To be eligible for updates and assistance, you must register your copy of P&T CP/M 2 by filling out and returning the registration card. Please refer to the following serial number should you contact your dealer regarding this software:

2-163- 11737

P&T CP/M® 2 USER'S MANUAL

Corvus Hard Disk Addendum

PICKLES & TROUT®

P&T CP/M® 2

for the

TRS-80 Models II, 12, and 16
User's Manual

Corvus Hard Disk Addendum

Published by
Pickles & Trout
P.O. Box 1206
Goleta, California, 93116
U.S.A.

10 9 8 7 6 5 4 3 2

IMPORTANT NOTE

You should have received P&T CP/M 2 on a diskette with a label like the one shown below. If you did not receive such a diskette, you may not have received a valid copy of the system. Please contact Pickles & Trout at once. All P&T CP/M 2 labels are printed in green ink on white paper.

If you purchased a registered user's copy of P&T CP/M 2 you are required to transfer the registration to your name. There is a fee for this transfer. If you fail to transfer the registration you will not receive the Pickles & Trout newsletter, will not be eligible for updates to the system, and will not be able to receive assistance from Pickles & Trout.

DISCLAIMER

The publisher has made a reasonable effort to insure that the computer programs described herein are correct and operate properly and that the information presented in this publication is accurate; however they are sold and licensed without warranties either expressed or implied, including, but not limited to, the implied warranties of merchantibility and fitness for a particular purpose. The publisher is not liable for consequential damages resulting from the use of this product either individually or in concert with other computer programs. Further, the publisher reserves the right to revise this publication and the programs described herein and to make changes from time to time in the contents thereof without obligation of the publisher to notify any person or organization of such revision or changes.

TABLE OF CONTENTS

1.1 Introduction

The Corvus Hard Disk version of P&T CP/M 2 fully supports the use of all Corvus hard disk drives for mass storage on the TRS-80 Models II, 12 and 16 Microcomputers. Up to 4 hard disk drives may be connected to a computer giving a total of up to 80 Mbytes of on-line storage. In addition, P&T CP/M 2 supports the Corvus Constellation, allowing the connection of up to 64 computers to a hard disk drive.

P&T CP/M 2 allows you complete flexibility in assigning the hard disk storage so that you may tailor the system to your needs. You may assign multiple logical drives to each physical hard disk drive. However each logical drive may be no larger than 8192 Kbytes (8,388,608 bytes) which is the maximum logical drive size supported by CP/M 2.

This supplement to the the P&T CP/M 2 User's Manual is designed to provide you with the additional information necessary to use P&T CP/M 2 with a Corvus Hard Disk system.

The hard disk version of P&T CP/M 2 is completely compatible with the floppy dis version. Most programs will not need any changes at all in order to run with the hard disk system.

If you desire assistance from Pickles & Trout it is absolutely necessary that your copy of P&T CP/M 2 be registered. To register your copy you must fill out and return the registration card you received with the original diskette. If you did not receive the card, please call Pickles & Trout at once for instructions on how to register your copy.

Please have your system serial number available when you contact Pickles & Trout regarding your copy of P&T CP/M 2; YOU WILL BE ASKED FOR IT. The serial number appears on the label of the master diskette, on the front of this manual, and is displayed on the console every time the system is reset.

Please realize that we must limit assistance with P&T CP/M 2 to matters concerning the operating system and its supporting utility programs. We would like to help everyone with their programming problems but it is impossible to do so. We will try to help you as much as we can but please do not expect us to give you step-by-step instructions on how to customize a specific software package. In such a case all we can do is give you some general direction and refer you to the appropriate parts of this addendum.

1.2 Files on the Diskette

You should find the following files on your master diskette in addition to the files described in the P&T CP/M 2 User's Manual. If you are upgrading from another version of P&T CP/M 2 you should find these files (and others for actually installing the upgrade) on the diskette you received.

ACCESS.COM — P&T utility program to allow you to alter the access mode of logical drives assigned to a hard disk from the console. See Section 6.2 of this addendum for further information.

BFBACKUP.COM P&T utility program for backing up a disk file which is larger than the capacity of a diskette onto multiple empty diskettes. See Section 6.3 of this addendum for further information.

BFRESTOR.COM P&T utility program for restoring a disk file backed up by BFBACKUP. See Section 6.4 of this addendum for further information.

