Software Reference Manual

HDOS SYSTEM

Chapter 2

General Operations

TABLE OF CONTENTS

INTRODUCTION

The purpose of the "System Set-Up" section was to familiarize you with the basics of the HDOS system. The "General Operation" and "System Optimization" sections of Chapter Two will show you how HDOS works and how to make it do your bidding. In specific, you will learn how to create and manipulate files by gaining an understanding of HDOS commands and their applications. To perform the examples in this section, you should have your WORKING DISKETTE mounted in SY∅:.

For easy reference, you will find that each command is listed in the left margin in bold print, with an explanation of its application alongside it. We recommend that you read all the way through each section before attempting to use any command. These command headings and their explanations provide only the most general information about a given command, and are intended primarily as mnemonics. A survey of some of the more useful HDOS commands and their applications is given in the "HDOS Cookbook" (Appendix C).

Just in case you have difficulty remembering what a given symbol represents, or how to recover from errors, a special HELP section (Page 2-5) immediately precedes the General Operation section. If you get halfway through the Manual and realize that you have forgotten all you were supposed to have learned, just type HELP ⊛ after the HDOS prompt ">". HDOS will list all the important commands and the format for using them.

The General Operation and System Optimization sections will concern themselves primarily with HDOS. For details on how to use BASIC and other system resources such as EDIT and ASM, refer to the appropriate chapters in the Software Manuals.

HELP

Format Symbols

DV: Device specification, where DV can be either a storage or non-storage device.

DVn: Disk drive name, where DV may be SY or DK, and n may be Ø, 1, or 2. Thus, DVn: may stand for SYØ:, SY1:, SY2:, DKØ:, DK1:, or DK2:.

FNAME   Primary file name

EXT   Extension

Device Symbols

SYØ: System drive unit zero. The diskette mounted in this drive is the system volume.

SY1:   System drive unit one.

SY2:   System drive unit two.

DKØ: Alternate drive unit zero.

DK1:   Alternate drive unit one.

DK2:   Alternate drive unit two.

TT: The system console terminal.

LP: Line printer. May be LPH14.DVD, LPH24.DVD, or LPH44.DVD renamed to LP.DVD.

AT: Alternate terminal.

ND: Null device.

Control Conventions

In addition to standard operating system commands, there are a number of conventional ASCII control codes which you can use for special purposes. The following list summarizes the HDOS control conventions:

CTRL-C   Causes HDOS or any utility program, such as PIP, to abort execution of your most recent command. Having stopped, HDOS or the utility will await further input. You can also use CTRL-C to terminate output to the line printer.

CTRL-D   Causes an exit from a utility program, such as PIP, FLAGS, and ONECOPY, back to the HDOS command mode level. When you are copying a file from TT:, CTRL-D indicates to HDOS that you have finished copying the file and want to return to the HDOS command mode.

CTRL-S   Temporarily halts output to the terminal until CTRL-Q is struck. CTRL-S is useful for examining long program listings or data files which would otherwise scroll up the screen before you could examine them.

CTRL-Q   Restarts output to the terminal following a CTRL-S.

CTRL-U   Causes HDOS to ignore the current terminal input line, allowing you to retype the entry.

CTRL-O   Discards output to the terminal until the next prompt is reached. This code is useful if you are interested in seeing only the beginning of a long list.

CTRL-P   Restarts output to the terminal when struck immediately after a CTRL-O. This code is useful if you want to see only the end of a long list.

BACKSPACE
DELETE
RUBOUT   Allow you to retype the last character. On a video terminal configured for this function, the system will backspace over the last character and remove it from the screen so you can retype it. On a teletype-style terminal, the system will repeat the last character as an indication of its deletion, so you can enter the correct character.

CTRL-Z   Cancels any ongoing HDOS activities when struck twice in succession. Also causes the data in an input file or buffer to be discarded. May be used to return to HDOS from BASIC or a utility program when all else fails.

SHIFT
RESET   When they are pressed simultaneously, these keys will return the H89 to the beginning of the Bootstrap procedure. This sequence should be used only in the event that CTRL-Z will not work, since there is some risk of data loss.

RST/∅
∅   When pressed simultaneously, these keys will return the H8 to the beginning of Bootstrap.

Common Error Messages

The following paragraphs list the error messages that you are most likely to encounter, and the methods for recovering from the errors. By far the most common cause of an error message is a typographical error. If HDOS prints an error message and you are unsure why, chances are you misspelled a word, omitted necessary spaces, or added extra spaces. In such a case, try entering the command line again. If this does not work, refer to the section of the operating instructions in which the format of the command is treated.

For a comprehensive list of error messages, refer to Appendix B (Page 2-75).

Illegal Command

HDOS does not recognize the command you have typed. You may have mistyped a legal command or entered an illegal one, such as GLUE. You may also be attempting to invoke a program, such as SET, which you have deleted from the diskette. Try entering the command again. If the command you have typed happens to be the name of a program, use the CAT or CAT/S command to make sure the program is on the disk.

Illegal Command Syntax

You have either typed a carriage return at the HDOS command mode level without entering any command, or, in the course of mistyping a command, you typed one of the special characters (such as ! @ &%* ( )+> <). Just retype the command.

?02 Illegal Format for File Name

The most common cause of this message is an extra space inserted between the device specification portion of a file name and the FNAME.EXT portion of the file name, e.g., SY1: BASIC.ABS. At any rate, the file specification was in an illegal format. You may get this message if HDOS supposes that you are referring to a file, even though you are not. For example, you will get this message if you accidentally type CAT%. In this example, HDOS assumes that CAT% is a file which you are trying in some way to manipulate. Try entering the command again. If this doesn't work, refer to the section entitled "Files" (Page 2-10).

?02 Attempted Write-Protection Violation

You have tried to rename or delete a file which has the W flag set. Make sure you really want to rename or delete the file, and if you do, use the FLAGS program to clear the W flag. Then enter the command again.

?02 The File is Already Present

You have attempted to copy or rename a file to a new file name that already exists in the volume's directory. If you are copying between disks, this message signifies that you are trying to copy a file named fileX to a disk which already contains a file whose name is fileX. HDOS is warning you that if you had continued the copying or renaming operation, a pre-existing file would have been deleted as a result of the copying or renaming process. If you are renaming a file, either select another name to which to rename the file, or rename or delete its pre-existing namesake. If you are copying, the easiest way to recover is to concoct another name for the destination file.

?02 File Cannot Be Located

HDOS cannot locate the file you have specified. The most common cause of this message is a misspelled file name (i.e., STSGEN.ABS instead of SYSGEN.ABS). You will also receive this message if you have omitted the device specification portion of the file name. In any event, try entering the command again.

?02 Volume Presently Mounted on the Device

You would receive this message if, for example, you instructed HDOS to MOUNT SY1:, but there is a disk currently mounted in SY1:. To recover in such a case, you would DISMOUNT SY1:, remove the old disk, insert the new disk, and then MOUNT SY1:.

?02 Unknown Unit for this Device

You have entered a command that deals with a disk in SY1:, SY2:, DKØ:, DK1:, or DK2:, but you did not MOUNT the disk before entering the command. MOUNT the disk and try again. You will also receive this message if you try to refer to SY1:, SY2:, DKØ:, DK1:, or DK2: without having the corresponding disk drives.

?02 The File's Contents are not Correct for This Operation

You have tried to RUN a program from the HDOS command mode level which is not written in machine language. There is no way to recover from this error, but don't worry. No alarms will sound.

GENERAL OPERATION

Files

Each file must have a unique name so that HDOS can store and find it on the diskette. HDOS will not allow two files on the same diskette to have identical names. The general format for file names is:

```
DVn:FNAME.EXT
```

where DVn: is the disk drive on which the file is stored, whether SYØ:, SY1:, or SY2:, DKØ:, DK1:, or DK2:; FNAME is the general name of the file; and .EXT is an "extension" which tells HDOS what a given file contains and whether or not it is an executable machine-code program.

DVn: (Drive Specification)

The DVn: portion of a file name tells HDOS where a given file is stored. A drive specification of SY2: indicates to HDOS that the file is stored on the disk mounted in system drive unit two; a drive specification of DKØ: indicates that the file is stored on alternate drive unit zero. Since the drive specification portion of a file name varies according to where the disk that contains the file is mounted, the same file will have a different drive specification according to whether it is mounted on SYØ:, SY1:, SY2:, DKØ:, DK1:, or DK2:.

A file name which includes a drive specification is called a fully qualified file name. The following are "fully qualified" file names:

```
SY1:BASIC.ABS
DK1:CLOCK.BAS
```

References to files which are contained on disks mounted in SY1:, SY2:, DKØ:, DK1:, and DK2: MUST be fully qualified. The drive specification may be omitted if you refer to a file which resides on the system volume (the disk mounted in SYØ:). If you do not specify a drive in the file name, HDOS will assume that you mean SYØ:.

The "directory" devices in the system, SYn: and DKn:, are the storage devices. The diskettes mounted on them contain directories which enable HDOS to locate files in order to execute your file manipulation commands. Note that SY1:, SY2:, DKØ:, DK1:, and DK2: are valid references only if you have a drive to correspond to each drive name.

The following is a list of valid drive specifications under HDOS:

SYØ:	System drive zero
SY1:	System drive one
SY2:	System drive two
DKØ:	Alternate drive zero
DK1:	Alternate drive one
DK2:	Alternate drive two

DV: (Device Specification)

The following is a list of valid device specifications under HDOS:

TT:	Console terminal
AT:	Alternate terminal
LP:	Line printer
ND:	Null device

The "non-directory" devices, TT:, AT:, LP:, and ND:, have no storage capability and maintain no directory. HDOS therefore does not directly associate files with such devices. Thus, if you specify an FNAME and .EXT after a nondirectory device specification, such as TT:OUTPUT.DOC, HDOS will print an error message, since it "knows" that OUTPUT.DOC cannot possibly be stored on TT:.

The null device is not a "real" peripheral. Any data that is output to ND: is simply discarded. Any attempt to input from ND: will result in an end-of-file condition. The null device is provided as a program debugging aid. Programs can write to ND: without using the extra storage space and access time that an output file would require.

In addition to these device names, any two-character device name followed by a colon is considered a valid device specification, provided that there is a device driver file corresponding to the device name. Thus, TI: is a legal device specification, provided that there is a device driver file on the disk called TI.DVD. Refer to the "Peripherals" section of this chapter for more information about device drivers.

FNAME.EXT   (File Name Format)

The FNAME portion of the file name is limited to eight characters, but need not consist of more than one character. The .EXT portion is limited to three characters, but can be shortened to one or two characters or omitted entirely. The characters used in FNAMEs and .EXTs may be either letters or numbers, but not special characters (e.g., !@#$%↑*( ) – _=+"':;<>, . ?/!}{ `~). The first character of the FNAME must be an alphabetical letter.

The following is a list of valid file names:

```
SY2:PRIME.BAS
SY1:LETTER.DOC
DK1:EQUAT.DAT
INCOMTAX.ASM            (HDOS assumes SYØ: when
                          no device is specified).
TT:                     The console terminal.
```

The following is a list of invalid file names:

```
SY1:TELEPHONE.LST      FNAME is too long.
COMPUTER.PROG          .EXT is too long.
SYØ:8LETTERS.DOC       FNAME cannot begin with
                       a number.
SY1:PROG#3.BAS         Only letters and numbers
                       are allowed.
DK3:PINBALL.ABS        DK3: is an illegal device
                       specification.
```

It is good practice to follow certain conventions when you assign names to your files. The primary or FNAME portion of the file name should generally correspond to the function or contents of the file. Thus, files which have similar contents or functions should have similar file names. Conventions such as this allow you to keep track of various files without keeping a separate list.

There are several conventions for the extension, and you should adhere to them as much as possible. These are:

.ABS   Absolute binary machine code
.ACM   Assembler-common subroutines
.ASM   Assembly language source programs
.BAS   BASIC programs
.DAT   Data files
.DOC   Documentation, such as instructions for
        using programs
.DVD   Device driver subroutines
.FOR   FORTRAN source programs
.REL   Relocatable programs
.SYS   Operating system programs

HDOS obeys an extension assignment convention that makes it easier to run programs that are stored in files. Programs which are written in assembly language and then translated into executable machine code should be stored on files with the .ABS extension. An example is INIT. The fully qualified name of INIT on the distribution diskette is SYØ:INIT.ABS. Yet you ran the program by simply typing INIT while in the command mode. HDOS recognizes the .ABS extension as an identifier for an executable machine-language program.

> **Command Mode**

There are a number of commands that will cause HDOS to list the contents of files, copy files, rename files, delete files, run programs, configure your system, and give status reports. In order for HDOS to execute any of these commands, the system must be booted up and in the command mode, with the ">" prompt printed at the left margin of your terminal. This > prompt will be printed alongside all of the examples in this Manual, but you will not type it in.

Most of the commands follow the same general format, but this format is flexible and varies according to what you wish to accomplish. The most general command type is:

    COMMANDΔDVn:FNAME.EXT ⊛

Most commands deal with files, although there are several exceptions and variations. If you omit the DVn: portion of the file name, then HDOS will assume that you refer to SYØ:. To reference a file on a disk mounted in a drive other than SYØ:, you must type the device name and number (e.g., DKØ:, DK1:, DK2:, SY1:, SY2:) before the file name. These are legal references only if you have more than one drive in your system.

The proper format for HDOS commands is listed on the following pages. Be sure to preserve the spacing and punctuation conventions of the format examples when you test these commands. If you do not use the correct format or "syntax", HDOS will respond to your entry with an error message. Error messages are nothing to be alarmed about. They usually indicate that you should enter the command again, this time in the proper format. Refer to Appendix B (Page 2-75) for more information about error messages.

MOUNT
DISMOUNT   **(Mounting and Dismounting Diskettes)**

The disk drive units are known as directory devices. This means that HDOS maintains a directory for the diskettes which are "mounted" on the drives. The operating system also uses a table which "maps" the location of every file on the diskette. For the sake of efficiency, parts of the directory and map are kept in memory while HDOS is running. When a diskette is removed from the system, or "dismounted," these directory and table segments must be written from memory back onto the diskette. If you add or delete files, you must dismount in order to reflect the most recent changes in the status of various files. But even if you change nothing on the diskette, the directory and table segments must be written back to the diskette from memory.

If you install a new diskette without dismounting the old one, HDOS will use the directory and tables for the old diskette, which are still in memory, to try to locate files on the new diskette. Since the directory and tables are different for each of your volumes, you must use the MOUNT and DISMOUNT commands when you insert and remove diskettes, both in order to ensure that no data is lost, and to ensure that HDOS uses the directory for volume X to search for files on volume X, rather than using the directory for volume X to locate files on volume Y.

Use the MOUNT command when you install a diskette. Only initialized diskettes can be mounted. The syntax is:

>MOUNT DVn: ⊛

For example:

>MOUNT SY1: ⊛

This command informs HDOS that there is a diskette installed in SY1:. HDOS reads the table and directory segments from the diskette into memory in preparation for your file manipulation commands. HDOS will not recognize any commands dealing with SY1: until the diskette is properly mounted.

When you are finished using the diskette mounted in SY1:, SY2:, DKØ:, or DK1:, you must use the DISMOUNT command to instruct HDOS to restore the directory information from memory to the disk. The proper command syntax is:

>DISMOUNT DVn: ⊛

Having dismounted DVn:, you can replace the dismounted diskette with another. DO NOT remove the diskette before it has been dismounted or files may be lost.

The mounting of SY∅: is automatically accomplished during Bootstrap. You cannot use the MOUNT command with SY∅:, but you can us the DISMOUNT command. Since SY∅: is the system volume, you lose HDOS when it is dismounted. For this reason, you will have to reboot the system after dismounting SY∅:. As an example:

```
>DISMOUNT SY∅:  (R)
VOLUME 010, DISMOUNTED FROM SY∅:
LABEL: WORKING DISKETTE

Install a Bootable Disk in SY∅: Hit RETURN to Reboot:
```

BYE   (Dismounting All Mounted Diskettes)

The BYE command is similar to DISMOUNT, except that BYE automatically dismounts ALL mounted devices and returns to Bootstrap. The procedure is:

```
>BYE  (R)
Volume 009, Dismounted from DK1:
Label: TAX DATA 1976
Volume 010, Dismounted from DK∅:
Label: PAYROLL AUG. 1976
Volume 202, Dismounted from SY2:
Label: TAX SCHEDULES 1976
Volume 100, Dismounted from SY1:
Label: WORKING DISKETTE
Volume 001, Dismounted from SY∅:
Label: SYSTEM VOLUME
Install a Bootable Disk in SY∅: Hit RETURN to Reboot:
```

TYPE   **(Listing the Contents of a File)**

The most basic system command allows you to type the contents of a file on the console terminal. Some files contain text in ASCII, which is meaningful when listed. Such files usually have a .BAS or .DOC extension. Other files are written in binary code, and have no meaning when listed on the console. These files have .ABS or .DVD extensions.

