:LOADER`<br>`JOB SETUP` | Command to get the Loader from the System tape (DTAO). |
| `TTY:←DTA0:DDTSYM,(I)USRDDT` | This command causes the Loader to put DDT-6 in core memory. |
| `LOADER FINISHED` | Typed by the Loader to indicate loading is complete. |
| `DDT` | Starts DDT. |
| `4000/ MOVE AC,L` | Typing a symbolic memory address followed by a forward slash causes the contents of the addressed register to be typed out on the teleprinter. |
| `X/ ADD 3,M  MOVE 1,A` | To change the contents of a particular address (e.g., register X) "open" register X by typing X/; typing the new information (here: MOVE 1, A) and a carriage return will cause the contents of X to become MOVE 1, A. |
| `5000$G` | A DDT-6 user may start his program at any address by typing the address (here: 5000) and following it by $G. |
| `5000$X;` | Use of convention shown defines the symbol X to be 5000. Location 5000 can henceforth be referred to using the symbol X. (The $ separates 5000 and X and the semicolon causes X to be defined as 5000 octal.) |
| `Y U` | If a symbol has not yet been defined, DDT will type a U (Undefined) following that symbol. The user may then define that symbol by the procedure described above. |
| `4000$B` | Typing an address (either numeric or symbolic) followed by $B inserts a breakpoint at that address. When the user's program attempts to execute the instruction at that address (4000) a program break occurs. Control returns to DDT. |
| `$P` | Typing this command restarts the program from the last breakpoint. |

## DYNAMIC DEBUGGING TECHNIQUE (DDT-6)

*Monitors the status of a running program*
*Enables user to modify program instructions or data at any point during run time*
*User may stop program at predetermined points*
*Input-Output modes include:*
*symbolic, mnemonic, floating point, numbers in any radix greater than 1*




DDT-6 is a powerful and easy-to-use on-line symbolic debugging system. DDT has a substantial turn-around time advantage over off-line debugging. Information required from the computer for debugging purposes is greatly reduced. DDT-6 can be used by programmers to pinpoint disastrous errors before the situation requires an all-of-core octal dump to recover any remaining traces of information. DDT-6 allows programmers to make symbolic changes to a program during run time. Insertion and deletion of instructions is a normal procedure for the DDT-6 user.

It is possible for several users to concurrently use DDT-6 to check out and run programs written in assembly language or FORTRAN at the symbolic assembly language level. The PDP-6 Time-Sharing Monitor provides the dynamic interaction needed to quickly debug a program without the high costs normally associated with on-line debugging with large systems.

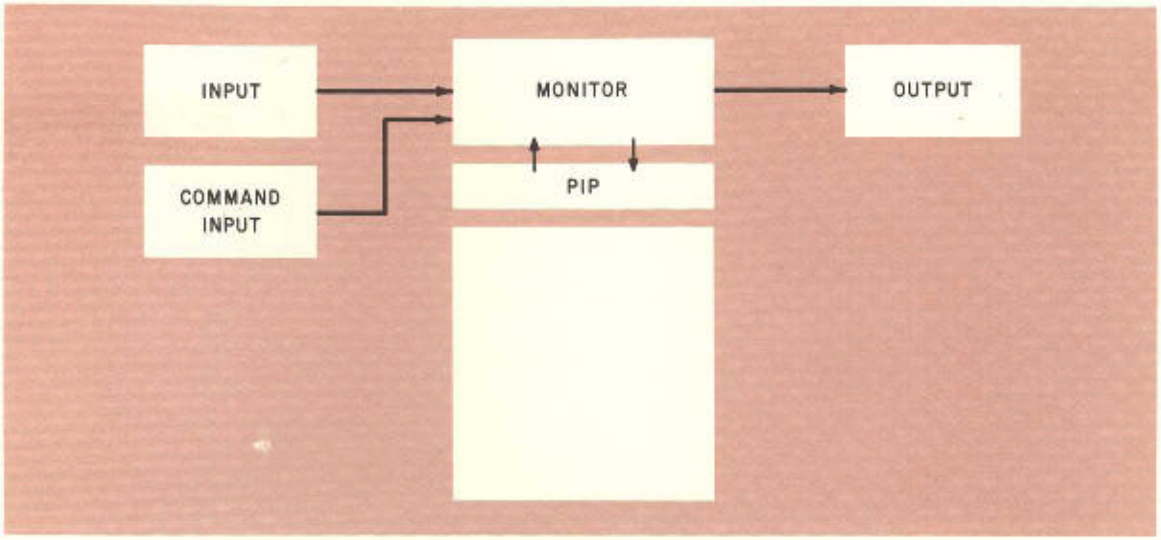
The PDP-6 Time-Sharing System allows the user to call upon DDT-6 whenever a debugging situation presents itself. More than 50 DDT-6 commands are available. Most commands are a single character to speed typing and to reduce chances of error. Several sample DDT-6 commands are shown at the left.