BKMOUNT.COM P&T utility program that is used in conjunction with the FILEBACK utility program. Typically this program would not be used independently of FILEBACK. See Section 6.5 of this addendum for further information.

CLEARDIR.COM P&T utility program that completely clears the directory area of a logical drive assigned to a hard disk. This utility should be used with care since its action is irreversible. See Section 6.6 of this addendum for further information.

DRIVEMAP.COM P&T Utility program that reports the Corvus block numbers used by each logical drive assigned to a hard disk drive. This information is useful when using the Corvus Mirror program for backing up and restoring information on the hard disk. See Section 6.7 of this addendum for further information.

FILEBACK.COM P&T utility program that backs up all files that are smaller than the capacity of a diskette from a logical drive on a hard disk to multiple empty diskettes. This program reads the directory of a hard disk and generates a series of SUB files to be used with SUBMIT. The SUB files make use of the PIP utility to perform the file transfers. The user is prompted to change diskettes when necessary during the backup process. The principle limitation of this backup technique is that logical drive A cannot be specified as the destination drive. See Section 6.8 of this addendum for further information.

HDCONFIG.COM P&T utility program to configure the disk storage on the system. This program allows you complete freedom in assigning logical drives on the system so that you can tailor the system to your needs. See Chapter 7 of this addendum for further information.

2.1 Conventions of Notation

In general, this addendum uses the same notation as the P&T CP/M 2 User's Manual for P&T CP/M 2.2m. For your convenience, an explanation is included here.

For ease of reference, all page numbers in this manual consist of two numbers. The first refers to the chapter number and the second to the page number within the chapter.

Figures within this manual are numbered in a similar way, but the second number denotes the figure, not the page, within the chapter. For example, Figure 5.8 refers to the eighth figure in the fifth chapter. If it is necessary to indicate a specific line within a figure, a hyphen separates the figure and line numbers (e.g. Line 5.8-12 means line 12 of Figure 5.8). Keep in mind that "Line ..." refers to a line of a figure, not a line of the text.

When numbers are used within the addendum, they should be considered to be decimal (base 10) unless otherwise noted. A hexadecimal number (base 16) is indicated by appending the letter "h" to the number (e.g. 1Ah). A binary number (base 2) is indicated by appending the letter "b" to the number (e.g. 101b). In figures that represent console displays, this convention will not be used if the program that generated the display does not follow it. Every effort has been made to make the figures representing console displays as accurate as possible. The text relating to a figure will specify the base of the numbers displayed if it is not obvious from the context.

In this addendum, the term "Kbyte" (or Kb) is taken to refer to 1024 bytes. The term "Mbyte" (or Mb) is used to refer to 1024 Kbytes (i.e. 1,048,576 bytes).

When it is necessary to refer to one of the named keys on the keyboard, the name of the key is enclosed in angle brackets. For example, <enter> refers to the key on the keyboard labeled "ENTER". If you are instructed to type <enter> at some point, it is expected that you will press the key labeled "ENTER" rather than typing the 7 characters "<", "e", "n", "t", "e", "r", ">".

Control keys and control codes are denoted by the characters "ctl-" followed by a letter and enclosed in angle brackets (e.g. <ctl-A>). In other manuals, control codes are often indicated by a caret or up-arrow immediately preceding the letter (e.g. ^A). The distinction between control keys and control codes is a fine one. This addendum will use the term "control key" to refer to the key that is actually typed to generate a control code. For example, to generate a <ctl-A>, you would type the "A" key on the keyboard while holding down the <ctrl> key; the <ctrl> key functions as a special type of shift key.

The control code is the numeric code that is generated by the keyboard and sent to the computer when a control key is typed. Control codes are sometimes called control characters. Keep in mind that the term "control code" may be used without a reference to the keyboard. For example, some programs may use control codes to perform certain functions, such as manipulating the console display. In this case, the codes are generated by the program and the keyboard is not involved.

Many figures show a dialog between the computer and the user. This technique is used heavily when explaining how to use various utility programs. In these dialogs,

characters displayed on the console by the computer are shown in plain text; characters typed in by the user are shown underlined.

In this addendum, both the terms "diskette" and "disk" are used. "Diskette" refers only to a floppy diskette. An example would be the diskette you originally received. The term "disk" has a general meaning. It can refer to a diskette or to a hard disk.