One file that contains meaningful information is called SYØ:SYSHELP.DOC. Type TYPE SYSHELP.DOC and a carriage return. The result of this system command is a printed list, as follows:

```
Valid System Commands:

HELP              Print This List
BYE               Dismount all Mounted Disks, and Reboot
CAT [DEV:]        List Files on Disk
COPY TO=FROM      Copy FROM file to TO
DATE [NEWDATE]    Display or Set Date
DELETE FNAME      Delete File(s)
DISMOUNT DEV:     Dismount Volume from Drive
PIP               Execute PIP
MOUNT DEV:        Mount Volume on Device
RENAME TO=FROM    Rename File FROM to TO
RUN FNAME         Run Program
SET dev:.opt      Select Option for Device
SET HELP          Documentation for SET Command
STATUS            Display Disk Statistics
TYPE FNAME        Type file contents on terminal
VER               Display the Current Version of HDOS
FNAME             Same as RUN SYØ:FNAME.ABS
```

This is a list of the commands that are legal under HDOS. You can reproduce this listing from the command mode at any time by simply typing HELP and a carriage return. HDOS automatically translates HELP to the command:

```
TYPE SYØ:SYSHELP.DOC.
```

CAT   (Cataloging Non-System Files)

The CAT command produces a directory listing of information about a file or group of files. You may use it with or without the device specifications SY∅:, SY1:, SY2:, DK∅:, or DK1:, but only one device may be specified at a time. If you do not specify a device, HDOS assumes you mean SY∅:.

Type CAT and a ⊛. The following listing will be printed:

```
Name    .Ext   Size      Date          Flags    17-NOV-80

BASIC   .ABS   42        17-NOV-80        W
    1 Files, Using 42 Sectors (136 Free)
```

This listing provides information about files on the disk mounted in SY∅: which are not essential to the system. To obtain a similar listing from a disk on another drive, type:

```
>CAT DVn:  ⊛
```

You can also list information about individual non-system files using the CAT command. The general format of this command is:

```
>CAT DVn:FNAME.EXT  ⊛
```

You may use the CAT command to print a catalogue listing of files on a configured line printer (see the "Peripherals" section of this chapter to configure your line printer).The formats for this use of the CAT command are:

```
>CAT LP:=DVn:  ⊛

>CAT LP:=DVn:FNAME.EXT  ⊛
```

Note that the command DIR is also a synonym for CAT, and works in exactly the same way in all instances.

CAT/S (Cataloging System and Non-System Files)

The CAT/S command produces a listing of all the files, both system and nonsystem, on the disk. The /S modifier informs HDOS that you wish to display files, the listing of which would normally be suppressed by the S flag.

Type CAT/S and a ⊛. The following will be printed:

```
Name      .Ext   Size    Date            Flags     17-Nov-80

HDOS      .SYS   31      17-Nov-80       SLW
HDOSOVL0  .SYS   26      17-Nov-80       SLW
HDOSOVL1  .SYS   11      17-Nov-80       SLW
SYSCMD    .SYS   12      17-Nov-80       SLW
PIP       .ABS   19      17-Nov-80       SLW
SY        .DVD   10      17-Nov-80       SL
ERRORMSG  .SYS   11      17-Nov-80       SW
SET       .ABS   12      17-Nov-80       SW
FLAGS     .ABS   4       17-Nov-80       SW
ONECOPY   .ABS   20      17-Nov-80       SW
DK        .DVD   15      17-Nov-80       SL
SYSHELP   .DOC   3       17-Nov-80       SW
HELP      .       2      17-Nov-80       SW
BASIC     .ABS   41      17-Nov-80       W
LP        .DVD   7       17-Nov-80       SW
RGT       .SYS   1       17-Nov-80       SLW
GRT       .SYS   1       17-Nov-80       SLW
DIRECT    .SYS   18      17-Nov-80       SLW

   18 Files, Using 245 Sectors (136 Free)
```

The preceding listing shows all the files on the system volume. Since no device specification was given, HDOS defaulted to SY∅:. If you have more than one drive and wish to see a catalogue of files on SY1:, SY2:, DK∅:, DK1:, or DK2:, you must specify the device in the command. The general format of the command is:

```
>CAT DVn:/S ⊛
```

For example:

```
>CAT SY1:/S ⊛
```

This directory will show all the files on the diskette mounted in SY1:.

Like the CAT command, CAT/S can be used to catalogue a specific file. /S informs HDOS that you want a catalogue listing for a file which has the S flag set. Most of the files which have the S flag set are system files. The format of the command is:

```
>CAT DVn:SYSFILE.EXT/S ⊛
```

Type CAT HDOS.SYS/S ⊛. The following will be printed:

Name	.Ext	Size	Date	Flags	17–Nov–80
HDOS	.SYS	31	17–Nov–80	SLW	

```
    1 Files, Using 31 Sectors (136 Free)
```

HDOS.SYS is a file that contains a major portion of the operating system. It consists of 31 sectors and has flags which indicate that it is a system file, that the file flags are locked and cannot be changed, and that the file is write-protected. Refer to the FLAGS section for more information about manipulating flags.

To print a catalogue listing which includes system files on a line printer, type:

```
>CAT LP:=DVn:/S ⊛
```

CAT/ALL
CAT/S/ALL   (Determining File Sector Allocation)

HDOS assigns sectors in groups, or clusters, in order to facilitate the process of extending a file (see "Theory of Operation" in Chapter 1). Thus, the number of sectors HDOS assigns to a file may or may not correspond to the number of sectors that it takes to store the data in the file. The CAT and CAT/S commands produce listings in which the size of the file is the number of sectors that it takes to store the data in the file. When appended to the CAT and CAT/S commands, the /ALL switch will produce a listing in which the size of the file reflects the actual number of sectors that have been allocated to the file. The general format for using the /ALL switch is:

       CAT DVn:/ALL ⊛

            or

       CAT DVn:/S/ALL ⊛

RUN   (Running Programs)

The format of the RUN command is:

```
>RUN DVn:FNAME.EXT ⊛
```

When you initialized your diskette(s) in the "System Set-Up Procedure," you were instructed to type "INIT." Had you desired, you could have typed:

```
>RUN SYØ:INIT.ABS ⊛
```

HDOS recognizes the contents of any file with the .ABS extension as an executable machine-code program. If you type only the FNAME portion of the file name while in the command mode, HDOS assumes that you mean "RUN SYØ:FNAME.ABS." Thus, to run BASIC, simply type:

```
>BASIC ⊛
```

If you tried the preceding example, type BYE and ⊛ and then type Y and ⊛ when BASIC prints "Are you sure?" This will exit you from BASIC to the HDOS Command Mode.

Refer to the appropriate section of your software manual for more information about system resources such as BASIC.

If you wish to run a program with the .ABS extension from a disk mounted on a drive other than SYØ:, you would type:

```
>RUN DVn:FNAME ⊛
```

```
        or
```

```
>DVn:FNAME ⊛
```

Thus, if you wanted to run BASIC from the disk in SY2:, you would enter:

```
>SY2:BASIC ⊛
```

All of the following formats are valid for running programs that have the .ABS extension:

```
>FNAME  ⊛
>FNAME.EXT  ⊛
>RUN  FNAME  ⊛
>RUN  DVn:FNAME  ⊛
>RUN  DVn:FNAME.EXT  ⊛
```

For example:

```
>FLAGS  ⊛
>FLAGS.ABS  ⊛
>RUN  FLAGS  ⊛
>RUN  SYØ:FLAGS  ⊛
>RUN  SYØ:FLAGS.ABS  ⊛
```

There is one exception to this rule, and this is SET.ABS. The SET program is so designed that you must type an argument after invoking the file. For example:

```
>SETΔSY:ΔSTEP  20  ⊛
>SET.ABSΔSY:ΔSTEP  20  ⊛
>RUNΔSETΔSY:ΔSTEP  20  ⊛
>RUNΔSYØ:SETΔSY:ΔSTEP  20  ⊛
>RUNΔSYØ:SET.ABSΔSY:ΔSTEP  20  ⊛
```

Refer to the SET section of this manual for more information about the SET command and SET options.

COPY   (Duplicating Files)

You may wish to have an extra copy of a file for the purposes of modification or safekeeping. Use the COPY command for such purposes. In general, all commands are a form of the COPY command. When you list the contents of a file, you are actually "copying" the file to the system console. When you run a program, you are actually "copying" the contents of a file into the memory of the computer. This concept will be discussed in more detail in the "Peripheral Interchange" section.

The general format for the COPY command is:

```
>COPY  DVn:DESTINAT.EXT=DVn:SOURCE.EXT  ⊛
```

The destination and source may be either a file name or a device (such as LP: or TT:), or a combination of the two. However, both destination and source cannot be a device name. You can omit the DVn: portion of both file names if the source file is on the system volume and you want the destination file stored there as well. If either of the files is not stored on the system volume, it is good practice to include the DVn: portion with both file names.

For now, you may copy one of the HDOS files by typing COPY TEMP.ABS=BASIC.ABS ⊛. The output generated will be as follows:

```
>COPY TEMP.ABS=BASIC.ABS ⊛

1 FILES COPIED
```

You have created an exact duplicate on the system volume of the file containing the program BASIC. The file is executable by means of any of the following commands:

```
>TEMP ⊛
>RUN TEMP.ABS ⊛
>TEMP.ABS ⊛
>RUN TEMP.ABS ⊛
>RUN SYØ:TEMP.ABS ⊛
```

A copy of a system file, such as HDOS, is not particularly useful. System files may be copied in a useable form only by means of the program SYSGEN, as explained in "SYSGEN" in Chapter One.

To copy a file to a peripheral, simply specify the peripheral device name in the destination portion of the COPY command. HDOS will treat the device name as if it were a file. For example, to copy a file to the terminal, type:

```
>COPY TT:=SYSHELP.DOC⊛
```

Or, to copy this file to a line printer, type:

```
>COPY LP:=DVn:SYSHELP.DOC ⊛
```

It is also possible to copy a file from TT: to disk, as is demonstrated in the following example:

```
>COPY TESTFILE.DOC=TT: ⊛
THIS IS A TEST. ⊛
 ^D (CTRL-D typed)
1 FILES COPIED
```

If you type CAT ⊛ after performing this example, the file SYØ:TESTFILE.DOC will be included in the catalog listing.

RENAME   (Renaming Files)

The RENAME command is used to change the name of any file except essential system files. System files cannot be renamed because they have the W and L flags set (for an explanation of flags, see the "FLAGS" section of this chapter). The general format for the RENAME command is:

```
>RENAME DVn:NEWNAME.EXT=DVn:OLDNAME.EXT ⊛
```

The DVn: portion of both file names must be the same, as in the following example:

```
>RENAME SY1:TAX.DAT=SY1:INCOM79.DAT ⊛
```

You may omit the DVn: portion of both file names if the file you want to rename is stored on the disk in SYØ:.

As an example, you can rename the file called TESTFILE.DOC that you created in the previous section by typing:

```
>RENAME TEST1.DOC=TESTFILE.DOC ⊛
```

DELETE   (Deleting Files)

From time to time you may decide that you have too many files in your system. You can get rid of extraneous files by using the DELETE command. Be forewarned, however, that there is no way to recover a file that has been deleted except to copy it from a "back-up" diskette, such as the distribution diskette. It is for this reason that the distribution diskette and system files are write-protected. Write-protection insures that essential system files will not be inadvertently destroyed. As a "safe" example of this command, you may delete the files that you copied in the previous sections by typing:

```
>DELETE TEST1.DOC,TEMP.ABS ⊛
```

FLAGS   (Write-Protection)

You may decide to write-protect your files to prevent them from being inadvertently deleted or modified. You can do this by means of the FLAGS program. In order to test FLAGS, you will need a file to manipulate. You can execute the following steps to create such a file:

1. Type COPY NEWFILE.DOC=TT: and a carriage return.

2. Type THIS IS A TEST and a carriage return.

3. Type CTRL-D.

4. Type TYPE NEWFILE.DOC and a carriage return.

The result of these steps should be printed output on the system console, as follows:

```
>COPY NEWFILE.DOC=TT:  ⊛
THIS IS A TEST  ⊛
^D
1 FILES COPIED
>TYPE NEWFILE.DOC  ⊛
THIS IS A TEST
>
```

You now have a file on SYØ: that can be used to test the FLAGS program.

5. Type FLAGS and a carriage return. HDOS will turn control over to the FLAGS program, which will identify itself and ask if you would like instructions.

6. Type YES and a carriage return. The following will be displayed:

```
FLAGS is used to set and/or clear the file flags. When
prompted for  the new flags, specify all the flags that are
to be set. Note that if  you set the 'L' flag, you will not
be able to clear it again. The legal  flags are:

W    Write-protect file. May not be renamed, replaced, or
     deleted.
S    Suppress normal listing or copying of file.
L    Lock the file from further flag changes.
```

The W flag, when set, prevents a file from being written to. You cannot rename, change the contents of, or delete a write-protected file unless you clear the W flag.

The S flag prevents the directory listing of a file when the CAT command or /L or /B switches are used. You can override this flag by using the CAT/S command, or the /L/S and /B/S switches (switches will be discussed in PIP, which immediately follows this section).

The L flag locks all the other flags. Once the L flag is set, you will not be able to change or clear the flags for a given file. If you also set the W flag, you will not be able to rename or delete the file.

The FLAGS program will prompt you with "File Name?" whenever it is ready for input. Type the name of the file and a carriage return to examine the flags. If you have more than one drive and you want to manipulate flags on SY1:, SY2:, DKØ:, DK1:, or DK2: you must specify the drive in the file name. Otherwise, FLAGS assumes you refer to SYØ:.

```
File Name? NEWFILE.DOC ⊛
```

At this point, the current flags will be identified and you will be prompted to enter new flags.

```
Current Flags=
 New Flags: W ⊛
```

In the preceding example, the operator set the W flag on NEWFILE.DOC.

If you simply type a carriage return, the file flags will be cleared, provided the L flag has not been set.

```
File Name? NEWFILE.DOC ⊛
Current Flags= W
 New Flags: ⊛
```

In the preceding example, the W flag was cleared, as the next example illustrates:

```
File Name? NEWFILE.DOC ⊛
Current Flags=
 New Flags: SL ⊛

File Name? NEWFILE.DOC ⊛
Current Flags= SL
 This file is locked; its flags cannot be changed.

File Name?
```

You cannot change the flags on NEWFILE.DOC any further, since you have set the L flag. However, since the W flag is not set, you can delete the file. Had you set the W flag along with the L flag, the file would be permanently recorded on your system volume, and you would have had to reinitialize the system volume in order to delete it.

Type CTRL-D to exit the FLAGS program.

PIP   (Peripheral Interchange Program)

To execute the commands explained above, HDOS uses a system program called PIP, which is an acronym for Peripheral Interchange Program. Since the file in which PIP resides has the .ABS extension, you may assume by convention that it contains an executable machine-code program. You can therefore enter PIP by simply typing PIP from the command mode. The result will be a printed prompt, as follows:

```
>PIP ⊛
:P:
```

The :P: prompt will be displayed at the left margin of the system console whenever the PIP program is awaiting input. To exit PIP, type CTRL-D.

The legal PIP commands are somewhat different from "normal" system commands. The general form is the COPY command, where a "destination" is followed by an "=", which is then followed by one or more "source" specifications:

```
:P:DVn:DESTINAT.EXT=DVn:SOURCE.EXT ⊛
```

As an example:

```
:P:SY∅:TEMP2.ABS=SY∅:BASIC.ABS ⊛

1 FILES COPIED
```

This example has the same effect as the COPY command. In this case, the destination is SY∅:BASIC.ABS and the source is SY∅:BASIC.ABS. The device specifications in this example are redundant, since if you do not specify a device, PIP assumes that you intend SY∅:.

If you do not specify a destination file, PIP will assume that you refer to TT: and will copy the contents of the file onto the terminal. Each of the following commands has exactly the same result:

```
:P:BASIC.ABS
:P:TT:=BASIC.ABS
:P:TT:=SY∅:BASIC.ABS
↑D (CTRL-D typed)
>TYPE BASIC.ABS
>COPY TT:=SY∅:BASIC.ABS
```

If you attempt any of these examples, the result will be a listing of binary "garbage". Hit CTRL–C to terminate output to the terminal. SYØ:BASIC.ABS contains a machine-code program rather than text written in ASCII.

It is possible to catalogue, rename, and delete files within PIP. These functions are accomplished by means of a "switch", which is either typed after a file name or names, typed after a disk drive name, or typed by itself in response to the :P: prompt. The legal switches are:

/L	Gives a catalogue of non-system files.
/L/S	Gives a catalogue listing of both system and non-system files.
/L/ALL	Gives the number of sectors allocated to all non-system files.
/L/S/ALL	Gives the number of sectors allocated to all system and non-system files.
/B	Gives a "brief" catalogue listing of non-system files.
/B/S	Gives a "brief" catalogue listing of all files.
/R	Used for renaming files.
/DEL	Used for deleting files.
/MOU	Used to mount a specified device.
/DIS	Used to dismount a specified device.
/RES	Used to reset a specified device.
/VER	Gives the version number of PIP.

/L (Cataloging Non-System Files)

The /L switch produces a directory listing, or catalogue, of those files that are not indispensable to the system, such as EDIT. The /L switch corresponds to the CAT command. The examples that follow are valid uses of the /L switch:

```
:P:SY1:/L ⏎
:P:SY1:EDIT.ABS/L ⏎
```

Note that if you want a catalogue of nonsystem files for a disk mounted in a drive other than SYØ:, you must specify a device name before the /L switch. If you performed the PIP "copy" example at the beginning of this section, the command SYØ:/L should produce a listing which contains the file TEMP2.ABS.