| | | | |
|---|---|---|---|
| **CORE SIZE:** | 3 x 1024 memory locations | **OUTPUT:** | Teletype and modified program in core |
| **INPUT:** | Control from user and/or main console | **REFERENCE:** | DDT-6 (DEC-6-0-UP-DDT-UM-FP-ACT-00) |

| | |
|---|---|
| `IJOB` | Start job |
| `CORE 1` | Assign 1024 words of core for PIP. |
| `GET DTA0:PIP`<br>`JOB SETUP` | Requests to the Monitor to load PIP from the system tape (DTAO).<br>The Monitor signal that PIP is in core. |
| `START` | Start up PIP to receive commands. |
| `LPT:←DTA1:XYZ,ABC.EXT` | Two files, XYZ and ABC.EXT on DECtape unit 1, are to be listed on the line printer by PIP. |
| `DTA2:SUBR/S←PTR:` | A file named SUBR is created on DECtape unit 2. The input data received from the paper tape reader (PTR) is transferred to DECtape as seven-bit ASCII characters with sequence numbers to be added by PIP. |
| `DTA1:(D)←ABC,TXT,XYZ` | Delete the files named ABC.TXT, XYZ. |
| `↑C` | |
| `KJOB` | |

## PERIPHERAL INTERCHANGE PROGRAM (PIP)

*Performs I/O transfers and conversions between PDP-6 peripherals*
*Eliminates the requirement for a satellite computer*
*Transfers all data formats*
*Sequences a file of information*
*Suppresses trailing spaces from cards to speed later processing*




The Peripheral Interchange Program performs any of the media conversions normally performed by an off-line peripheral computer. PIP operates concurrently with other PDP-6 jobs which are in the compilation or execution phase. Concurrent operation is made possible by the extremely flexible PDP-6 priority interrupt system.

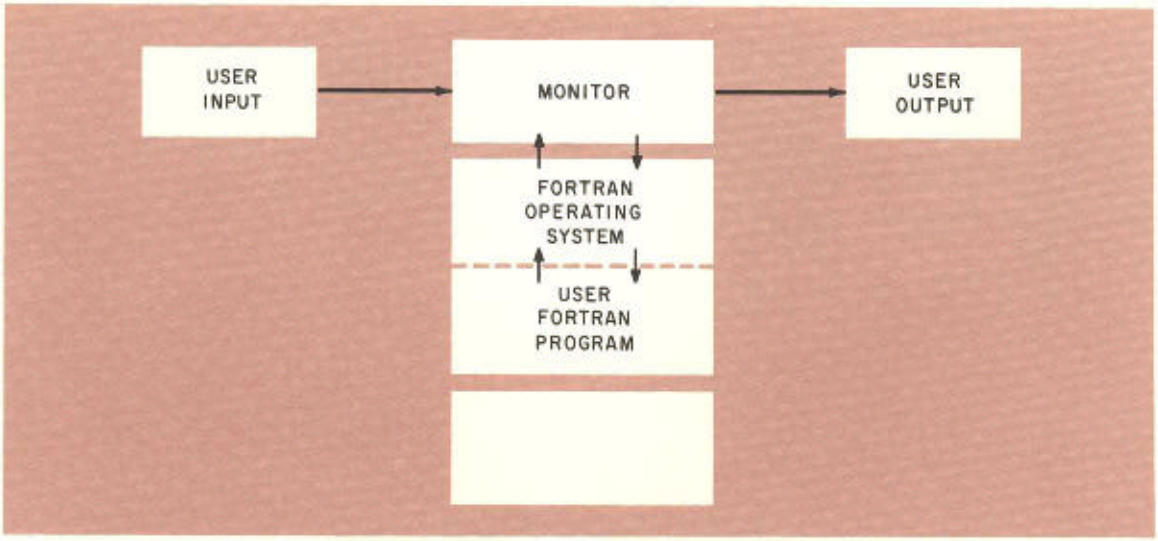
Commands for directing the information transfer operations of PIP name a source file and a destination file. The command form, from left to right, is: the output destination specification, the special character (←), and the input source specification. More than one input specification is allowed in a data transfer command. Special command qualifiers are used to specify modes of input or output. These include a binary mode for transfers of binary data, a mode for removing trailing blanks from card images, and a sequence mode for adding line sequence numbers to alphanumeric data.