In this addendum the term "CP/M" is used in referring to general features and capabilities of the CP/M operating system. The term "P&T CP/M 2" refers specifically to the Pickles & Trout adaptation of the CP/M operating system for the TRS-80 Model II/12/16.

It is frequently desirable to distinguish between physical and logical disk drives. A physical drive refers to the actual hardware of a disk drive. Examples of physical drives are diskette drives and hard disk units. CP/M refers to disk storage in terms of logical drives, denoted by the letters A through P.

There is not always a 1-to-1 correspondence between logical and physical drives since several logical drive can be assigned to a single hard disk or diskette drive. In P&T CP/M 2 a logical drive never includes more than one physical drive. The physical diskette drives are referred to by numbers from 0 to 3. Physical diskette drive 0 is the built-in drive on a Model II and the left hand built-in drive on a Model 16 or 12. There is always a physical diskette drive 0 on the system. Physical diskette drive 1 is the first expansion drive on a Model II system and the right hand built-in drive on a Model 12 or 16. Physical diskette drives 2 and 3 are the remaining expansion drives.

The physical hard disk drives are referred to by numbers from 1 to 4. Physical hard disk drive 1 is the primary hard disk unit, physical hard disk drive 2 is the first secondary hard disk unit, etc.

3.1 Getting Started

If you have purchased an upgrade of your copy of P&T CP/M 2, follow the instructions packaged with the diskette for installing the new modules in your module library. If you purchased P&T CP/M 2 as a Corvus hard disk system, these modules are already in the library.

Before beginning to use P&T CP/M 2 you should make a working system diskette. See Chapter 3 the P&T CP/M 2 User's Manual for instructions. The first working system diskette you generate will include drivers for only the floppy drives on your system. In order to access the hard disk drives on your system you must use the MODSEL (see Chapter 7 the P&T CP/M 2 User's Manual) utility program to include the necessary modules in the system.

While it is executing, MODSEL will present you with the names of any "Hard Disk Driver Modules" that are available for you to use. These are the modules that contain the I/O routines that actually communicate with the hard disk. You should choose a Corvus hard disk driver module.

If you do not select a hard disk driver module for inclusion in the system, the system will be configured just like a standard floppy only system. This option can be useful to keep your system running if your hard disk drive goes in for repair.

The module named CORV1 is designed to be used with the older type A (or rev A) drives. The Module named CORV2 is designed to be used with the newer types B and H (or revs B and H) drives. Although CORV1 will work with the newer drives, CORV2 is more efficient and will provide better system performance. If there are no hard disk modules listed, either you do not have a hard disk version of the system or, if you are upgrading, you have not installed the new modules in your module library.

If you choose a hard disk driver module, MODSEL will also ask you to choose a "Disk Table Module". It is this module that actually defines how the disk drives on your system are used. We have supplied with the system a number of these modules for typical disk configurations. Refer to Appendix A of this addendum for a description of these modules. If one of the supplied modules provides a configuration that meets your need you need merely select it.

If none of the supplied disk table modules seems to be quite right, you can create your own using the HDCONFIG utility program. Chapter 7 of this addendum discusses in detail the considerations you should be aware of before creating your own disk table module. You should read Chapter 7 before custom configuring your system.

After you have used MODSEL to select the hard disk related modules (and any other modules you wish to include in the system) the procedure for configuring a working system diskette is exactly the same as for a floppy only system. See the P&T CP/M 2 User's Manual for details.

The Corvus disk drive may come with data recorded on it that appears to be valid directory information to P&T CP/M 2. If, when you begin to use the hard disk, you get error messages that indicate that the disk or directory is full, you will need to use the CLEARDIR program to erase the spurious information in the directories of

logical drives on the hard disk. See Section 6.6 of this addendum for information about using CLEARDIR.

NOTE: The Corvus disk drives are delivered formatted and ready to run. In most cases, you will never need to format the disk. If for some reason you do need to format the disk, the format operation is performed by the Corvus CDIAGNOS program. Be sure to consult the Corvus documentation carefully before formatting a drive.

4.1 Differences From the Floppy Version

The Corvus Hard Disk version of P&T CP/M 2 is very similar to the standard floppy version. Virtually all programs that run with the floppy version will also run with the Corvus Hard Disk version. There are a few minor differences which are described below.