/L/S   (Cataloging System and Non-System Files)

The /L/S switch enables you to list all the files in the system, exactly like the CAT/S command. The /S in both the CAT/S command and the /L/S switch is a modifier which causes system files and files which have the S flag set to be included in the listing along with those files which you have created or copied. /S is used in PIP with /L just as it is used in the command mode with CAT. For example:

```
:P:AT:=HDOS.SYS/L/S 🔁
```

Note that a destination, AT:, was specified, so that this listing was printed on the alternate terminal instead of the console terminal.

/L/ALL
/L/S/ALL   (Determining File Sector Allocation)

Recall that HDOS assigns sectors in groups, or clusters in order to facilitate the process of extending a file. Thus, the number of sectors HDOS assigns to a file may or may not correspond to the number of sectors that it takes to store the data in the file. The /L and /L/S switches produce listings in which the size of the file reflects the total number of sectors required to store the data in the file. When appended to the /L and /L/S switches, the /ALL switch will produce a listing in which the size of the file reflects the actual number of sectors that have been allocated to the file.

/B
/B/S   (Brief Catalog Listings)

You can obtain a more abbreviated directory listing by using the /B and /B/S switches. They are used in exactly the same way as /L and /L/S, with the exception that the listings they produce omit details such as the flags and creation date of the file.

Using LP: with PIP

You can print catalogue listings produced by /L, /L/S, /B, and /B/S on a line printer by means of PIP. For example, to obtain a /L/S listing for SY∅:, type:

```
:P:LP:=/L/S  ⊛
```

Or, to obtain the same listing for a disk drive other than SY∅:, type:

```
:P:LP:=DVn:/L/S  ⊛
```

/R   (Renaming Files)

The /R switch is used in the same manner as the RENAME command. For example:

```
:P:NEWFILE.ABS=TEMP2.ABS/R  ⊛
```

The /R switch cannot be used to rename essential system files, such as HDOS. It will not work if the source file does not exist or if the destination file is already present. However, if you do try to rename an essential system file, or try to specify a nonexistent source file or a pre-existing destination file, nothing will be damaged. HDOS will simply print an error message and await further input.

/DEL   (Deleting Files)

The /DEL switch, like the DELETE command, can be dangerous if you misuse it. You can never recover a file that is deleted with this command, except by making a new copy from a back-up file. The format is similar to the format for the /R command:

```
:P:NEWFILE.ABS/DEL  ⊛
```

You may want to delete some files which have the S flag set, such as unnecessary device drivers. To do this, you will have to add the /S switch after the file name. For example:

```
:P:ATH85.DVD/S/DEL  ⊛
```

/MOU, /DIS   (Mounting and Dismounting Diskettes)

The /MOU and /DIS switches are used in the same manner as the MOUNT and
DISMOUNT commands. They allow you to change the diskettes in the drives.
You MUST specify which device you want mounted or dismounted, even if you
want to mount or dismount SY∅:. For example:

```
:P:SY1:/DIS ⊛
VOLUME 090, DISMOUNTED FROM SY1:
LABEL: BASIC DATA FILES
:P:SY1:/MOU ⊛
VOLUME 082, MOUNTED ON SY1:
LABEL: ASSEMBLY PROGRAMS
```

There is an important difference between the HDOS MOUNT and DISMOUNT
commands and the PIP switches. If you dismount SY∅: at the level of the HDOS
command mode, you cannot perform any file manipulations until after you have
rebooted the system with either the system diskette you have just dismounted, or
with another system diskette. But in PIP, you can dismount SY∅:, the system
volume, and proceed with no diskette in SY∅: without rebooting. In effect, you
are forcing PIP to be a "stand-alone utility". That is, PIP has the capability to
"stand alone", or operate without any system volume mounted. This feature is
useful if you have two drives and wish to make copies of diskettes. It allows you
to mount a diskette other than a system volume in SY∅:.

If you do plan to dismount the system volume using /DIS, you will have to LOAD any devices you wish to use before executing PIP. A sample use of the LOAD command is as follows:

```
>LOAD LP: ⊛

>LOAD DK: ⊛

>PIP ⊛

:P:SY1:/MOU ⊛

VOLUME 090, MOUNTED ON SY1:
LABEL: BASIC DATA FILES

:P:SYØ:/DIS ⊛

VOLUME 200, DISMOUNTED FROM SYØ:
LABEL: GAMES DISK

:P:LP:=SY1:/L/S ⊛
:P:TT:=DK1:/L/S
```

Note that you do not have to LOAD devices SY: and TT:. Both SY: and TT: are permanently resident in memory whenever the system is running.

If you use the SYØ:/DIS option, you will enter the boot routine when you exit PIP.

/RES   (Switching Disks)

The /RES switch will both mount and dismount a diskette. For example, if you want to change the diskette in DK1:,

```
:P:DK1:/RES ⊛

Volume 010, Dismounted from DK1:
Label: WORKING DISKETTE

Please Replace the Diskette in Drive DK1:
```

When the message "Please Replace the Diskette in Drive DVn:" is displayed, remove the diskette that is currently in the drive and install the diskette you want mounted. The /RES switch will automatically continue the mounting operation when you close the drive door.

You can also use the /RES switch to reset SYØ:. This has the same effect as using /DIS to dismount SYØ: and then /MOU to mount a new diskette on SYØ:. As with SYØ:/DIS, you are using PIP as a stand-alone program and you are therefore making HDOS inactive. Again, you must load any devices you want to use before resetting SYØ:. When you exit PIP after using SYØ:/RES, you will enter the boot routine.

/VER   (Displaying the Version Number of PIP)

The /VER switch will display the version number of PIP. You do not need to specify a device with the /VER switch, since PIP always resides on the default device, SYØ:. The version number is used by HDOS to ensure that the version of PIP that you have on your system volume is compatible with the version of HDOS that you are using. The version numbers of HDOS, PIP, and utility programs should correspond.

PIP   General Information

After you have become accustomed to PIP, you will probably find its "short-hand" notation more convenient than the command mode. To further expedite your operations with PIP shorthand, you can type PIP in the command mode, followed by a PIP switch or series of switches. Thus:

>PIP /L/S ⑳

has the same effect as

>PIP ⑳
:P:/L/S ⑳

When you type PIP at the command mode level followed by a command line, PIP will execute your command line and then return you to HDOS. You will therefore not be able to use the PIP command SY∅:/RES except within PIP. The command:

>PIP SY∅:/RES ⑳

will cause PIP to reset SY∅: and then immediately exit PIP, so that you will have to reboot the system before you can perform any file manipulations.

The various file functions of copying, renaming, and so on are not actually duplicated between PIP and the system command mode, as it may seem. When you type a command, the system first decodes it, using a program which resides in the file SY∅:SYSCMD.SYS. SYSCMD.SYS contains certain "built-in" commands, and if the command you type is one of these, such as STAT, VER, and DATE, SYSCMD.SYS executes it. Otherwise, SYSCMD.SYS checks to ascertain whether the command is a "transient" command — that is, a command which is a program residing in a file, such as SET.ABS and ONECOPY.ABS. All other transient commands, such as COPY, RENAME, etc. reside in PIP.ABS.

If the command you have typed is neither a built-in command nor a transient command, SYSCMD.SYS prints an error message, which it finds on SY∅:ERRORMSG.SYS. If your command is a transient command, then SYSCMD.SYS passes it on to PIP for execution. PIP normally resides in a file called SY∅:PIP.ABS. In order to execute any transient command, HDOS reads SY∅:PIP.ABS into your system's memory. The command is then passed on to PIP, which uses other system resources, such as device drivers, to execute the command. Thus, even though you type COPY in the command mode level of HDOS, it is PIP that actually performs the copy operation.

If you have only one or two file operations to perform, you will probably find it more convenient to use the command mode forms of the commands. For more extensive file manipulation, it will be faster to run PIP and command PIP directly.

Remember that you can exit from PIP back to the command mode by typing CTRL-D. You can also obtain a listing of PIP commands by typing HELP from the PIP mode. This causes the file SYØ:HELP to be listed on your terminal. For example:

```
:P:HELP ⊛
```

PIP COMMAND FORMAT:

```
        DEST=SOURCE1. . . . SOURCEn/SWITCH1. . . /SWITCHn
```

```
Switch              Meaning

/LIST               List Device Directory
/BRIEF              Short Listing of Device Directory
/DELETE             Delete Source Files
/DISMOUNT           Dismount Volume from Device
/MOUNT              Mount Volume on Device
/RENAME             Rename Source File(s) to Destination File
                    Name
/RESET              Dismount/Mount Volume on Device
/SHOW               Display Files with "S" Flag Set
/VERSION            Display the Current Version of PIP
/ALLOCATION         Show Number of Sectors Allocated to File(s)
```

You can print this HELP listing on your line printer by typing:

```
:P:LP:=HELP ⊛
```

ONECOPY (Copying Files Between Disks with One Disk Drive)

ONECOPY is very similar to PIP. If you have only one disk drive, ONECOPY is the only way that you can copy files from one diskette to another. ONECOPY will copy multiple files in a single operation, or you can specify individual files. To invoke the ONECOPY program, simply type ONECOPY after the > prompt. The ONECOPY program will tell you to remove all disks and then instruct you to insert the source disk and type ⊛. After you have inserted the source disk and typed ⊛, the program will display an :OC: prompt, which indicates that the ONECOPY program awaits the names of the files you want copied.

To copy single files using ONECOPY, use the format:

```
:OC:FNAME.EXT,FNAME.EXT ⊛
```

Note that you can specify only one or many file names in this format.

To copy multiple files in a single operation, use the general format:

```
:OC:*.EXT,*.EXT ⊛
```

or

```
:OC:FNAME.*,FNAME.* ⊛
```

Or, to copy all the files on a diskette:

```
:OC:*.* ⊛
```

Like PIP, ONECOPY recognizes the /VER, /L, /L/S, /B, and /B/S switches. The use of the switches is the same in ONECOPY as in PIP.

As in PIP, ONECOPY utilizes the /MOU switch, although the effect of ONECOPY /MOU differs from PIP /MOU. /MOU makes it possible to switch diskettes whenever the :OC: prompt is displayed. Thus:

```
:OC:/MOU ⏎
```

```
Insert New Disk:
```

At this point, you should insert an initialized diskette and type a carriage return. The :OC: prompt will again be displayed:

```
:OC:
```

The new diskette is now the source. You may specify any number of files to be copied from it, and you may use any of the switches to obtain file listings. If you wish to switch to yet another diskette, type /MOU again, remove the diskette, and insert another. Note that ONECOPY /MOU both dismounts and mounts a diskette, while the PIP switch /MOU serves only to mount a diskette. In this sense, ONECOPY /MOU corresponds more closely to PIP /RES than to PIP /MOU.

To exit ONECOPY, type CTRL-D. After exiting, you will enter the boot routine.

Wildcards and Multiple File Designation

MULTIPLE FILE DESIGNATION OPTION

There are two ways to manipulate more than one file with the same command. The simplest way is to use more than one file name in the source or destination fields of the commands. For example:

```
>TYPE NEWFILE.DOC,ERRORMSG.SYS ⊛
THIS IS A TEST
128 CTL-C Struck
129 CTL-B Struck
130 Data Exhausted
^C
>
```

The contents of both files are printed on the console. Typing CTRL-C terminates the output.

You can also specify multiple file names in PIP, and with the DELETE and CAT commands.

CONCATENATION

When multiple file designations are used with the COPY command, or with a copy command within PIP, the result will be a file that is a combination of the files which are specified on the right-hand side of the = symbol. An example which utilizes the files listed in the previous example follows:

```
>COPY BIGFILE.DOC=NEWFILE.DOC,ERRORMSG.SYS ⊛
```

The result of this command is a file on SYØ: called BIGFILE.DOC, which is a concatenation of NEWFILE.DOC and ERRORMSG.SYS. The following example illustrates concatenation using the PIP copy format:

```
:P:BIGFILE.TXT=NEWFILE.DOC,ERRORMSG.SYS ⊛
```

WILDCARD OPTION

* * Wildcard

The *.* wildcard is yet another way of accessing multiple files. It can be used with the command mode commands, in ONECOPY, and in PIP. The general format of the wildcard option is:

    DVn:*.EXT

       or

    DVn:FNAME.*

       or

    DVn:*.*

The effect of the wildcard option is to permit you to access all files that have the unmodified portion of the file name in common. HDOS recognizes *.DOC, for example, as a command to search for all files that match the format FNAME.DOC. The following example illustrates the use of a wildcard to obtain a directory listing:

    >PIP ⊛
    :P:*.SYS/B/S ⊛

    HDOS   .SYS   HDOSOVLO.SYS   HDOSOVL1.SYS   SYSCMD .SYS
    ERRORMSG.SYS   RGT   .SYS   GRT   .SYS   DIRECT .SYS

This is a brief listing of the essential system files. Note that all the file names have the same .SYS extension, while all the primary file names are unique.

You can substitute the * into either the FNAME or the .EXT field, and you may use * in both the FNAME and .EXT fields. Thus, *.* is a valid command which signifies that you want to manipulate all the files on the disk, since all files match the *.* reference.

The *.* command is particularly useful for copying files within ONECOPY. If you type *.* after the :OC: prompt, ONECOPY will COPY all non-system files from the system volume to another disk.

If you have a multiple-drive system, you can transfer many files between any two drives in one operation. First, mount an initialized diskette in the drive to which you want to transfer files, and then, using either COPY or PIP, specify drive names in the format:

```
DVn:*.*=DVn:*.*
```

All files will be transferred from the source drive to the destination drive.

If you are copying files between drives, the *.* wildcard enables you to transfer a given file in such a way that the copy that is transferred will have the same name as the original, as is illustrated in the following example:

```
>COPY SY1:*.*=SYØ:BASIC.ABS ⊗
```

The effect of the preceding command is to create a file on SY1: named BASIC.ABS. Of course, you could have produced the same effect by typing:

```
>COPY SY1:BASIC.ABS=SYØ:BASIC.ABS ⊗
```

The * wildcard can also be used to rename files during the copying process:

```
>COPY DK1:*.TXT=SYØ:NEWFILE.DOC ⊗
```

The effect of this example was to copy NEWFILE.DOC to a file on DK1: called NEWFILE.TXT. The next example illustrates the use of wildcards to rename multiple files in a multiple-drive copy operation:

```
>COPY SY2:*.TXT=DKØ:*.DOC ⊗
```

In this example, the operator copied all files on the diskette mounted in DKØ: that have the .DOC extension to the diskette in SY2:, and assigned the extension .TXT instead of .DOC to the new files in SY2:. The FNAMEs of the files on DKØ: were preserved during the transfer.

You can also copy files using a combination of both wildcards and multiple file designations, but you may only designate multiple source files. Thus, the following is a valid combination of wildcard and multiple file designation:

```
>COPY SY1:*.*=HELP.DOC,ERRORMSG.SYS ⊗
```

while the following is an invalid combination of wildcard and multiple file designation:

```
>COPY SY1:*.CAT,*.DOG=HELP.DOC,ERRORMSG.SYS ⊗
```

This second example is invalid because more than one file was specified in the destination field of the command.

????????.??? Wildcard

You can use another type of wildcard as a substitute for letters in a portion of a file name. This wildcard is the "?". Since the FNAME portion of a file name may be up to eight characters in length and the .EXT portion may be up to three characters, the wildcard ????????.??? is exactly the same as *.*.

If you use "?" in the FNAME portion of a file designation, you must use at least as many "?"s as there are characters in the name of the file you want to manipulate. Thus,

```
>CAT ????.ASM ⊗
```

will list those .ASM files whose FNAME field contains four characters or less.

The "?" can be used along with the "*" wildcard. For example, if you had several files on the disk in SYØ: called CHAPTER1.DOC, CHAPTER2.TXT, and CHAPTER3.DOC, the command

```
>CAT CHAPTER?.* ⊗
```

would list all three of these documents.

Note that you can use the CAT command and the /L and /S switches with multiple file designations and wildcards, but you may not refer to more than one device in the same command. The next example is invalid and will cause an error message:

```
>CAT SYSHELP.DOC,SY1:HDOS.ACM ⏎
```

This example is invalid because both SYØ: and SY1: are used in the same CAT command. The following is a valid use of CAT with a multiple file designation:

```
>CAT SYSHELP.DOC,HDOS.ACM ⏎
```

There is also a restriction on the use of the *.* wildcard with CAT. If you type CAT *.* ⏎, the system will produce a listing of all non-system files. If you want to list all files on the disk using the *.* wildcard with CAT, you will have to use the /S modifier to override the S flag on the system files, as in the following example:

```
>CAT DK2:*.*/S ⏎
```

SYSTEM OPTIMIZATION

SET   (System I/O Configuration)

The SET command is used to configure your system for the particular input and output devices that you have. For example, you can use the SET command to specify how many characters your terminal can handle on one line, to set the step rate of your disk drives, to set the baud rate of your line printer, and so on. The general form of the SET command is:

>SETΔDV:ΔOPTION ⊛

You can obtain this general command format for SET from the command mode by typing:

>SETΔHELP ⊛

Or, if you want to know the possible SET options for a given device, type:

>SETΔDV:ΔHELP ⊛

For instance, if you wanted to determine the possibilities for optimizing the configuration of your line printer, you would type:

>SETΔLP:ΔHELP ⊛

Once you have set a given option, HDOS writes it on the disk and in memory so you do not have to re-SET the option each time you reboot the system. However, your initial configuration is not indelible; so if you alter your hardware you can re-SET the option. Any changes you make by means of SET remain in effect until you reuse the command.