An example of how the Peripheral Interchange Program is used from a console is illustrated by the printout and explanation on the facing page.

| | | | |
|---|---|---|---|
| **CORE SIZE:** | 1 x 1024 memory locations | **OUTPUT:** | Any output device the PDP-6 Monitor has available for use. |
| **COMMAND FORMAT:** | Destination file ←Source 1, . . . . Source N. | **REFERENCE:** | Multiprogramming System Manual (DEC-6-0-EX-SYS-UM-IP-PRE00) Chapter VIII, the Peripheral Interchange Program (PIP). |
| **INPUT:** | Source data from any input device the PDP-6 Monitor has available for use. | | |

## FORTRAN OPERATING SYSTEM AND LIBRARY

*Complete arithmetic package*
*Utility package including dump and chaining facilities*
*Directs output to any I/O device*
*Arithmetic functions and I/O conversion routines available to both FORTRAN and non-FORTRAN programs*
*Variable length input*
*I/O devices may be assigned at run time*




The FORTRAN operating system scans formats and directs input-output between user FORTRAN programs and the Monitor. The operating system requires 1,500 memory locations and must be loaded with a FORTRAN program for execution.

The FORTRAN library is listed in the accompany table. These functions may also be called by machine language programs.

FORTRAN Library

| Entry | Function | Size (octal) |
|---|---|---|
| EXP. 1 | Integer number to integer power | 15 |
| EXP. 2 | Floating point number to integer power | 17 |
| EXP. 3 | Floating point number to floating point power | 3 |
| ACOS | Arc cosine | 5 |
| ASIN | Arc sine | 11 |
| SQRT | Square root | 32 |
| SINH | Hyperbolic sine | 6 |
| COSH | Hyperbolic cosine | 6 |
| TANH | Hyperbolic tangent | 44 |
| EXP | Exponential | 52 |
| LOG, LOG10 | Logarithm base e, logarithm base 10 | 40 |
| SIN, COS, SIND, COSD | Sine, cosine (radians) Sine, cosine (degrees) | 63 |
| ATANZ | Arc tangent of two arguments (y, x) | 12 |
| ATAN | Arc tangent | 61 |
| FLOAT | Float a fixed point number | 6 |
| XFIX | Fix a floating point number | 5 |
| INT | Floating point truncation | 12 |
| XINT | Floating point truncation (integer result) | 11 |
| MOD | Floating point remaindering function | 12 |
| DUMP, PDUMP | Selective core dump | 250 |
| CHAIN | Chaining routine for multiple core loads | 200 |

# DIGITAL SALES AND SERVICE

## MAIN OFFICE AND PLANT

146 Main Street, Maynard, Massachusetts 01754
Telephone: From Metropolitan Boston: 646-8600
Elsewhere: AC617-897-8821
TWX: 710-347-0212 Cable: Digital Mayn. Telex: 092-027