1. The Corvus Hard Disk system allows you complete freedom in assigning logical drives. For example, you may have logical drives A, B, and C assigned to the hard disk and logical drives D and E assigned to diskette drives. However, this may lead to confusion in some of the diskette oriented utility programs (FORMAT, DISKTEST, etc.) when you are asked to specify the diskette disk drive on which some operation is to take place.

   In order to alleviate the confusion, all of these utility programs allow you to specify diskette drives by either the logical drive letter assigned to the drive or the physical drive number. For a Model II the built-in drive is physical drive 0 and the expansion drives are physical drives 1 through 3. For a Model 12 or 16, the two built in drives are physical drives 0 and 1 while the expansion drives are physical drives 2 and 3.

2. Several new utility programs oriented toward the hard disk have been added to the system. These programs have to do with configuring the system and backing up and restoring data on the hard disk. These programs are described in Chapter 6 of this addendum.

4. A new system feature has been added that allows you to limit access to any logical drives assigned to a hard disk. This feature allows you to write protect individual logical drives on a hard disk without write protecting the entire disk. This feature can also be used to implement a password scheme for gaining access to logical drives on the hard disk. The access limitation function of the system is described in Chapter 5. The ACCESS utility program, described in Section 6.2 provides a means of modifying the drive access from the console.

5. The system MENU function DP cannot set the number of floppy drives on a hard disk system. The number of drives is determined by the drive parameter table module that is selected for inclusion in the system (see Chapter 3 of this addendum and Chapter 6 of the P&T CP/M 2 User's Manual). You may create your own drive parameter table modules using the HDCONFIG program as described in Chapter 7 of this addendum.

NOTES

5.1 Limiting Drive Access

The Corvus Hard Disk version of P&T CP/M 2 includes a method of limiting access to the logical drives that are assigned to the hard disk. There are two forms of access limitation available: access to the drive can be completely denied or access can be limited to reading only.

The type of access to a drive may be modified by a program using Special System Functions 27 and 28 (see Chapter 16 of the P&T CP/M 2 User's Manual). In addition, the system configuration program, HDCONFIG, allows you to specify the type of access to be permitted to each logical drive when the system is cold booted. See Chapter 7 of this addendum for details of configuring the system with the HDCONFIG utility program.

The utility program ACCESS provides a way to change the type of access to a logical drive on a hard disk interactively from the system console. See Section 6.2 of this addendum for information on using ACCESS.

The ability to completely deny access to a logical drive can be used to provide regulated access to a disk drive. For example, suppose the system is configured with 4 logical drives assigned to a hard disk. Each of these 4 drives might be used for different types of programs. Bookkeeping programs could be on one of the drives, customer records on another, inventory management on a third, and program development on the fourth. The system could be configured so that all of these logical drives are not accessible when the system first comes up; the user would be initially limited to using the diskette drives.

A program could then be written that would allow a user to change the status of one or more of the logical drives so that it becomes accessible. This program could include some sort of security measures such as requiring passwords or other identification before allowing access to a logical drive. Note that passwords and other security measures are not part of the system; they are implemented by a program of some sort. It is possible for anyone with a knowledge of the system to write a program (or use the ACCESS utility program) to gain access to any logical drive to which access is blocked; hence this scheme cannot be considered to provide high level security.

Access control can be particularly important when running a Corvus system with multiple users connected to the same hard disk drive. With appropriate programming, the access control facility of P&T CP/M 2 can be used to insure that different users on the system do not interfere with each other. See Chapter 9 for a discussion of multiple user access to a hard disk drive.

If a user or a program attempts to gain access to a logical drive to which access is denied, the error message given in Figure 5.1 will be given. The "xx" in the message will be replaced by a hexadecimal representation (00=A, 01=B,...0F=P) of the logical drive to which access was attempted. You may use the ERROR program to get an explanation of the error code if you wish. The most typical result of this error is that any running program will abort and the user will be returned to the system command level.

```
Hard disk lock out error, code = HLxx
```

Figure 5.1 Error Message for Drive Access Violation

NOTE: Logical drive A **must** be **accessible** when the system is cold booted. If logical drive A is not made accessible, the system will not run because the first thing it does after a cold boot is to try to log onto logical drive A. Logical drive A may be made initially "read only", however.