Like PIP, the SET program has been assigned a version number, which you can display by entering:

>SETΔVER ⊛

The tables that follow summarize the SET options, default values for the options, and the devices to which the SET command applies. Table A lists all HDOS devices. Table B lists the SET options for HDOS; Tables C through K list the options for each device. The options that are preset on the distribution diskette are marked with an asterisk. To make the most of this information, you will probably want to refer to the "Peripherals" section, which follows immediately.

Table A

HDOS DEVICES

Device Name	Description
SY:	System disk drives (primary boot)
DK:	Alternate disk drives (secondary boot)
TT:	Console terminal, input and output
LP:	Line printer
AT:	Alternate terminal

Table B

SET HDOS OPTION

Option	Description
HELP	Prints SET HDOS options.
* DATE	User prompted for date at boot-up.
NODATE	Suppress date prompt at boot-up.

NOTE: "NODATE" files created under HDOS Version 2.0 cannot be catalogued under previous versions of HDOS.

Table C

SET SY: OPTION

STEP n   Sets the speed at which any 5 1/4-inch primary boot drive steps
          between tracks on the diskette. Step time for all 5 1/4-inch SY:
          drives is set using this command. The step time for 8″ drives is
          preset. Use TEST17 to determine the value of "n"; it should be
          between 8 and 30. The seek time of the slowest drive is the fastest
          time you can use for all SY: drives your system. The distribution
          disk is set at a step rate of 30.

Table D

SET DK: OPTION

STEP n   Sets the speed at which any 5 1/4-inch secondary boot drive steps
          between tracks on the diskette. The step time for all 5 1/4-inch DK:
          drives is set using this command. The step time for 8″ drives is
          preset. Use TEST17 to determine the value of "n"; it should be
          between 8 and 30. The seek time of the slowest drive is the fastest
          time you can use for all the DK: drives your system. The distribu-
          tion disk is set at a step rate of 30.

Table E

SET TT: OPTION

Option	Description
HELP	Prints the SET options for TT:.
* NOBKS	Uses the backslash character (\) for errors.
BKS	Enables backspacing to correct typing errors.
* BKM	Causes BACKSPACE (CTRL-H) to be treated as a DELETE.
NOBKM	Lets HDOS receive the backspace character.
* MLI	Maps lower case input to upper case.
NOMLI	Allows lower case input to HDOS.
* MLO	Maps lower case output to upper case.
NOMLO	Allows lower case output from HDOS.
* NOTAB	HDOS expands TAB (CTRL-I).
TAB	Lets the terminal process TABs. (Faster)
* 2SB	Uses two stop bits. (Universal)
1SB	Uses one stop bit. (Normal)
WIDTH n	Sets the terminal screen width to n characters. 80 is the default value. HDOS starts a new line if more than n characters are sent to the terminal screen. The value of n must be between 20 and 255.
FILL c n	Sets "c" as the ASCII code for the character which needs "n" null characters inserted after it. Fill is usually needed after a carriage return (13) on some slow, hard-copy terminals. This allows time for the completion of the return motion before the next characters are printed.

With most terminals, you should SET the following options:

```
SET TT: 1SB
SET TT: FILL 13 0
```

Table F

OPTIONS FOR LPH14.DVD

Option	Description
HELP	Prints the SET options for LPH14.
* 6LPI	Sets the H14 for 6 lines per inch.
8LPI	Sets the H14 for 8 lines per inch.
PAGE n	Sets the number of lines per page to "n". If "n" is zero, lines are printed continuously. Default value is 60.
PORT n	Sets the port address for LPH14 to "n". Default value of "n" is 340Q.
WIDTH m,n	Sets the Width Control switch position. "n" is the narrow position and "m" is wide. The only legal values are 80, 96, and 132. The default setting is 80,132.
BAUD n	Sets the baud rate for LPH14. You must set only standard rates (i.e., 4800, 2400, 1200, etc.) Default is 4800.

* Preset option from the distribution disk.

Table G

OPTIONS FOR LPH24.DVD

Option	Description
HELP	Prints the SET options for LPH24.
* FORM	Automatically inserts a form feed at the end of a listing.
NOFORM	Disables automatic insertion of form feed.
PAGE n	Sets the number of lines per page to "n". If "n" is zero, lines are printed continuously. Default value is 60.
PORT n	Sets the port address for LPH24 to "n". Default value of "n" is 340Q.
WIDTH n	Sets "n" as the width of the printer. If more than "n" characters are sent, a new line is started. The range of n is 0-132. If 0 is used, the new line feature is disabled. Default value is 132.
LENGTH n	Sets "n" as the length of the form being used. This feature is useful when you are printing short form listings such as checks and labels. When this feature is used, a form feed will cause the printer to move to the top of the next item to be printed instead of advancing an entire page. The range of "n" is 4-122. The default value is 60.
BAUD n	Sets the baud rate for LPH24. You must set only standard rates (4800, 2400, 1200. . .). Default is 4800.

Table H

OPTIONS FOR LPH44.DVD

	HELP	Prints the SET options for LPH44.
*	FORM	Automatically inserts a form feed at the end of a listing.
	NOFORM	Disables automatic insertion of form feed.
*	TABX	TAB command causes software generation of the tab.
	NOTABX	TAB command causes the tab code (09) to be sent to the peripheral device.
	PAGE n	Sets the number of lines per page to "n". If "n" is zero, lines are printed continuously. Default value is 60.
	PORT n	Sets the port address to "n". Default value of "n" is 340Q.
	WIDTH n	Sets "n" as the width of the printer. If more than "n" characters are sent, a new line is started. The range of n is 0-132. If 0 is used, the new line feature is disabled. Default value is 132.
	AUTO-CR	Automatically inserts a carriage return upon receipt of a new-line code (012Q).
	NOAUTO-CR	Disables AUTO-CR feature.
	BAUD n	Sets the baud rate You must set only standard rates (4800, 2400, 1200. . .). Default is 1200.

* Preset option from the distribution disks.

Table I

OPTIONS FOR ATH44.DVD

	HELP	Prints the SET options for ATH44.
*	FORM	Automatically inserts a form feed at the end of a listing.
	NOFORM	Disables automatic insertion of form feed.
*	TABX	TAB command causes software generation of the tab.
	NOTABX	TAB command causes the tab code (09) to be sent to the peripheral device.
	PAGE n	Sets the number of lines per page to "n". If "n" is zero, lines are printed continuously. Default value is 60.
	PORT n	Sets the port address to "n". Default value of "n" is 340Q.
	WIDTH n	Sets "n" as the width of the printer. If more than "n" characters are sent, a new line is started. The range of n is 0-132. If 0 is used, the new line feature is disabled. Default value is 132.
	AUTO-CR	Automatically inserts a carriage return upon receipt of a new-line code (012Q).
	NOAUTO-CR	Disables AUTO-CR feature.
	BAUD n	Sets the baud rate You must set only standard rates (4800, 2400, 1200. . .). Default is 1200.

NOTES:

1. Alternate terminal I/O is buffered. Therefore, data sent to and from the alternate terminal will not be displayed until the buffer is full or until the operator indicates an end-of-file condition by typing CTRL-D at the input device.

2. Input entered at the alternate terminal will not be echoed at the alternate terminal.

3. BACKSPACE, DELETE, CTRL-U, and other line editing functions of the console terminal cannot be used to edit input from the alternate terminal.

Table J

OPTIONS FOR ATH84.DVD

Option	Description
HELP	Prints the SET options for ATH84.
* 1SB	Sets for one stop bit.
2SB	Sets for two stop bits.
* MLC	Sets to map lower case to upper for both input and output.
NOMLC	Allows lower case input and output.
WIDTH n	Sets the terminal width to "n" characters. Default value is 80. HDOS starts a new line if more than "n" characters are sent. The value of "n" must be between 20 and 255. The default value is 80.
PAD n	Sends "n" pad characters after a carriage return. Pad is needed on some slow hard-copy terminals. This allows time for completion of the return before the next characters are printed. Default is zero.
PORT n	Sets the port address for ATH84. Default is 320Q.
BAUD n	Sets the baud rate for ATH84 to "n". Use only standard baud rates (4800, 2400, 1200, etc.). Default is 300.

NOTES:

1. Alternate terminal I/O is buffered. Therefore, data sent to and from the alternate terminal will not be displayed until the buffer is full or until the operator indicates an end-of-file condition by typing CTRL-D at the input device.

2. Input entered at the alternate terminal will not be echoed at the alternate terminal.

3. If the alternate terminal is a hard-copy device, BACKSPACE, DELETE, CTRL-U, and other line editing functions of the console terminal cannot be used to edit input from the alternate terminal.

Table K

OPTIONS FOR ATH85.DVD

Option	Description
HELP	Prints the SET options for ATH85.
* 1SB	Sets for one stop bit.
2SB	Sets for two stop bits.
* MLC	Sets to map lower case to upper for both input and output.
NOMLC	Allows lower case input and output.
WIDTH n	Sets the terminal width to "n" characters. Default value is 80. HDOS starts a new line if more than "n" characters are sent. The value of "n" must be between 20 and 255. The default value is 80.
PAD n	Sends "n" pad characters after a carriage return. Pad is needed on some slow, hard-copy terminals. This allows time for completion of the return before the next characters are printed. Default is zero.
PORT n	Sets the port address for ATH85. Default is 374Q.

NOTES:

1. Alternate terminal I/O is buffered. Therefore, data sent to and from the alternate terminal will not be displayed until the buffer is full or until the operator indicates an end-of-file condition by typing CTRL-D at the input device.

2. Input entered from the alternate terminal will not be echoed at the alternate terminal.

3. If the alternate terminal is a hard-copy device, BACKSPACE, DELETE, CTRL-U, and other line editing functions of the console terminal cannot be used to edit input from the alternate terminal.

Peripherals

To facilitate expansion and maintenance of the system, HDOS was designed in a modular fashion, as a number of subprograms that communicate with one another. Each of these subprograms is responsible for a logically distinct task. For example, the subprogram which processes commands is separate from that which processes I/O. When the command processor needs input, it asks the I/O handler for data. Likewise, when the command processor generates output, it passes the data along to the I/O handler. To compare HDOS to a person, one might liken the command processor to the brain, the external peripherals to the sense organs, and the I/O handlers, such as device drivers, to the nerves which translate and transmit data from the sense organs to the brain.

HDOS does not directly communicate with peripherals. Rather, it communicates with peripherals indirectly, by means of device drivers. Any device that is to be interfaced to HDOS must therefore be interfaced by means of a device driver. When HDOS writes to device XX:, it merely supplies the bytes to be written, invokes the driver, and relies upon the driver to convert the data into the format required by the specific device. Thus, only device drivers "know how to talk to" peripheral devices.

Since device drivers are not inherent parts of the operating system, HDOS I/O is quite flexible. Heath will support new device configurations by writing the appropriate device driver and supplying it on the distribution diskette.

To manage the various devices in the system, HDOS maintains a table which supports up to seven device drivers. Of these seven devices, one is permanently reserved for TT:, the system console terminal. Moreover, the system requires the disk driver SY.DVD in order to control the primary boot disk drive units. This leaves room for five additional device drivers that will accommodate your specific system requirements.

Whenever the system is booted-up, HDOS scans the directory for entries of the form XX.DVD, and then uses these entries to build a device table. Thus, device driver names must be two characters long, and the file name extension must be .DVD. The two-character file name is used to define the device name. For example, the driver for your line printer should be called LP.DVD. Since the maximum number of devices exclusive of SY: and TT: that HDOS can support is five, HDOS will include only the first five device drivers in the device table and will simply discard any others. Once HDOS has entered the devices in the device table during boot-up, they remain usable as long as the current system disk is mounted in SYØ:.

The TT: device driver is permanently resident in HDOS and HDOSOVLn files. Therefore, since HDOS does not have to scan the directory in order to "know" that there is a device TT: in the system, no TT.DVD files are present.

The system distribution disk is supplied with a number of general-purpose device drivers designed to accommodate both Heath and non-Heath peripherals:

LPH14.DVD   An H14 line printer interfaced via a Multiport Serial I/O card at address 340Q (0E0H).

LPH24.DVD   An H24 line printer (TI810) interfaced via a serial I/O card at address 340Q (0E0H).

LPH44.DVD   An H44 line printer (Diablo 1640) interfaced via a serial I/O card at address 340Q (0E0H).

ATH44.DVD   A Diablo 1640 KSR Printer/Terminal configured via a serial I/O card at address 340Q (0E0H).

ATH84.DVD   An alternate terminal configured at address 320Q (0D0H) via an H8-4 card.

ATH85.DVD   An alternate terminal configured at address 374Q (0FCH) via an H8-5 card.

DK.DVD   Secondary boot device driver. May be an H47 floppy disk drive unit interfaced via an H89-47 card at address 170Q (078H) or an H17 floppy disk at address 174Q (07CH).

ND.DVD   A null device driver (used primarily for test purposes).

The following steps outline the procedure you would use to incorporate any of the Heath-supplied device drivers except DK.DVD and ND.DVD, which require no software configuration.

1.   Decide which device drivers you will need. Keep in mind that the maximum number of user-selectable drivers is currently five. For instance, assume that an H14 is to be included in your system. In such a case, you would have to select LPH14.DVD as your device driver.

2.   Configure your hardware to match the specifications of the selected device driver. (Refer to Chapter One, the "System Configuration" section of this Manual).

3.   Copy the device drivers to the appropriate two-letter file name. For example, to use LPH14:

```
RENAME LP.DVD=LPH14.DVD ⏎
```

4.   Enter a BYE command, and then reboot HDOS. It is necessary that you reboot because the HDOS device table is built only on boot-up. If you rename LPH14.DVD to LP.DVD and then try to use LP: without rebooting the system, HDOS will not recognize any commands to LP:, since there was no LP: in the directory when HDOS last built the device table.

By way of example, an illustration implementing LP: with a Diablo printer is given below:

```
ACTION? <BOOT> BOOT

SYSTEM HAS 32K OF RAM

HDOS Version 2.0
 Issue 50.06.00
Date (DD-MMM-YY)? 15-APR-81 ⏎

Volume 100, Mounted On SY0:
Label: SYSTEM VOLUME

>RENAME LP.DVD=LPH44.DVD
>BYE
Volume 100, Dismounted from SY0:
Label: SYSTEM VOLUME


Install a Bootable Disk in SY0: Hit Return to Reboot: ⏎


ACTION? <BOOT>      From this point on, HDOS will
                    recognize references to device LP:.
```

From now on, the new device will appear as part of the operating system. You may now use device LP: throughout HDOS.

The general format for accessing device LP: from the HDOS command mode is:

```
>COMMAND LP:=DVn:SOURCFIL.EXT ⊛
```

There are only three HDOS commands that you may legally use to access LP: from the HDOS command mode. These are COPY, and CAT, CAT/S. Their use is illustrated below.

```
>COPY LP:=HDOS.ACM ⊛

>CAT LP:=SYØ: ⊛

>CAT LP:=SYØ:/S ⊛
```

Since you did not fully qualify the file names in the COPY examples above, HDOS would assume that ERRORMSG.SYS and HDOS.ACM were located on the disk in SYØ:. Therefore, if you wanted a catalogue listing of files on SYØ:, you could also have omitted the device specification in the CAT examples. Thus, the following are equivalent to the CAT examples above:

```
>CAT LP:= ⊛

>CAT LP:=/S ⊛
```

If you have more than one disk drive, however, you must fully qualify the file name if you are using COPY, and you must specify a device with CAT, CAT/S. Thus:

```
>COPY LP:=SY2:HDOS.ACM ⊛

>CAT LP:=SY1: ⊛

>CAT LP:=DKØ:/S ⊛
```

Within PIP, you may use the /L, /L/S, /B, and /B/S switches to access LP:, as in the examples below:

```
:P:LP:=SY1:/L/S ⏎
```

```
:P:LP:=/B/S ⏎
```

You can also copy the contents of files to LP:, as in the following examples:

```
:P:LP:=DK1:HDOS.ACM ⏎
```

```
:P:LP:=SY2:ERRORMSG.SYS ⏎
```

You can list the format for PIP switches on your line printer by typing:

```
:P:LP:=HELP ⏎
```

Since COPY and PIP may all be used with wildcards and multiple file designations, wildcards and multiple file designations may be used to COPY, and PIP output to LP:.

To terminate output to the line printer before listing is complete, type CTRL-C.

It is possible to use more than one line-printing device with your system. Since you cannot have two files named LP.DVD, copy the device driver you want to use to any two-letter FNAME that you deem appropriate. For example, you might call the Diablo driver DB.DVD. After renaming and rebooting, you can list a file on the Diablo by copying the file to device DB: instead of LP:. If you decide to connect your second line-printing device to a port other than the default port, use the SET XX: PORT n command, with XX: being the new two-letter device driver name.

When you finish copying the device driver files that you need onto your working diskette, you can delete any other .DVD files. If you acquire a new peripheral later, you can copy the appropriate device driver file from your distribution diskette.

BOOT   (Booting from a Drive other than SYØ:)

You can use the BOOT command to initiate Bootstrap from the command mode. It is a valid command only if the specified disk contains the file BOOT.ABS. The format of the command is:

```
>DVn:BOOT ⏎
```

For example:

```
>SY1:BOOT ⏎
```

HDOS will respond with the message:

```
Device<SYØ:>?
```

At this point, you should enter the current logical name of the drive from which you want to boot. For example:

```
Device<SYØ:>?DK1: ⏎
```

In this example, HDOS would dismount all mounted disks and wait for you to insert a system volume in DK1:.