## DIGITAL SALES OFFICES

*NORTHEAST OFFICE:*
146 Main Street, Maynard, Massachusetts 01754
Telephone: AC617-646-8600 TWX: 710-347-0212

*NEW YORK OFFICE:*
1259 Route 46, Parsippany, New Jersey 07054
Telephone: AC201-335-0710 TWX: 510-235-8319

*WASHINGTON OFFICE:*
Executive Building
7100 Baltimore Ave., College Park, Maryland 20740
Telephone: AC301-779-1100

*SOUTHEAST OFFICE:*
Suite 91, Holiday Office Center
3322 Memorial Parkway, S.W., Huntsville, Ala. 35801
Telephone AC205-881-7730 TWX: 205-533-1267

*ORLANDO OFFICE:*
1510 E. Colonial Drive, Orlando, Florida 32803
Telephone: AC305-422-4511 TWX: 305-275-0641

*PITTSBURGH OFFICE:*
300 Seco Road, Monroeville, Pennsylvania 15146
Telephone: AC412-351-0700 TWX: 710-797-3657

*CHICAGO OFFICE:*
910 North Busse Highway, Park Ridge, Illinois 60068
Telephone: AC312-825-6626 TWX: 312-823-3572

*DENVER OFFICE:*
Suite 205
5200 South Quebec Way, Englewood, Colo. 80110
Telephone: AC303-771-1180 TWX: 910-444-2212

*ANN ARBOR OFFICE:*
3853 Research Park Drive, Ann Arbor, Mich. 48104
Telephone: AC313-761-1150 TWX: 810-223-6053

*LOS ANGELES OFFICE:*
8939 Sepulveda Boulevard, Los Angeles, Calif. 90045
Telephone: AC213-670-0690 TWX: 910-328-6121

*SAN FRANCISCO OFFICE:*
2450 Hanover, Palo Alto, California 94304
Telephone: AC415-326-5640 TWX: 910-373-1266

*IN CANADA:*
Digital Equipment of Canada, Ltd.,
150 Rosamund Street, Carleton Place, Ontario, Canada
Telephone: AC613-237-0772 TWX: 610-561-1650

*IN EUROPE:*
Digital Equipment GmbH, Theresienstrasse 29
Munich 2/West Germany
Telephone: 29 94 07, 29 25 66 Telex: 841-5-24226

Digital Equipment Corporation (UK) Ltd.
11 Castle Street
Reading, Berkshire, England
Telephone: Reading 57231 Telex: 851-84327

*IN AUSTRALIA:*
Digital Equipment Australia Pty. Ltd.,
89 Berry Street
North Sydney, New South Wales, Australia
Telephone: 92-0919 Telex: 790AA-20740
Cable: Digital, Sydney

## DIGITAL SALES REPRESENTATIVES

*IN THE SOUTHWEST:*

DATRONICS INC.

7800 Westglen Drive, Houston, Texas 77042
Telephone: AC713-782-9851 TWX: 713-571-2154

DATRONICS INC.

Post Office Box 782, Kenner, Louisiana 70062
Telephone: AC504-721-1410

DATRONICS INC.

Post Office Box 13384, Fort Worth, Texas 76118
Telephone: AC817-281-1284 TWX: 817-281-3120

*IN THE NORTHWEST:*
SHOWALTER-JUDD, INC.
1806 South Bush Place, Seattle, Washington 98144
Telephone: 206-324-7911 TWX: 910-444-2212

*IN JAPAN:*
RIKEI TRADING CO.,
12, 2-Chome, Shiba Tamura-cho, Minato-ku,
Tokyo, Japan
Telephone: 591-5246 Cable Rikeigood, Tokyo
Telex: 781-TK-4208 RIKEI

*IN SWEDEN:*
TELARE AB
Industrigatan 4, Stockholm K, Sweden
Telephone: 54 33 24 Telex: 854-10178

digital

5360 Printed in U.S.A. 10-5/65