Access to a logical drive on a hard disk can also be limited to reading only. This feature of the system can be used to write protect individual logical drives on a hard disk. This feature can be particularly useful while doing program development. The logical drives that should not be written to by the program can be set to "read only" status so that any attempt by the program to write to the drive will be trapped and an error message given.

If a logical drive is set for "read only" access, any attempt to write to that drive will result in the error message shown in Figure 5.2. The most typical result of either write protect error message is that any program that is running will be aborted and the user will be returned to the system command level.

```
Hard disk write prot error, code = HP00
```

Figure 5.2 Error Message for Software Write Protect

6.1 Introduction

This section describes the additional utility programs (listed below) that are supplied with the Corvus Hard Disk version of P&T CP/M 2. These programs provide additional functions needed to use the hard disk version of P&T CP/M 2.

page	program name	purpose
6.2	ACCESS	To allow the user to change the type of access permitted to logical drives on a hard disk from the system console.
6.6	BFBACKUP	To back up a file which is larger than the capacity of a single diskette to multiple empty diskettes.
6.14	BFRESTOR	To restore a file that was backed up by BFBACKUP.
6.25	BKMOUNT	Provides a means of changing diskettes when used with the FILEBACK utility program. This program is not normally used by itself.
6.26	CLEARDIR	To wipe clean the directory area of a logical drive on a hard disk. this program is typically used on a new Corvus drive to clear out the data pattern left on the disk when it was formatted.
6.28	DRIVEMAP	To report the Corvus disk blocks assigned to each logical drive which has been defined on a hard disk.
6.30	FILEBACK	To generate SUBMIT files that will backup all files from a given logical drive to multiple empty diskettes. Note that none of the files may be larger than the total capacity of an empty diskette.
6.35	HDCONFIG	To configure the disk storage allocation of the hard disk system.

6.2 Utility name: ACCESS

Purpose: To allow the user to change the access mode of a logical drive defined on a hard disk from the console.

General Description

The ACCESS utility program allows the user to change the access mode of logical drives assigned to a hard disk from the console. ACCESS has two modes of operation, the interactive mode and the command line mode.

In the interactive mode, ACCESS displays the current access mode of each logical drive defined for the system and allows you to make changes at will. The logical drives that are assigned to diskette drives are also displayed even though there is no access control for them.

In the command line mode, all information about changing the access mode of various drives is given on the command line that executes ACCESS. ACCESS reads this information and makes the appropriate changes, reporting each change on the console as it is made.

Note that the changes made by ACCESS remain in effect only until they are changed by using ACCESS again or until a cold boot (RESET) occurs. The access mode for various drives can also be changed by other programs using the Special System Functions provided for that purpose. If you want the system to cold boot with a particular access modes in effect, you must use the HDCONFIG utility program to reconfigure the system to that configuration.

Using ACCESS - Interactive Mode

At any time while ACCESS is running you may press the <break> key to return to the operating system. If you press the <break> key, ACCESS will ask you if you really want to quit. If you respond affirmatively, you will be returned to the system. If you respond negatively, you will be returned to the place at which you pressed <break>.

Figure 6.1 shows the command line for executing ACCESS in the prompted mode.

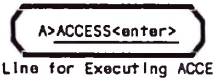

Figure 6.1 Command Line for Executing ACCESS in Prompted Mode

Figure 6.2 shows the basic display of ACCESS. Lines 6.2-4 to 6.2-19 report the current access mode of each logical drive on the system. Note that diskette drives are also reported even though there is no access control for them. On Line 6.2-22 you are asked to enter the logical drive letter for a drive whose access mode you want to change or "0" if you want to return to the operating system.

```
 1:          P&T CP/M 2 Hard Disk System Drive Access Manager - ver 1.xx
 2:                       Copyright 1982 by Pickles & Trout
 3:
 4:                              A - read/write
 5:                              B - read/write
 6:                              C - read/write
 7:                              D - floppy
 8:                              E - floppy
 9:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:          Enter logical drive letter (A-P) to change (0 to quit):
23:
24:
```

Figure 6.2 Console Display of ACCESS in Prompted Mode

Figure 6.3 shows the console dialog for changing logical drive A to "read only" mode.

```
Enter logical drive letter (A-P) to change (0 to quit): A<enter>
        Enter type of access ( NONE, RO, RW ): RO<enter>
```