Whatever drive is specified after the "Device<SYØ:>" message is assigned the name SYØ: by HDOS, regardless of what its name is when the system is booted normally from hardware drive Ø. Other drives of the same type as the drive from which you are booting are then assigned names according to the form of the BOOT command, as listed in "Table 2-1", on the following page. Any drives that are not the same type as the drive from which you are booting are assigned the names DKn:, with the n corresponding to the hardware number of each drive.

You may want to try this command to observe the "rotation" of physical to logical devices. In daily operations, however, it is much easier to keep track of events if you always boot from the hardware device set to Ø, which then corresponds to SYØ:. Booting from the hardware device set to 1 or 2 is in practice, useful **only** when you are experiencing hardware difficulties in operating drive Ø.

Table 2-1

Determining SY: Names after Using the BOOT Command

Hardware Device Number

Command	Primary Boot	Secondary Boot	Drive Name
BOOT SY∅:	0		SY∅:
	1		SY1:
	2		SY2:
BOOT SY1:	0		SY2:
	1		SY∅:
	2		SY1:
BOOT SY2:	0		SY1:
	1		SY2:
	2		SY∅:
BOOT DK∅:		0	SY∅:
		1	SY1:
		2	SY2:
BOOT DK1:		0	SY2:
		1	SY∅:
		2	SY1:
BOOT DK2:		0	SY1:
		1	SY2:
		2	SY∅:

STAT (Displaying the System Status)

You may wish to check the status of your system from time to time. To do this, type:

    >STAT ⏎

The operating system will identify itself by printing the issue number of HDOS, the date of issue, and today's date. It will then print a series of listings that reflect the status of HDOS since the last boot-up.

Memory Usage lists the split octal addresses of the upper physical memory limit and the lower boundary of HDOS. Any address lower than the address of the lower bound of HDOS is available for your use. The maximum overlay size, given in split octal, shows the maximum length (in bytes) of the HDOS overlays.

The Overlay Status and Device Status listings make use of the following symbols:

I	Currently resident in memory
P	Permanently resident in memory
R	Capable of being read from
W	Capable of being written to
D	Capable of storing files and maintaining a directory.

There is an important distinction between the symbols "I" and "P". Any portion of the operating system whose status is "I" is being temporarily stored in memory. This means that if the space occupied by the portion of the system whose status is "I" is needed for some reason, HDOS will write that portion back to the disk and then utilize the space it formerly occupied. Somewhat more preferential treatment is given to a portion of the system whose status is "P". A portion of the system whose status is "P" remains in memory until the system is rebooted. The "P" status is reserved for indispensable portions of the operating system, such as the SY: and TT: device drivers.

The Overlay Status shows which, if any, of the operating system overlays are currently in memory. If one of the overlays is in memory, the name of the overlay will be followed by I or P.

The Device Status listing shows those devices which HDOS has entered into the device table. As the "P" symbol indicates, TT: and SY: are always permanently resident in memory. Since SY: and TT: are required whenever the operating system is running, they are also always currently resident in memory (as the "I" symbol indicates). The "RW" after TT: indicates that data can be read from as well as written to TT:, the system console terminal, while the "W" alongside LP: indicates that data may only be written to the line printer. The "D" symbol in the SY: and DK: entries signifies that SY: and DK: are devices which are capable of storing files and maintaining a directory of those files.

VER   (Displaying the Version Number of HDOS)

The VER command displays the current version number of HDOS. The format is:

```
>VER ⑨

HDOS   Version:    2.0
```

The version number is used by HDOS to make certain that the versions of HDOS and the utilities are compatible. If they are not compatible, HDOS will issue an error message when you try to run one of the incompatible programs.

DATE   (Examining and Changing the Current Date)

When you booted up the system, you were asked to enter the current date. If you wish to examine the date, type DATE while HDOS is in the command mode. If you wish to change the current date, type DATE and then your revised date, using the DD-MMM-YY format you were required to use when you booted up. Thus:

```
>DATE ⑨
HEATH HDOS ISSUE#50.06.00 15-OCT-80 14-SEP-81
```

Note that there are two dates included in the listing. The first date is the date of issue, and the second date is the date that was entered when the system was last booted-up.

In the following example, the operator changed the date to 16-SEP-81.

```
DATE 16-SEP-81 ⑨
```

Nonessential Files

Some of the system files are not absolutely necessary for everyday operations. For instance, after you have optimized your system using the SET options, you may delete the file SYØ:SET.ABS from your system volume. First, clear the S and W flags. Then use the DELETE command or the /DEL switch to remove the file. Having deleted SET, you will no longer be able to effectively use SYSGEN with this diskette, since SYSGEN expects the entire system to be intact. Therefore, you should clear the flags on SYSGEN and delete it as well. It is not absolutely necessary that you delete these files, and it may even be inconvenient in some applications; but removing nonessential files will provide more free space for program or data storage.

A list of nonessential files follows. You will have to clear the flags on all the files, including FLAGS, except the files with the .DVD extension. You can delete the device driver files using the form FNAME.DVD/S with the DELETE command or the PIP switch /DEL.

     SET.ABS
     SYSGEN.ABS
     SYSHELP.DOC
     HELP.
     ONECOPY.ABS
     FLAGS.ABS   (After clearing the flags on the preceding files.)
     ND.DVD
     ATH44.DVD
     ATH84.DVD
     ATH85.DVD
     LPH14.DVD
     LPH24.DVD
     LPH44.DVD

Since the functions provided by the programs residing on these files will no longer be available on this particular diskette, we strongly recommend that you keep at least one master system volume, in addition to the distribution diskette, which will provide all the available functions. At any rate, even though HDOS provides built-in safeguards such as write-protection, the efect of incidental household menaces such as dust, extremes of temperature, and power outages, not to mention "pilot error", can easily obliterate much tedious labor. For this reason, you should always keep backup copies of your own important files as well as the HDOS system files.

SUMMARY

Your system should now be configured for your terminal and any peripherals, and the drive seek time should be optimized for your drive or drives.

The examples you have used throughout this procedure are only a small sampling of the possible commands and functions of HDOS. By varying these examples, you will acquire "hands-on" experience. Experimentation can cause no damage, thanks to the error-detection and write-protection facilities of HDOS. If you should delete or damage a file, you can always re-SYSGEN from the distribution diskette. Therefore, do not be timid about exploring and enjoying the capabilities of HDOS.

Appendix A

GLOSSARY

NOTE: All commands are listed in full capital letters.

Access: The act of finding a storage location in memory or on a mass-storage medium in order to read data from it or write data into it. See "direct-access," "random-access," and "sequential-access."

Allocation: The act of setting aside a certain amount of memory or an area of a mass-storage device, to be used for running programs or storing data.

ASCII: Abbreviation for American Standard Code for Information Interchange, a standard 8-bit information code used to store alphanumeric data.

Block: See "cluster."

Bootstrap: The program or process by means of which communication is established between hardware and software. In order for a computer to "run", it must contain a program. In order to load programs into a computer, the computer must be running. In other words, the system must "lift itself by its bootstraps" before it can operate. Early computer systems were started, or "booted-up", by manually switching in a series of binary instructions from the front panel. Nowadays, most computer systems have bootstrap programs already loaded into read-only memories. or ROMs. The bootstrap program enables the computer to run whenever the power is turned on. Bootstrap, or "boot" can also be used to describe the process of transferring from a basic start-up program to a more sophisticated program, such as an operating system.

Buffer: An area of user or system RAM which is set aside for communications with peripherals, including the disk drives. The HDOS disk buffer consists of 256 bytes of memory, which is the same size as a diskette sector. When accessing a file, the operating system reads a sector into the buffer so that a program can gain access to the data. Buffers vary in size, depending on the peripheral with which they are associated, considerations of efficiency, and the amount of available memory. A buffer for a terminal might consist of only one byte, while a disk buffer should be at least as large as the minimum unit of storage on the disk, which is 256 bytes.

CATALOG: A command that instructs the operating system to print a summary of useful information about a set of files, such as the file names, sizes, and dates of creation.

CLOSE: A command that indicates to the opterating system that a process no longer requires access to the data in a file. If the file was changed during the execution of the program, the disk storage area utilized for the file may be updated. The directory will also be updated to reflect the changes to the file, such as its size and location on the disk media.

Cluster: A contiguous portion of storage area on the disk medium. In the case of HDOS, the minimum cluster size is two sectors of 256 bytes each.

Command: Information communicated to the operating system which instructs the system to perform some action, such as deleting a file.

Console: Another word for the peripheral from which a computer system is controlled. An "operator" or "user" communicates with the operating system by means of a console.

Contiguous: Described objects or storage areas that are located next to each other. Similar to "continuous."

COPY: The act of placing the contents of one file into another file. The data contained in the two files is then identical; however, the names and physical locations of the files are different.

Create: The act of setting-up a new file. This involves giving it a name for future reference. The operating system will find space for the file on the disk if sufficient space if available. It will also update the directory to indicate the presence of the new file, unique among the files on this particular disk.

Default: A condition that exists when no action is taken to override it. For instance, a device driver may print lines which are 80 characters in length unless it is instructed to make them shorter or longer. The default line length would then be 80 characters.

DELETE: A command that instructs the operating system to remove a file from the directory, and to free the area on the disk that it occupies for other purposes.

Destination: A file into which data is to be written.

Device: A peripheral into which data is to be written, or from which data is to be read, by means of input/output commands or instructions.

Device Driver: An operating system program that controls a peripheral. See "device independence."

Device Independence: A feature that allows a user program to refer to a peripheral by a symbolic name, as if it were a file, instead of requiring a section of the program to be written specifically for the purpose of controlling the peripheral. Thus, a program can command the operating system to input data to the named device or output data from it. The operating system, in turn, uses a device driver which is associated with the device name in order to accomplish the I/O.

Diagnostic: A program that is used to troubleshoot computer systems, or the various components of a computer system. The most common "diagnostics" are programs that are used to find possible read/write errors in memory devices.

Direct-Access: A concept used with some disk systems to describe the ability to access a given block of data by using the directory to find its physical position on the disk. This eliminates the need to read all the data that precedes the desired block as a means of finding it. The term "random access" is sometimes used to describe this capability.

Directory: A data area used by the operating system that holds the location and size of each disk file, referenced by its name. In some ways, it is similar to a city telephone directory, but with files instead of people.

Driver: See "device driver."

Extension: The portion of a file name that distinguishes it from another file of the same general category. For instance, an assembled language program that is used to compute poker odds could be called "POKER.ASM", while the assembly machine-instructions for the program could be stored on a file called "POKER.ABS". The extension is the portion of the name that is located to the right of the period; it may consist of zero to three characters under HDOS.

File: A data structure that is generally associated with a disk or other direct-access device. The disk is analogous to an office filing cabinet, with the files corresponding to the folders of information on the magnetic recording medium of the disk. Data is read from files and written onto files by means of operating system commands which reference each file by a unique name. The system handles the problems of finding the data and making it available to a process. Files must be "open" to be accessed, and must be "closed" when no longer needed.

Free: The act of making an area of memory available for other purposes. For example, when a file is closed, its buffer is "freed".

Handler: See "device driver."

Hard Error: A disk read/write error caused by a malfunction in the electronic or electromechanical hardware which does not go away when successive attempts to read or write are made. A hard error is usually the result of an error in writing caused by dust, static electricity, a scratched disk, or by various kinds of electronic interference or noise from electric motors, radio transmitters, and so on.

Initialize: A command to the operating system that instructs it to prepare a floppy-disk for new data storage. A new floppy-disk must be initialized before you can use it. If the floppy-disk already contains data, the data will be destroyed if that volume is initialized.

INITIALIZATION: A utility which prepares a floppy disk for new data storage. New floppy disks must be initialized before they can be used. If a diskette already contains data, the data will be destroyed during the initialization process.

I/O: Abbreviation for input/output.

Interrupt: A hardware signal to the computer, used extensively by operating systems, that causes the current process to cease and another to take it place. This facility speeds up the operation and handling of peripherals. The interrupt routine is similar to a subroutine in that it eventually returns control to the original process. The difference is that an interrupt may occur at almost any time and is controlled by external events, such as a keystroke at a terminal.

Library: A collection of programs that may be stored on the same disk and used in conjunction with each other. For example, an operating system can be a library of separate programs that are capable of calling one another.

Load: The process of transferring data from a peripheral into RAM.

Loader: A program that transfers data from a peripheral into RAM.

Map: A picture of how data and programs are distributed in memory, or a table which shows where files are located on a mass storage device.

Medium: Generally a magnetic substance, such as a floppy-disk surface, upon which data can be permanently recorded. Media can usually be removed and replaced by other physically similar media.

OPEN: A command to the operating system that makes the contents of a specific file available to a process.

Operating System: A rather complicated set of programs that is generally associated with disks and other mass storage devices. Its function is analogous to that of a policeman directing traffic at a busy intersection. Specifically, it may keep track of large amounts of data on disk files, control peripherals, control the distribution of memory among various programs, regulate the execution of programs, keep track of the amount of time and memory used for various purposes, and even improve its own speed and efficiency. The degree of sophistication is generally directly related to the size and cost of the computer system.

Overhead: That portion of the computer system's time, memory, and storage required to implement the functions of the system. The time, memory, and storage required for overhead is not available to the user.

Overlay: A fixed-size area of memory that is shared, in turn, by more than one process. For example, HDOS may require extra memory for the purpose of opening and closing files; the available memory may also be required by a user program. When this occurs, HDOS will save the contents of that memory area in a file, "swap-in" the appropriate subprogram, open or close the appropriate file(s), and then "swap-in" the original contents of the area. The process is called "overlaying."

Primary Memory: The high-speed RAM in which programs are executed and in which data is stored so as to be immediately accessible.

Protection: The means by which any of the various processes of the operating system are prevented from writing over an important area of memory or disk space.

RAM: An acronym for "Random Access Memory": A RAM allows a given location to be read from or written to in the same amount of time as any other equivalent location, regardless of physical position. This term is sometimes used interchangeably with "direct access".

Read: The act of examining the contents of a memory location, or the process of transferring the contents of a file into a buffer or area of RAM.

Real-Time Clock: An electronic counter that interrupts the processor at given time intervals. The H8 and H89 have a real-time clock which generates interrupts at intervals of two milliseconds.

RENAME: A command that changes the name of a file without affecting its contents or physical location.

Resource: A valuable portion of a computer system, such as a peripheral, a memory, or a program. Resources can be shared by several processes in advanced system; in any case, they are reusable and relatively permanent.

ROM: An acronym for "Read-Only Memory." A ROM is a memory whose contents cannot be changed; however, it can be read like any other memory.

Secondary Memory: Generally, a large-volume, low-cost, and relatively slow memory device. It can be a peripheral such as a diskette. In the case of large computer systems, it could be a cheaper version of the primary memory, where programs and data are stored when they are not begin used frequently.

Sector: The minimum accessible unit of storage on a disk. The size may be determined by physical or logical parameters, or both. In the case of the diskette and HDOS, the sector size is 256 bytes; while the minimum file size is one cluster, or two sectors.

Seek: The action taken by the head of a disk drive in finding the correct track when data is read from a file or written into a file. "Seek time" partially determines the speed of a disk access. The other main factor, called "rotational latency", is the time required for the desired sector to rotate under the head.

Sequential-Access: A type of I/O in which a unit of storage can be made available for reading or writing only by reading every unit of storage which precedes it on the recording medium. This may result from the physical characteristics of the storage device, such as a magnetic tape, or it may be a limitation imposed by the operating system.

Soft Error: An error in reading a disk that may be caused by dust, noise, or an interrupt at the incorrect time. It is similar to the "hard error" except that a soft error may be corrected by an attempt to repeat the failed process. If several retries do not correct the problem, the error is reclassified as a hard error.

Source: In the case of operating system commands, the source is the original file, which is to be renamed or copied. In the case of programs, the source is the highest-level code which is converted by the compiler, interpreter, or assembler into machine-executable instructions, or "object code".

Swap: The act of removing the contents of a memory area temporarily while the memory is used for other purposes. See "overlay."

Switch: A symbolic code that is used to issue a command to the operating system. Also a variable that is interpreted by a process in order to influence its flow-of-control.

Syntax: The formal or "rigid" order in which commands or instructions must be written to enable an operating system or other software process to recognize them.

Target: See "destination."

Track: A circular area on a disk that consists of a given number of sectors. In the case of 5-1/4-inch disks, HDOS allocates 40 tracks per disk, with each track composed of 10 sectors. In the case of 8-inch disks, HDOS allocates 77 tracks per disk. The tracks on a single-density 8-inch disk are subdivided into 13 sectors; the tracks on a double-density 8-inch disk are subdivided into 26 sectors.

Utility: A program which is called upon by either the user or the operating system in order to perform a function or group of functions. PIP, EDIT, INIT, SYSGEN, ONECOPY, etc. are all examples of HDOS utilities.

Volume: An interchangeable storage unit, such as a cassette tape or a floppy-disk. The volume contains data and is placed in a "drive" so the data may be "accessed."

Write: The act of transferring data into a memory location or register, or outputting it to a peripheral, including a disk. The head on a disk writes binary information onto the magnetic medium, which is the physical location of a given file.

Appendix B

SYSTEM ERROR MESSAGES

This section describes the error messages generated by the HDOS operating system. Error message falls into two general categories: those which start with ?nn,where nn = two digits, and those which don't. Error messages with no ?nn are produced by the program you are currently running. For example, if you are using the Text Editor, EDIT, and get a message with no ?nn in it, look in the Text Editor Manual for an explanation. Messages with the ?nn in them are produced by some component of the HDOS operating system, and are discussed here. The messages are grouped according to their ?nn number.

?00 — Bootstrap Errors

Error messages which start with ?00 are generated by the system while it is being booted up.

?00 DISK READ ERROR DURING BOOT

> An unrecoverable (hard) disk error occurred during the bootstrap process. Try booting again. If the problem persists, either your drive or your volume is bad. Try booting a different system disk.

?00 * ERROR *

> An unrecoverable (hard) disk error occured while checksuming the disk. The sector number printed immediately after this message is the one containing the error.

?00 REQUIRED FILE HDOS.SYS MISSING

> The file HDOS.SYS is not on the volume in SY∅:. The disk has not been SYSGENed, or has been SYSGENed incorrectly. Reinitialize it and then SYSGEN it correctly.

?00 THIS DISK HAS NOT BEEN PROPERLY SYSGENED

> Some error in the format of the HDOS system files was detected. The disk cannot be booted. The disk must be reinitialized, and then SYSGENed.

?00 THIS DISK MUST BE INITIALIZED AND THEN SYSGENED BEFORE IT CAN BE USED

> This disk must be initialized before you can use it. This message normally appears when you try to boot up a disk that has been destroyed by TEST.

?00 THIS DISK MUST BE SYSGENED BEFORE IT CAN BE BOOTED.

> This disk has not been SYSGENed, and thus can not be booted as a system disk. Use SYSGEN to make it a system disk.

?01 — Build Phase Errors

These error messages appear during the second half of the boot process when the HDOS operating system is being built into memory from the system disk. Most of these messages indicate damage to the files on the disk. First, try rebooting the system. If the problem persists, then this disk cannot be booted as a system disk. If you own two drives, mount the disk in SY1: and copy the files you want to keep onto a different disk. If you own only one drive, use ONECOPY (run by booting up on some other disk) to copy off your important files. Then, reinitialize the disk, and reSYSGEN it.

?01 DISK I/O ERROR DURING BOOT

> An unrecoverable (hard) disk error occurred on the system disk, and the boot operation cannot proceed. The disk volume may be bad, or you may have a bad drive. Retry the boot.

?01 DISK STRUCTURE IS CORRUPT

> The directory and/or the free space table on this disk are damaged, and HDOS cannot restore the damaged files. Contact Heath Technical Correspondence for advice.

?01 FORMAT ERROR IN DRIVER FILE

> The file does not contain a valid driver program. Currently, only Heath-written device drivers are supported. You should not attempt to write your own.

?01 HDOS REQUIRES AT LEAST 24K!

> Your system does not contain enough RAM to run HDOS, or the RAM is faulty, or it is not addressed correctly. Use a memory diagnostic to make sure that the RAM is working properly, and is jumpered to the correct addresses.

?01 MISSING FILE SY∅:HDOSOVLn.SYS

> The file SY∅:HDOSOVLn.SYS is necessary to run HDOS, and is not present. This normally indicates an incorrectly SYSGENed disk.

?01 OVERLAY TOO BIG

> One of the HDOS overlays is too large to fit into memory.

?01 SYSTEM NOT SYSGENED PROPERLY, OR FILES DAMAGED

> A system file is damaged. This can be the result of a software or hardware error.

?01 TOO MANY DEVICE DRIVERS

> The system volume from which you are attempting to boot-up contains more than five device drivers. Exclusive of SY.DVD, the maximum number of device drivers that may be contained on a disk is currently five.

?01 UNABLE TO MOUNT SYSTEM DISK

> The system volume from which you are attempting to boot-up does not contain the file SY.DVD, the system disk driver.

?02 — Error Messages

These messages are generated by the operating system and may appear at any time. Usually they are in response to some request from the program you are running which, in turn, is usually caused by some command from you. Normally, HDOS looks up these error messages in the file SYØ:ERRORMSG.SYS to give an understandable message. If the file SYØ:ERRORMSG.SYS is missing, or if the system disk has been dismounted, HDOS will simply type the error message number. The numbers are listed first, followed by the message they represent. Look up the message in the second group for a discussion of its meaning.

Most of these error messages will be meaningless to you; they are generated by HDOS when a program makes a mistake when issuing a request to HDOS. The Heath products supplied with HDOS will not make these mistakes. Normally, only users debugging assembly language programs will see most of these error messages. The ones that the average user will see are self explanatory.

?02 SYS ERR # 001

> End of file.

?02 SYS ERR # 002

> No free space on media.

?02 SYS ERR # 003

> Illegal "SYSCALL" function code.

?02 SYS ERR # 004

> Channel is already in use.

?02 SYS ERR # 005

> Device is not capable of this operation.

?02 SYS ERR # 006

> Illegal format for device name.

?02 SYS ERR # 007

> Illegal format for file name.

?02 SYS ERR # 008

    Not enough memory for the device driver.

?02 SYS ERR # 009

    Channel is not open.

?02 SYS ERR # 010

    Illegal function request.

?02 SYS ERR # 011

    File usage conflicts.

?02 SYS ERR # 012

    File cannot be located.

?02 SYS ERR # 013

    Unknown device name.

?02 SYS ERR # 014

    Illegal channel number.

?02 SYS ERR # 015

    The volume directory is full.

?02 SYS ERR # 016

    The file's contents are not correct for this operation.

?02 SYS ERR # 017

    Not enough RAM for this program.

?02 SYS ERR # 018

    Read failure on the device.

?02 SYS ERR # 019

    Write failure on the device.

?02 SYS ERR # 020

      Attempted write-protection violation.

?02 SYS ERR # 021

      Disk is write-protected.

?02 SYS ERR # 022

      The file is already present.

?02 SYS ERR # 023

      Aborted by device driver.

?02 SYS ERR # 024

      File is locked against flag change.

?02 SYS ERR # 025

      A file is already open.

?02 SYS ERR # 026

      Illegal or unknown switch specified.

?02 SYS ERR # 027

      Unknown unit for this device.

?02 SYS ERR # 028

      Non-null file name is required.

?02 SYS ERR # 029

      Device is incapable of write operations (or is write locked).

?02 SYS ERR # 030

      Unit not available.

?02 SYS ERR # 031

      Illegal value.

?02 SYS ERR # 032

>   Illegal option.

?02 SYS ERR # 033

>   Volume presently mounted on the device.

?02 SYS ERR # 034

>   No volume presently mounted on the device.

?02 SYS ERR # 035

>   File open on the device.

?02 SYS ERR # 036

>   No provisions made for remounting more disks.

?02 SYS ERR # 037

>   This disk must be initialized before it can be mounted.

?02 SYS ERR # 038

>   Unable to read this disk, it probably has not been properly initialized.

?02 SYS ERR # 039

>   Disk structure is corrupt. Contact Technical Correspondence for help.

?02 SYS ERR # 040

>   Not the correct version of HDOS for this program.

?02 SYS ERR # 041

>   No operating system mounted, required for this operation.

?02 SYS ERR # 042

>   Illegal overlay index.

?02 SYS ERR # 043

>   Overlay too large.

?02 ABORTED BY DEVICE DRIVER

> The device driver could not successfully complete the requested operation.

?02 A FILE IS ALREADY OPEN

> The specified channel is already open.

?02 ATTEMPTED WRITE PROTECTION VIOLATION

> You requested a write-type operation on a write-protected file. These write-type operations are WRITE, DELETE, RENAME, and REPLACE.

?02 CHANNEL IS ALREADY IN USE

> The I/O channel specified in the HDOS call is already in use.

?02 CHANNEL IS NOT OPEN

> The I/O channel must be opened before you issue this request.

?02 CAN'T RUN SYØ:SYSCMD.SYS

> The file SYØ:SYSCMD.SYS is either damaged or missing, and is necessary to run HDOS.

?02 DEVICE IS INCAPABLE OF WRITE OPERATION (OR IS WRITE LOCKED)

> The device is write-disabled, or is incapable of accepting write operations.

?02 DEVICE IS NOT CAPABLE OF THIS OPERATION

> The device specified is not capable of the operation. For example: a read request from a write-only device such as a line printer, or a directory operation such as LIST on a non disk device.

> The most common cause of this message is a write operation on a disk that has the write-protect tab installed.

?02 DISK IS WRITE PROTECTED

> The write operation was refused because the disk is write protected.

?02 DISK STRUCTURE IS CORRUPT. CONTACT TECHNICAL CORRESPONDENCE FOR HELP.

> The internal map of the disk has changed in HDOS. The most common cause is incorrectly dismounting or mounting diskettes, especially when they have the same volume number.

?02 END OF FILE

> An End-of-File was read on the file. There are no more sectors to read.

?02 FATAL SYSTEM ERROR

> A read or write error occurred on a distribution or a system diskette. The distribution diskette must always be inserted in SY∅:.

?02 FILE CANNOT BE LOCATED

> The specified file name is not on the specified device.

?02 FILE IS LOCKED AGAINST FLAG CHANGE

> The file cannot have its flags changed because the L (locked) flag is set.

?02 FILE OPEN ON THE DEVICE

> A dismount or reset was issued to a device, and a channel was not previously closed.

?02 FILE USAGE CONFLICTS

Conflicting requests have been made for this file. A file may not be deleted, replaced, written to, or renamed while it is open for read. Also note that a program may not delete, replace, write to, or rename the file it was loaded from.

?02 ILLEGAL 'SYSCALL' FUNCTION CODE

An illegal request code was given in an assembly language SCALL statement. This will only occur with user-written programs.

?02 ILLEGAL CHANNEL NUMBER

A request specified a nonexistent channel number.

?02 ILLEGAL FORMAT FOR DEVICE NAME

The device specification part of the file name is not correctly specified.

?02 ILLEGAL FORMAT FOR FILE NAME

The file specification was in an illegal format.

?02 ILLEGAL FUNCTION REQUEST

A request was made to HDOS to perform an illegal function. For example: requesting a write to a channel opened for read.

?02 ILLEGAL OR UNKNOWN SWITCH SPECIFIED

An illegal or unknown option switch (/xxx) was specified in the command line.

?02 ILLEGAL OPTION

A specified SET option was not recognized by HDOS.

?02 ILLEGAL OVERLAY INDEX

A call was made to an invalid index number.

?02 ILLEGAL VALUE

The value entered was out of bounds. This error occurs most commonly in a SET command.

?02 NO FREE SPACE ON MEDIA

> All sectors on the volume are in use, so the write request cannot be
> honored.

?02 NO OPERATING SYSTEM MOUNTED, REQUIRED FOR THIS OPERATION

> HDOS and the required overlays are not present. Check the directory of
> SY∅:.

?02 NO VOLUME PRESENTLY MOUNTED ON THE DEVICE

> A dismount was issued to a device that had nothing previously
> mounted.

?02 NO PROVISIONS MADE FOR REMOUNTING MORE DISKS

> The overlays of HDOS required to process a mount command are not
> present.

?02 NON-NULL FILE NAME IS REQUIRED

> Disk files require that the name field in the file specification must
> contain at least one character. The extension may be empty (null).
> Non-disk devices allow empty (null) file names.

?02 NOT THE CORRECT VERSION OF HDOS FOR THIS PROGRAM

> The version numbers of HDOS and the requested program do not agree.
> You should make sure that all files have the most recent version
> numbers.

?02 NOT ENOUGH MEMORY FOR THE DEVICE DRIVER

> Not enough free RAM exists to load the necessary device driver.

?02 NOT ENOUGH RAM FOR THIS PROGRAM

> There is not enough free RAM to load this program.

?02 OVERLAY TOO LARGE

> The assemble overlay exceeds maximum size. It cannot be larger than
> HDOSOVL∅.SYS:

?02 READ FAILURE ON THE DEVICE

> An unrecoverable (hard) error occurred on the last attempted read operation from this device.

?02 THE FILE IS ALREADY PRESENT

> You attempted to rename a file to a new name that already exists in that volume's directory.

?02 THE FILE'S CONTENTS ARE NOT CORRECT FOR THIS OPERATION

> An attempt to RUN a file which is not an absolute binary program. Only absolute binary (assembly language) files may be run (the programs supplied by Heath are all in this format). Note that the extension .ABS, by convention, represents absolute binary files. But an absolute binary file does not have to have the .ABS extension, and a file may have the .ABS extension and yet not be in absolute binary format.

?02 THE VOLUME DIRECTORY IS FULL

> The volume's directory is full; no more file names can be added until some are deleted. A volume directory for the diskette holds about 198 file names.

?02 UNABLE TO READ THIS DISK, IT PROBABLY HAS NOT BEEN PROPERLY INITIALIZED

> An attempt was made to mount a diskette that appears to be uninitialized. Be sure that the diskette was re-initialized after "Test".

?02 UNIT NOT AVAILABLE

> The device unit requested is not installed or not operable.

?02 UNKNOWN DEVICE NAME

> An unknown device was specified in the file name. Note that a device driver for each device you wish to use must be on the system disk when it is booted (except for SY: and TT:, which are built into HDOS).

?02 UNKNOWN UNIT FOR THIS DEVICE

> This device type does not have the specified unit number. The most common cause of this error is specifying a disk drive name without first mounting the disk in that drive.

?02 THIS DISK MUST BE INITIALIZED BEFORE IT CAN BE MOUNTED

> The volume you have attempted to mount does not contain the necessary HDOS formatting. Only initialized diskettes may be mounted.

?02 WRITE FAILURE ON THE DEVICE

> An unrecoverable (hard) error occurred on the last attempted write operation on this device.

?02 VOLUME PRESENTLY MOUNTED ON THE DEVICE

> A mount was issued to a device that has not been dismounted.

Appendix C

HDOS COOKBOOK

This appendix provides you with a number of examples of possible HDOS functions and commands. The possible applications of the command will be given along with the exact command syntax, including prompts.

These examples are not intended as explanations of how HDOS accomplishes various functions. Rather, they are intended as a survey of some of the more useful commands and options. For a more detailed explanation of any of the commands listed below, refer to the "General Operation" section.

Listing Files

The following variations of the TYPE command allow you to type the contents of various files on your system console terminal. It is necessary to specify a device in the file name if you wish to type files from a disk mounted in a drive other than SYØ:. You may type files from a disk on SYØ: without an actual specification of SYØ:. Wildcards and multiple file designations are valid and cause files to be listed one after the other. Note that you may use CTRL-C to stop the listing entirely, CTRL-S to halt it temporarily, and CTRL-Q to restart the listing after a CTRL-S. We recommend that you do not type files with the .ABS, .DVD, or .SYS extensions because such files do not normally contain meaningful information written in ASCII. If you wish to type the contents of system files using a wild card, you must use the /S modifier.

```
>TYPE FNAME.EXT
>TYPE FNAME.EXT/S
>TYPE DVn:FNAME.EXT
>TYPE DVn:FNAME.EXT/S
>TYPE *.EXT
>TYPE *.EXT/S
>TYPE *.*
>TYPE *.*/S
>TYPE FNAME.EXT,. . .,FNAMEn.EXT
>TYPE DVn:FNAME.EXT,. . .,DVn:FNAMEn.EXT
>TYPE DVn:*.EXT,. . .,DVn:FNAMEn.*
```

Cataloging Files

The following commands produce a catalogue listing of a specific file or files. If you want a catalogue listing of files on a disk mounted in a drive other than SYØ:, you must specify the device name. If you specify no device name, HDOS catalogues SYØ:. You must use the /S modifier if a system file is to be catalogued. If the * wildcard is used in place of either FNAME or .EXT, multiple files may be catalogued if two or more files have the * part in common. Note that there are a great many possible commands that are not shown.

```
>CAT
>CAT/S
>CAT FNAME.EXT
>CAT FNAME.EXT/S
>CAT FNAME.EXT,. . .,FNAMEn.EXT
>CAT DVn:
>CAT DVn:/S
>CAT DVn:FNAME.EXT
>CAT DVn:FNAME.EXT/S
>CAT DVn:*.EXT
>CAT DVn:*.EXT/S
>CAT DVn:FNAME.*
>CAT DVn:*.*
>CAT DVn:*.*/S
```

Mounting and Dismounting Disks

If you want to use disks with devices other than SYØ:, it is necessary that you mount them on the H17 and/or H47 drive units. The system volume (the disk in SYØ:) is automatically mounted during bootstrap. When you have finished using a volume, it is necessary to dismount it before you power the system down or mount a new volume. If you do not dismount volumes, data which is held in memory may be lost because HDOS is not given a chance to update the volume(s). The correct procedures for mounting and dismounting disks are as follows:

```
>MOUNT SY1: 🅡
Volume 002, Mounted on SY1:
Label: SPARE SYSTEM VOLUME

>MOUNT DK1: 🅡
Volume 180, Mounted on DK1:
Label: ASSEMBLY PROGRAMS

>DISMOUNT DK1: 🅡
Volume 180, Dismounted from DK1:
Label: ASSEMBLY PROGRAMS

>DISMOUNT SY1: 🅡
Volume 002, Dismounted from SY1:
Label: SPARE SYSTEM VOLUME

>DISMOUNT DKØ: 🅡
Volume 005, Dismounted from DKØ:
Label: GENERAL LEDGER AUGUST 1980

>DISMOUNT SYØ: 🅡
Volume 001, Dismounted from SYØ:
Label: SYSTEM VOLUME

Install a Bootable Disk in SYØ: Hit RETURN to Reboot:
```