Figure 6.3 Changing Drive A to "read only" Access

Figure 6.4 shows the console display after logical drive A has been changed to "read only" and logical drive C has been changed to "no access".

```
 1:          P&T CP/M 2 Hard Disk System Drive Access Manager - ver 1.xx
 2:                       Copyright 1982 by Pickles & Trout
 3:
 4:                              A - read only
 5:                              B - read/write
 6:                              C - no access
 7:                              D - floppy
 8:                              E - floppy
 9:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:          Enter logical drive letter (A-P) to change (0 to quit):
23:
24:
```

Figure 6.4 Console Display After Changing Access Mode of Drives A and C

Using ACCESS - Command Line Mode

ACCESS also has a command line mode that allows you to change the access mode of several drives with one command line. This mode also allows SUBMIT files to change drive access modes if necessary. When using the command line mode, you specify the drives to be changed by typing the drive letter followed by an equals sign followed by one of three access modes (RO, RW, or NONE). For example the string "B=NONE" will set logical drive B to "no access". Similarly, "B=RO" will set logical drive B to "read only". Figure 6.5 shows an example of using the command line mode of ACCESS.

```
A>ACCESS A=RO B=RO C=RW<enter>
logical drive A set to read only
logical drive B set to read only
logical drive C set to read/write

A>
```

Figure 6.5 Executing ACCESS in the Command Line Mode

As shown in Figure 6.5, ACCESS reports each change in drive access mode as it is made. This assures you that ACCESS actually made the changes you wanted. If a change you wanted made is not reported, either ACCESS could not understand the command line you typed or you asked for an illegal operation. Examples of illegal operations are specifying a logical drive that is not defined for the system, specifying a logical drive assigned to a diskette drive, or trying to set logical drive A to "no access".

Other examples of command lines that could be used with ACCESS are:

ACCESS E=NONE
Makes logical drive inaccessible.

ACCESS B=RO D=RW F=RW
Makes logical drive B accessible for reading only and logical drives D and F accessible for both reading and writing.

Possible Error Messages

Not a valid response, please re-enter
This message is displayed either alone or with other messages if the response you give to a question is not among the acceptable responses. Frequently the question asked will indicate what responses are considered acceptable. An example of a case where this message would be given is entering "Q" when prompted for a logical drive letter.

Please respond with A-P or 0 only
This message indicates that you did not enter one of the acceptable responses to the request for a logical drive letter.

That logical drive is not on the system
This message is displayed if you enter the letter of a logical drive that is not defined for the system. The drives that are defined are displayed on the console by ACCESS.

There is no access control for diskette drives
This message indicates that you are trying to change the access mode of a diskette drive. The system does not support access control for diskette drives.

Logical drive A must always be accessible
This message indicates that you have tried to set logical drive A to "no access". The system requires logical that drive A always be accessible in order to function. Logical drive A may be set to either "read only" or "read/write", however.

Please respond with "NONE", "RO" or "RW" only
This message is displayed if the response you made to the query for the type of access for a logical drive could not be understood by ACCESS. Use only the responses listed.

You may not make the current drive inaccessible
ACCESS will return to the command mode of the system with the current drive unchanged. If drive B was the current drive when you executed ACCESS, it will still be the current drive when ACCESS is finished. Since the system must always have access to the current drive, ACCESS does not allow you to set the current drive to "no access". You may, however, set the current drive to "read only" access.

6.3 Utility name: BFBACKUP

Purpose: **To back up a file that is larger than the capacity of a diskette to multiple empty diskettes.**

General Description

Normal file transfer programs like PIP and FASTCOPY will work only when there is sufficient space on the destination drive for the entire file being transferred. BFBACKUP provides a means of backing up a large file to multiple diskettes. Sequence information is transferred along with the file to insure that the file is properly restored (by BFRESTOR). BFBACKUP will optionally perform a read back verification of the data after it is written to the diskette to insure data integrity.

BFBACKUP has two modes of operation. In the interactive mode, BFBACKUP prompts for all information regarding source and destination drives and file names. In the command line mode the source and destination drives, the file name, and whether or not to verify the output files are specified on the command line that executes BFBACKUP. The command line option allows you to set up SUBMIT files that automatically initiate the backup of a certain file.