Running Programs

Executable binary programs have the .ABS extension under the HDOS conven-
tion. This extension enables you to type the FNAME portion of a file name from
the command mode as an abbreviated command to run the program contained in
the file SYØ:FNAME.ABS. In order to run a program contained on a file in a
device other than SYØ:, you must specify a device in the file name. For example:

```
>FNAME
>DVn:FNAME
>RUN FNAME
>RUN FNAME.ABS
>RUN DVn:FNAME
>RUN DVn:FNAME.ABS
```

Duplicating Files

It is possible to copy the contents of one or more files with a single command.
The general command syntax is:

```
>COPY DVn:DESTINAT.EXT=DVn:SOURCE.EXT ⊛
```

Some examples of this command are:

```
>COPY DVn:FNAME.EXT=FNAME.EXT
>COPY FNAME.EXT=DVn:FNAME.EXT
>COPY DVn:FNAME.EXT=AT:
>COPY DVn:*.*=DVn:*.EXT
>COPY AT:=*.*
```

Deleting Files

It is possible to delete one or more files using only one DELETE command. You
must be careful with this command because valuable information could be lost if
you were to delete the wrong file(s). System files are protected by flags and
cannot be deleted until the flags are changed. Those files which are both write-
protected and locked can never be deleted unless the diskette is reinitialized.
Here are a few examples of the DELETE command:

```
>DELETE FNAME.EXT
>DELETE DVn:FNAME.EXT
>DELETE DVn:FNAME.EXT,. . .,FNAMEn.EXT
>DELETE *.EXT
>DELETE FNAME.*
>DELETE *.*
```

Peripheral Interchange

In general, you can use PIP to accomplish the same things that you can accomplish in the command mode. This does not include running programs, or such general-purpose functions as SET or DATE. However, PIP is the basis for most command mode file manipulations. The following is a list of some useful PIP commands:

```
:P:HELP
:P:FNAME.EXT
:P:DVn:FNAME.EXT
:P:DVn:DESTINAT.EXT=DVn:SOURCE.EXT
:P:DVn:DESTINAT.EXT=DVn:SOURCE.EXT/S
:P:/L
:P:/L/S
:P:LP:=/L
:P:DVn:FNAME.EXT/L
:P:FNAME.*/L
:P:FNAME.EXT,. . .,FNAMEn.EXT/L/S
:P:/B
:P:/B/S
:P:*.*/B/S
:P:DVn:*.*=*.*
:P:NEWNAME.EXT=OLDNAME.EXT/R
:P:DVn:FNAME.EXT,. . .,DVn:FNAMEn.EXT/DEL
:P:DVn:/RES
:P:DVn:/DIS
:P:/VER
```

System Optimization

There are several variations of the SET command which enable you to optimize the performance of your disk drive system. For instance, you can adjust the seek time of your disk drives to operate at highest reliable speed. You can also configure the operating system to utilize the console terminal in such a way that the terminal can backspace and support lower-case input and output. Use the following format to obtain help with the SET command:

```
>SET HELP
```

General Command Format:

```
SET xx: opt
   xx: --  Device Name
   opt --  Desired Option
```

For HELP with a specific device, type:

```
SET xx: HELP
```

To determine the version of SET, type:

```
SET Ver
```

ONECOPY

This program is a stand-alone utility which enables owners of single-drive systems to copy files from one diskette to another. The destination file(s) automatically have the same name as the source file(s). The number of times that you need to swap diskettes will depend upon how much memory your computer has. The following list demonstrates some of the possible ONECOPY commands:

```
:OC:FNAME.EXT
:OC:FNAME.*
:OC:*.EXT
:OC:*.*
:OC:*.*/L/S
:OC:/B
:OC:/B/S
:OC:/MOU
```

Appendix D

HDOS PATCH UTILITY (PATCH)

PATCH is an HDOS utility used to patch absolute binary programs. PATCH may be used to patch user-written programs which have not been write-protected. PATCH will not modify any Heath-supplied system programs, nor will it patch assembly source programs or BASIC programs.

You can patch locations in your program that are not defined within the program, but those locations must follow the current last-word address of your program. Thus, if your program occupied locations 042200 to 050000, you could extend the program by entering patches to locations greater than 050000.

To use PATCH:

1. Run PATCH, using the command format RUN DVn:PATCH, or simply DVn:PATCH.

2. PATCH will prompt you for a file name. Enter the device name and the name of the binary file you wish to patch. For example:

    SY1:DEMO2.ABS

3. PATCH will now prompt you with "ADDRESS?". Enter the address of the first patch as a byte-octal number. For example:

    ADDRESS?   042200

4. PATCH will then display an address and byte value, followed by a backslash (\). You can reply in one of three ways:

    A. Type a three-digit new value.

    B. Type ⑨ to leave this byte unchanged.

    C. Type CTRL-D to bring back the ADDRESS? prompt.

5. When you have finished patching your program, type CTRL-D in response to the ADDRESS? prompt. PATCH will then insert the patches into your program.

NOTE

PATCH does not patch the program until the entire series of patches has been entered and CTRL-D has been typed in response to the ADDRESS? prompt. Until that time, you may use CTRL-C or CTRL-Z to cancel the patch session and leave your file unchanged.

Appendix E

H17 ROM CODE LISTING

H17ROM - H17 ROM CODE LISTING.
MEMORY DIAGNOSTIC.

```
50.                   *       COPYRIGHT (C) HEATH CO., 1977
53.
54.  030.000                  ORG     30000A
55.
56.
57.  030.000  303 014 037     JMP     BOOT            ;BOOT CODE

59.                   **      MEMORY DIAGNOSTIC.
60.                   *
61.
62.  030.003  041 300 377     LXI     H,-64
63.  030.006  071             DAD     SP
64.  030.007  353             XCHG                    ;(HL) = END
                                                      ;(DE) = END+1
65.  030.010  041 100 040     LXI     H,40100A        ;(HL) = START
66.  030.013  166             HLT                     ;PAUSE FOR ADJUSTMENT
67.
68.
69.                   *       (HL) = START
70.                   *       (DE) = END
71.
72.                   *       ZERO TEST AREA
73.
74.  030.014  042 076 040 MEM1  SHLD  40100A-2
75.  030.017  066 000         MVI     M,0
76.  030.021  043             INX     H
77.  030.022  315 216 030     CALL    $CDEHL
78.  030.025  302 017 030     JNE     MEM1
79.
80.                   *       START TESTING MEMORY. INCREMENT EACH BYTE IN TURN, AND COMPARE
81.                   *       THAT RESULT TO THE EXPECTED VALUE
82.
83.  030.030  006 000  MEM2   MVI     B,0             ;(B) = EXPECTED VALUE
84.  030.032  052 076 040     LHLD    40100A-2
85.  030.035  004             INR     B
86.
87.  030.036  064     MEM3    INR     M
88.  030.037  176             MOV     A,M
89.  030.040  270             CMP     B
90.  030.041  312 046 030     JE      MEM4            ;(A) = VALUE
                                                      ;IS OK
91.
92.                   *       HAVE ERROR. (HL) = ADDRESS OF BYTE IN ERROR
93.
94.  030.044  166             HLT
95.  030.045  000             NOP
96.
97.  030.046  043     MEM4    INX     H
98.  030.047  315 216 030     CALL    $CDEHL
99.  030.052  302 036 030     JNE     MEM3            ;NOT AT END OF PASS
100. 030.055  303 032 030     JMP     MEM2            ;AT END OF PASS
```

```
HEATH H8ASM V1.4  01/20/78          PAGE   3
15:54:10  11-MAY-78
```

H17ROM - H17 ROM CODE LISTING.
COMMON DECKS.

```
                   105X  **    $COMP - COMPARE TWO CHARACTER STRINGS.
                   106X  *
                   107X  *     $COMP COMPARES TWO BYTE STRINGS.
                   108X  *
                   109X  *     ENTRY   (C) = COMPARE COUNT
                   110X  *             (DE) = FWA OF STRING #1
                   111X  *             (HL) = FWA OF STRING #2
                   112X  *     EXIT    'Z' CLEAR, IS MIS-MATCH
                   113X  *             (C) = LENGTH REMAINING
                   114X  *             (DE) = ADDRESS OF MISMATCH IN STRING#1
                   115X  *             (HL) = ADDRESS OF MISMATCH IN STRING #2
                   116X  *             'C' SET, HAVE MATCH
                   117X  *             (C) = 0
                   118X  *             (DE) = (DE) + (OC)
                   119X  *             (HL) = (HL) + (OC)
                   120X  *
                   121X  *     USES    A,F,C,D,E,H,L
                   122X
030.060  032       123X  $COMP  LDAX  D
030.061  276       124X         CMP   M          COMPARE
030.062  300       125X         RNE              NO MATCH
030.063  023       126X         INX   D
030.064  043       127X         INX   H
030.065  015       128X         DCR   C
030.066  302 060 030 129X       JNZ   $COMP      TRY SOME MORE
030.071  311       130X         RET              HAVE MATCH


                   133X  **    $DADA - PERFORM (H,L) = (H,L) + (O,A)
                   134X  *
                   135X  *     ENTRY   (H,L) = BEFORE VALUE
                   136X  *             (A) = BEFORE VALUE.
                   137X  *     EXIT    (H,L) = (H,L) + (O,A)
                   138X  *             'C' SET IF OVERFLOW
                   139X  *     USES    F,H,L
                   140X
                   141X
030.072  325       142X  $DADA  PUSH  D
030.073  137       143X         MOV   E,A
030.074  026 000   144X         MVI   D,O
030.076  031       145X         DAD   D
030.077  321       146X         POP   D
030.100  311       147X         RET              EXIT
```

```
HITROM - H17 ROM CODE LISTING.                                    HEATH H8ASM V1.4  01/20/78    PAGE   4
COMMON DECKS.                                                     15:54:11  11-MAY-78

                        150X **        $DADA.    $DADA. -- ADD (O,A) TO (H,L)
                        151X *
                        152X *         ENTRY   NONE
                        153X *         EXIT    (HL) = (HL) + (OA)
                        154X *         USES    A,F,H,L
                        155X
                        156X
030.101  205            157X $DADA.    ADD     L
030.102  157            158X           MOV     L,A
030.103  320            159X           RNC
030.104  044            160X           INR     H
030.105  311            161X           RET


                        164X **                  $DU66 - UNSIGNED 16 / 16 DIVIDE.
                        165X *
                        166X *         (HL) = (BC)/(DE)
                        167X *
                        168X *         ENTRY   (BC), (DE) PRESET
                        169X *         EXIT    (HL) = RESULT
                        170X *                 (DE) = REMAINDER
                        171X *
                        172X *         USES    ALL
                        173X
030.106  172            174X $DU66     MOV     A,D           ;TWOS COMPLEMENT (DE)
030.107  057            175X           CMA
030.110  127            176X           MOV     D,A
030.111  173            177X           MOV     A,E
030.112  057            178X           CMA
030.113  137            179X           MOV     E,A
030.114  023            180X           INX     D
030.115  172            181X           MOV     A,D
030.116  263            182X           ORA     E
030.117  312 205 030    183X           JZ      DU665         ;IF DIVIDE BY 0.
030.122  257            184X           XRA     A
                        185X *
                        186X *         SHIFT (DE) LEFT UNTIL:
                        187X *         1) DE > BL
                        188X *         2) OVERFLOW
                        189X *
                        190X
030.123  142            191X DU661     MOV     H,D
030.124  153            192X           MOV     L,E
030.125  011            193X           DAD     B
030.126  322 143 030    194X           JNC     DU662         ;IS TOO LARGE
030.131  074            195X           INR     A             ;COUNT SHIFT
030.132  142            196X           MOV     H,D
030.133  153            197X           MOV     L,E
030.134  051            198X           DAD     H
030.135  353            199X           XCHG                  ;(DE) = (DE)*2
030.136  332 123 030    200X           JC      DU661         ;IF NOT OVERFLOW
                        201X *
                        202X *         (DE) OVERFLOWED, PUT IT BACK.
```

```
H17ROM - H17 ROM CODE LISTING.                                    HEATH H8ASM V1.4  01/20/78          PAGE  5
COMMON DECKS.                                                     $DU66       15:54:12  11-MAY-78

203X
204X   030.141  353          XCHG
205X   030.142  075          DCR     A              REMOVE EXTRA COUNT
206X
207X                  *       READY TO START SUBTRACTING. (A) = LOOP COUNT
208X
209X   030.143  140  DU662   MOV     H,B            (H,L) = WORKING VALU
210X   030.144  151          MOV     L,C
211X   030.145  001 000 000  LXI     B,0            (BC) = RESULT
212X   030.150  365          PUSH    PSW            SAVE (A)
213X   030.151  031          DAD     D
214X   030.152  332 163 030  JC      DU664          IF SUBTRACT OK
215X   030.155  175          MOV     A,L            ADD BACK IN
216X   030.156  223          SUB     E
217X   030.157  157          MOV     L,A
218X   030.160  174          MOV     A,H
219X   030.161  232          SBB     D
220X   030.162  147          MOV     H,A
221X   030.163  171  DU664   MOV     A,C
222X   030.164  027          RAL
223X   030.165  117          MOV     C,A
224X   030.166  170          MOV     A,B
225X   030.167  027          RAL
226X   030.170  107          MOV     B,A
227X
228X                  *       RIGHT SHFT (DE)
229X
230X   030.171  067          STC
231X   030.172  172          MOV     A,D
232X   030.173  037          RAR
233X   030.174  127          MOV     D,A
234X   030.175  173          MOV     A,E
235X   030.176  037          RAR
236X   030.177  137          MOV     E,A
237X   030.200  361          POP     PSW
238X   030.201  075          DCR     A
239X   030.202  362 150 030  JP      DU663          IF NOT DONE
240X   030.205  353  DU665   XCHG                   (D,E) = REMAINDER
241X   030.206  140          MOV     H,B            (HL) = RESULT
242X   030.207  151          MOV     L,C
243X   030.210  311          RET

246X                  **      $HLIHL - LOAD HL INDIRECT THROUGH HL.
247X                  *
248X                  *       (HL) = ((HL))
249X                  *
250X                  *       ENTRY   NONE
251X                  *       EXIT    NONE
252X                  *       USES    A,H,L
253X
254X   030.211  176  $HLIHL  MOV     A,M
255X   030.212  043          INX     H
```

```
H17ROM - H17 ROM CODE LISTING.                              HEATH H8ASM V1.4  01/20/78        PAGE  6
COMMON DECKS.                                               15:54:13  11-MAY-78
                                                $HL1HL

030.213 146    256X          MOV    H,M
030.214 157    257X          MOV    L,A
030.215 311    258X          RET

               261X ** $CDEHL - COMPARE (DE) TO (HL)
               262X *
               263X * $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
               264X *
               265X * ENTRY   NONE
               266X * EXIT    'Z' SET IF (DE) = (HL)
               267X * USES    A,F
               268X *
               269X *
030.216 173    270X $CDEHL   MOV    A,E
030.217 255    271X          XRA    L
030.220 300    272X          RNZ               IF DIFFERENT
030.221 172    273X          MOV    A,D
030.222 254    274X          XRA    H
030.223 311    275X          RET

               278X ** $CHL - COMPLEMENT (HL).
               279X *
               280X * (HL) = -(HL)             TWO'S COMPLEMENT
               281X *
               282X * ENTRY   NONE
               283X * EXIT    NONE
               284X * USES    A,F,H,L
               285X *
               286X *
030.224 174    287X $CHL     MOV    A,H
030.225 057    288X          CMA
030.226 147    289X          MOV    H,A
030.227 175    290X          MOV    A,L
030.230 057    291X          CMA
030.231 157    292X          MOV    L,A
030.232 043    293X          INX    H
030.233 311    294X          RET
```

HEATH H8ASM V1.4 01/20/78   PAGE   7
15:54:15 11-MAY-78
$INDL

```
297X  **         $INDL - INDEXED LOAD.
298X  *
299X  *          $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
300X  *
301X  *          THIS ACTS AS AN INDEXED FULL WORD LOAD.
302X  *
303X  *          (DE) = ( (HL) + DISPLACEMENT )
304X  *
305X  *          ENTRY     ((RET)) = DISPLACMENT (FULL WORD)
306X  *                    (HL) = TABLE ADDRESS
307X  *          EXIT      TO (RET+2)
308X  *          USES      A,F,D,E
309X
310X
                030.234  343   311X  $INDL  XTHL        (HL) = RET, ((SF)) = TBL ADDRESS
                030.235  136   312X         MOV   E,M
                030.236  043   313X         INX   H
                030.237  126   314X         MOV   D,M   (DE) = DISPLACEMENT
                              315X
                030.240  043   316X         INX   H
                030.241  343   317X         XTHL        ((SF)) = RET, (HL) = TBL ADDRESS
                030.242  353   318X         XCHG        (DE) = TBL ADDRESS, (HL) = DISPLACEMENT
                030.243  031   319X         DAD   D     (HL) = TARGET ADDRESS
                030.244  176   320X         MOV   A,M
                030.245  043   321X         INX   H
                030.246  146   322X         MOV   H,M
                030.247  157   323X         MOV   L,A   (HL) = ((HL))
                030.250  353   324X         XCHG        (DE) = VALUE, (HL) = TABLE ADDRESS
                030.251  311   325X         RET

328X  **         $MOVE - MOVE DATA.
329X  *
330X  *          $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
331X  *          IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
332X  *          FIRST TO LAST.
333X  *
334X  *          IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
335X  *          LAST TO FIRST.
336X  *
337X  *          THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
338X  *
339X  *          ENTRY     (BC) = COUNT (MUST BE < 32768)
340X  *                    (DE) = FROM
341X  *                    (HL) = TO
342X  *          EXIT      MOVED
343X  *                    (DE) = ADDRESS OF NEXT FROM BYTE
344X  *                    (HL) = ADDRESS OF NEXT *TO* BYTE
345X  *                    'C' CLEAR
346X  *          USES      ALL
347X
348X
                030.252        349X  $MOVE  EQU   *
```

```
H17ROM - H17 ROM CODE LISTING.                          HEATH HBASM V1.4  01/20/78    PAGE   8
COMMON DECKS.                                           15:54:16  11-MAY-78

350X  030.252 170              MOV   A,B       $MOVE
351X  030.253 261              ORA   C
352X  030.254 310              RZ              NONE TO MOVE
353X  030.255 175              MOV   A,L       COMPARE *FROM* TO *TO*
354X  030.256 223              SUB   E
355X  030.257 174              MOV   A,H
356X  030.260 232              SBB   D
357X  030.261 332 311 030      JC    MOV2      IS MOVE DOWN (TO LOWER ADDRESSES)
358X                      *
359X                      *    IS MOVE UP (TO HIGHER ADDRESSES)
360X                      *
361X  030.264 013              DCX   B
362X  030.265 011              DAD   B         (HL) = *TO* LWA
363X  030.266 345              PUSH  H         SAVE *TO* LIMIT
364X  030.267 353              XCHG
365X  030.270 011              DAD   B         (HL) = *FROM* LWA
366X  030.271 345              PUSH  H         SAVE *FROM* LIMIT
367X                      *
368X  030.272 176  MOV1        MOV   A,M       MOVE BYTE
369X  030.273 022              STAX  D
370X  030.274 033              DCX   D         INCREMENT *TO* ADDRESS
371X  030.275 053              DCX   H         INCREMENT *FROM* ADDRESS
372X  030.276 013              DCX   B         DECREMENT COUNT
373X  030.277 170              MOV   A,B
374X  030.300 247              ANA   A
375X  030.301 362 272 030      JP    MOV1      MORE TO GO
376X  030.304 321              POP   D         (DE) = *FROM* LIMIT
377X  030.305 341              POP   H         (HL) = *TO* LIMIT
378X  030.306 023              INX   D
379X  030.307 043              INX   H
380X  030.310 311              RET             DONE
381X                      *
382X                      *    IS MOVE DOWN (TO LOWER ADDRESSES)
383X                      *
384X  030.311 032  MOV2        LDAX  D         MOVE BYTE
385X  030.312 167              MOV   M,A
386X  030.313 043              INX   H         INCREMENT *FROM*
387X  030.314 023              INX   D         INCREMENT *TO*
388X  030.315 013              DCX   B         DECREMENT COUNT
389X  030.316 170              MOV   A,B
390X  030.317 261              ORA   C
391X  030.320 302 311 030      JNZ   MOV2      IF NOT DONE
392X  030.323 311              RET             DONE

395X                      **   $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
396X                      *
397X                      *    (HL) = (DE)*10
398X                      *
399X                      *    ENTRY    (DE) = MULTIPLIER
400X                      *    EXIT     'C' CLEAR IF OK
401X                      *             (HL) = PRODUCT
402X                      *             'C' SET IF ERROR
```

```
HI7ROM - H17 ROM CODE LISTING.                          HEATH H8ASM V1.4  01/20/78      PAGE  9
COMMON DECKS.                                           15:54:17  11-MAY-78

                                      403X  *          USES  D,E,H,L,F
                                      404X
                                      405X
030.324  353          406X  $MU10     XCHG              ;(HL) = MULTIPLIER
030.325  051          407X            DAD   H           ;(HL) = X*2
030.326  330          408X            RC
030.327  124          409X            MOV   D,H
030.330  135          410X            MOV   E,L
030.331  051          411X            DAD   H           ;(HL) = X*4
030.332  330          412X            RC
030.333  051          413X            DAD   H           ;(HL) = X*8
030.334  330          414X            RC
030.335  031          415X            DAD   D           ;(HL) = X*10
030.336  311          416X            RET

                      419X  **        $MU66 - UNSIGNED 16X16 MULTIPLY.
                      420X  *
                      421X  *         ENTRY (BC) = MULTIPLICAND
                      422X  *               (DE) = MULTIPLIER
                      423X  *         EXIT  (HL) = RESULT
                      424X  *               ;Z; SET IF NOT OVERFLOW
                      425X  *         USES  ALL
                      426X
                      427X
030.337  257          428X  $MU66     XRA   A           ;SAVE OVERFLOW STATUS
030.340  365          429X            PUSH  PSW
030.341  041.000.000  430X            LXI   H,0         ;(HL) = RESULT ACCUMULATOR
                      431X
030.344  170          432X  MU661     MOV   A,B
030.345  037          433X            RAR
030.346  107          434X            MOV   B,A
030.347  171          435X            MOV   A,C
030.350  037          436X            RAR
030.351  117          437X            MOV   C,A
030.352  322.364.030  438X            JNC   MU662       ;IF BIT CLEAR
030.355  031          439X            DAD   D
030.356  322.364.030  440X            JNC   MU662       ;IF NOT OVERFLOW
030.361  361          441X            POP   PSW
030.362  074          442X            INR   A
030.363  365          443X            PUSH  PSW
030.364  170          444X  MU662     MOV   A,B         ;SEE IF MULTIPLIER 0
030.365  261          445X            ORA   C
030.366  312.005.031  446X            JZ    MU663       ;IS ZERO; AM DONE
030.371  353          447X            XCHG
030.372  051          448X            DAD   H           ;(D,E) = (DE)*2
030.373  353          449X            XCHG
030.374  322.344.030  450X            JNC   MU661       ;IF NOT OVERFLOW
030.377  361          451X            POP   PSW
031.000  074          452X            INR   A
031.001  365          453X            PUSH  PSW          ;FLAG OVERFLOW
031.002  303.344.030  454X            JMP   MU661       ;PROCESS NEXT BIT
                      455X
```

```
                                                    $MU66

031.005  361         454X  MU663  POP   PSW
031.006  311         457X         RET                 (A,F) = OVERFLOW STATUS.

                     460X  **       $MU86 - MULTIPLY 8X16 UNSIGNED.
                     461X  *
                     462X  *        $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                     463X  *        BIT VALUE.
                     464X  *
                     465X  *        ENTRY   (A) = MULTIPLIER
                     466X  *                (DE) = MULTIPLICAND
                     467X  *                (HL) = RESULT
                     468X  *        EXIT    'Z' SET IF NOT OVERFLOW
                     469X  *
                     470X  *        USES    A,F,H,L
                     471X  *
031.007  041 000 000 472X  $MU86  LXI   H,0           (HL) = RESULT ACCUMULATOR
031.012  305         473X         PUSH  B
031.013  104         474X         MOV   B,H           (B) = OVERFLOW FLAG
031.014  267         475X  MU860  ORA   A             CLEAR CARRY
                     476X  *
031.015  037         477X  MU861  RAR
031.016  322 026 031 478X         JNC   MU862         IF NOT TO ADD
031.021  031         479X         DAD   D
031.022  322 026 031 480X         JNC   MU862         NOT OVERFLOW
031.025  004         481X         INR   B
031.026  267         482X  MU862  ORA   A
031.027  312 044 031 483X         JZ    MU863         IF DONE
031.032  353         484X         XCHG
031.033  051         485X         DAD   H
031.034  353         486X         XCHG
031.035  322 015 031 487X         JNC   MU861         LOOP IF NOT OVERFLOW
031.040  004         488X         INR   B
031.041  303 014 031 489X         JMP   MU860
                     490X  *
031.044  260         491X  MU863  ORA   B             SET *Z* FLAG IF NOT OVERFLOW
031.045  301         492X         POP   B             RESTORE (BC).
031.046  311         493X         RET

                     496X  **       $RSTALL - RESTORE ALL REGISTERS.
                     497X  *
                     498X  *        $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                     499X  *        RETURNS TO THE PREVIOUS CALLER.
                     500X  *
                     501X  *        ENTRY   (SP) = PSW
                     502X  *                (SF+2) = BC
                     503X  *                (SF+4) = DE
                     504X  *                (SF+6) = HL
                     505X  *                (SF+8) = RET
                     506X  *        EXIT    TO *RET*, REGISTERS RESTORED
```

H17ROM - H17 ROM CODE LISTING.
COMMON DECKS.

```
                        507X  *              USES  ALL
                        508X  *
                        509X  *
031.047 361             510X  $RSTALL POP  FSW
031.050 301             511X         POP  B
031.051 321             512X         POP  D
031.052 341             513X         POP  H
031.053 311             514X         RET

                        516X  **  $SAVALL - SAVE ALL REGISTERS ON STACK.
                        517X  *
                        518X  *   $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        519X  *
                        520X  *   ENTRY  NONE
                        521X  *   EXIT   (SP) = FSW
                        522X  *          (SP+2) = BC
                        523X  *          (SP+4) = DE
                        524X  *          (SP+6) = HL
                        525X  *
                        526X  *
                        527X  *   USES   H,L

031.054 343             528X  $SAVALL XTHL          PUSH H, (HL) = RETURN ADDRESS.
031.055 325             529X          PUSH D
031.056 305             530X          PUSH B
031.057 365             531X          PUSH FSW
031.060 351             532X          PCHL          RETURN TO CALLER.

                        535X  **  $TJMP - TABLE JUMP.
                        536X  *
                        537X  *   USAGE
                        538X  *
                        539X  *     CALL $TJMP
                        540X  *     DW   ADDR1        (A) = INDEX
                        541X  *      .               INDEX = 0.
                        542X  *      .
                        543X  *      .
                        544X  *     DW   ADDRN        INDEX = N-1.
                        545X  *   ENTRY  (A) = INDEX.
                        546X  *   EXIT   TO PROCESSOR
                        547X  *          (A) = INDEX*2
                        548X  *   USES   A,F
                        549X  *
                        550X  *
                        551X  *
031.061 007             552X  $TJMP RLC              (A) = INDEX*2
                        553X
031.062 343             554X  $TJMP, EQU   *
031.062 343             555X        XTHL            (HL) = TABLE ADDRESS
031.063 365             556X        PUSH FSW        SAVE INDEX*2.
031.064 315 101 030     557X        CALL $DADA.
```

H17ROM - H17 ROM CODE LISTING
COMMON DECKS

```
031.067 176      558X          MOV     A,M
031.070 043      559X          INX     H
031.071 145      560X          MOV     H,M
031.072 157      561X          MOV     L,A        (A) = INDEX*2
031.073 361      562X          POP     PSW        ADDRESS ON STACK
031.074 343      563X          XTHL               JUMP TF PROCESSOR
031.075 311      564X          RET

                 567X  **   $TBRA -- BRANCH RELATIVE THROUGH TABLE.
                 568X  *
                 569X  *    $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                 570X  *    JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                 571X  *    ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                 572X  *
                 573X  *          CALL    $TBRA
                 574X  *          DB      LAB1-*     INDEX = 0 FOR LAB1
                 575X  *          DB      LAB2-*     INDEX = 1 FOR LAB2
                 576X  *          DB      LABN-*     INDEX = N-1 FOR LABN
                 577X  *
                 578X  *    ENTRY   (A) = INDEX
                 579X  *            (RET) = TABLE FWA
                 580X  *    EXIT    (RET) = COMPUTED ADDRESS
                 581X  *    USES    F,H,L
                 582X  *
                 583X
031.076          584X  $TBRA   EQU     *
031.076 343      585X          XTHL               (HL) = TABLE ADDRESS
031.077 325      586X          PUSH    D
031.100 137      587X          MOV     E,A
031.101 026 000  588X          MVI     D,0
031.103 031      589X          DAD     D
031.104 132      590X          MOV     E,M        (HL) = ADDRESS OF ELEMENT
031.105 031      591X          DAD     D          (HL) = PROCESSOR ADDRESS
031.106 321      592X          POP     D
031.107 343      593X          XTHL
031.110 311      594X          RET

                 597X  **   $TBLS -- TABLE SEARCH.
                 598X  *
                 599X  *    TABLE FORMAT
                 600X  *
                 601X  *          DB      KEY1,VAL1,
                 602X  *          .
                 603X  *          .
                 604X  *          DB      KEYN,VALN,
                 605X  *          DB      0
                 606X  *
                 607X  *    ENTRY   (A) = PATTERN
```

H17ROM - H17 ROM CODE LISTING.
COMMON DECKS.

```
                                        $TBLS

              608X  *           EXIT    (H,L) = TABLE FWA
              609X  *                   (A) = PATTERN IF FOUND
              610X  *                   'Z' SET IF FOUND
              611X  *           USES    A,F,H,L
              612X

              613X
031.111  305  614X  $TBLS   PUSH  B
031.112  107  615X          MOV   B,A
031.113  176  616X  $TBL1   MOV   A,M            (A) = CHARACTER
031.114  270  617X          CMP   B
031.115  312 133 031  618X   JZ   $TBL2          IF MATC
031.120  247  619X          ANA   A
031.121  043  620X          INX   H
031.122  043  621X          INX   H
031.123  302 117 031  622X   JNZ  $TBL1          SKIP FAST
                                                 IF NOT END OF TABLE
031.126  053  623X          DCX   H
031.127  053  624X          DCX   H
031.130  264  625X          ORA   H              CLEAR 'Z'
031.131  076 000  626X      MVI   A,0            SET (A) = 0 FOR OLD USERS
              627X
              628X  *   DONE
              629X  *
031.133  301  630X  $TBL2   POP   B
031.134  043  631X          INX   H
031.135  311  632X          RET


              635X  **        $TYFTX - TYPE TEXT.
              636X  *
              637X  *         $TYFTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
              638X  *
              639X  *         IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
              640X  *         A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
              641X  *
              642X  *         ENTRY   (RET) = TEXT
              643X  *         EXIT    10,(RET+LENGTH)
              644X  *         USES    A,F
              645X  *
031.136  343  647X  $TYFTX  XTHL                 (HL) = TEXT ADDRESS.
031.137  315 144 031  648X  CALL  $TYFTX         TYPE IT
031.142  343  649X          XTHL
031.143  311  650X          RET
              651X
031.144  176  652X  $TYFTX  MOV   A,M
031.145  346 177  653X      ANI   177
031.147  377 002  654X      DB    SYSCALL,,SCOUT
031.151  276  655X          CMP   M
031.152  043  656X          INX   H
031.153  312 144 031  657X  JE    $TYFTX         MORE TO GO.
031.156  311  658X          RET
```

```
661X  **                    ;$UDD - UNPACK DECIMAL DIGITS.
662X  *                     ;
663X  *                     ; UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
664X  *                     ; DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
665X  *                     ;
666X  *                     ;  ENTRY  (B,C) = ADDRESS VALUE
667X  *                     ;         (A)   = DIGIT COUNT
668X  *                     ;         (H,L) = MEMORY ADDRESS
669X  *                     ;  EXIT   (HL)  = (HL) + (A)
670X  *                     ;  USES   ALL
671X  *
672X
673X  $UDD  031.157          $UDD  EQU   *
674X        031.157 315 072 030     CALL  $DADG
675X        031.162 345            PUSH  H          ; SAVE FINAL (H,L) VALUE
676X
677X  UDD1  031.163 365      UDD1  PUSH  PSW
678X        031.164 345            PUSH  H
679X        031.165 021 012 000     LXI   D,10
680X        031.170 315 106 030     CALL  $DU66      ; (H,L) = VALUE/10
681X        031.173 345            PUSH  H
682X        031.174 301            POP   B          ; (B,C) = REMAINDER
683X        031.175 341            POP   H
684X        031.176 076 060         MVI   A,'0'
685X        031.200 203            ADD   E          ; ADD REMAINDER
686X        031.201 053            DCX   H
687X        031.202 167            MOV   M,A        ; STORE DIGIT
688X        031.203 361            POP   PSW
689X        031.204 075            DCR   A
690X        031.205 302 163 031     JNZ   UDD1       ; IF MORE TO GO
691X        031.210 341            POP   H          ; RESTORE H
692X        031.211 311            RET              ; RETURN

695X  **                    ;$ZERO - ZERO MEMORY
696X  *                     ;
697X  *                     ; $ZERO ZEROS A BLOCK OF MEMORY.
698X  *                     ;
699X  *                     ;  ENTRY  (HL) = ADDRESS
700X  *                     ;         (B)  = COUNT
701X  *                     ;  EXIT   (A)  = 0
702X  *                     ;  USES   A,B,F,H,L
703X
704X
705X  $ZERO 031.212 257      $ZERO XRA   A
706X  ZRO1  031.213 167      ZRO1  MOV   M,A
707X        031.214 043            INX   H
708X        031.215 005            DCR   B
709X        031.216 302 213 031     JNZ   ZRO1       ; IF MORE
710X        031.221 311            RET
711
```

INDEX