BFBACKUP is intended only for backing up large files. It should not be used if the entire file will fit on a single diskette. NOTE: The parts of the backed up file that are stored on the diskettes will typically not be of use individually. It is necessary to use BFRESTOR to recreate a copy of the original file before it can be used.

Using BFBACKUP - Interactive Mode

At any time while BFBACKUP is running you may press the <break> key to return to the operating system. After you press <break>, BFBACKUP will ask you if you really want to quit. If you respond affirmatively, you will be returned to the command level of the system. If you respond negatively, you will be returned to the point at which you pressed <break>.

Similarly you may press the <F1> key to start over at the beginning of BFBACKUP. After you press <F1>, BFBACKUP will ask you if you really want to start over. If you respond affirmatively, BFBACKUP will start over just as if you had re-executed it. If you respond negatively, BFBACKUP will return you to the point at which you pressed <F1>.

Any time BFBACKUP asks a question requiring a "yes/no" answer, it will accept "Y", "y", "T", "t", and "1" as affirmative responses. "N", "n", "F", "f", and "0" will be accepted as negative responses. All other responses are not valid.

Figure 6.6 shows the command line that will execute BFBACKUP in the prompted mode.

```
A>BFBACKUP<enter>
```

Figure 6.6 Command Line to Execute BFBACKUP In Prompted Mode

The initial console dialog is shown in Figure 6.7. BFBACKUP first asks you for the drive from which a file is to be transferred (Line 6.7-9). In this example, logical drive C is specified. You are then asked for the destination drive (Line 6.7-11). In this example, logical drive D is specified. On Line 6.7-15, you are asked to specify the file to be backed up. At this point you may press <ctl-D> (as shown) to display the directory of the source drive.

```
 1:            Pickles & Trout  Big-File  Backup Utility  -  ver 1.xxx
 2:
 3:                    ( Hit <BREAK> to quit, <F1> to restart )
 4:
 5:
 6:
 7:
 8:
 9:  Enter Source drive . . . . . (A-P): C<enter>
10:
11:  Enter floppy-disk Destination drive  (A-P): D<enter>
12:
13:  Enter name of file to back up
14:   - or -
15:  Press  CTRL-D  to  view  Source  directory: <ctl-D>
16:
```

Figure 6.7 Initial Console Dialog of BFBACKUP In Prompted Mode

If you request that the directory of the source drive be displayed, it will be displayed in the form shown in Figure 6.8. The directory listing will include all system (or hidden files). These are files that have the "SYS" attribute set and hence are not normally displayed in a directory listing. If there are more files than can be displayed on the console at one time, Line 6.8-22 will have the message "More directory entries follow . . . .". More entries will be displayed after you press <enter>. This will continue until all of the directory entries have been displayed.

```
 1:  Directory (including "hidden" files) . . . .
 2:
 3:  C:  ASM      COM  :  CRT      DEF  :  DDT      COM  :  DUMP     ASM
 4:  C:  DUMP     COM  :  ED       COM  :       .       :       .
 5:  C:       .        :       .        :       .       :       .
 6:  C:       .        :       .        :       .       :       .
 7:  C:       .        :       .        :       .       :       .
 8:  C:       .        :       .        :       .       :       .
 9:  C:       .        :       .        :       .       :       .
10:  C:       .        :       .        :       .       :       .
11:  C:       .        :       .        :       .       :       .
12:  C:       .        :       .        :       .       :       .
13:  C:       .        :       .        :       .       :       .
14:  C:       .        :       .        :       .       :       .
15:  C:       .        :       .        :       .       :       .
16:  C:  XSUB     COM  :  DATIME   COM  :  AUTOEXEC COM  :  CLEARDIR COM
17:  C:  DOCHECK  COM  :  BIGFILE  EXT  :
18:
19:
20:
21:
22:                          End of directory listing
23:                       (press <enter> to continue)
```

Figure 6.8 Display of Source Disk Directory from BFBACKUP

After the last group of directory entries is displayed and you press <enter>, you will be returned to the initial console dialog as shown in Figure 6.9. You may then enter the name of the file to be transferred on Line 6.9-15. You may include a drive designation in the file name if you wish but, if you do so, it must match the source drive entered on Line 6.9-9. After you have entered the file name you will be asked if you want each diskette verified after it is written. Verification will cause the backup to take about twice the time of an unverified backup but it will give you maximum assurance that the file was backed up accurately. In the example, verification is requested.

```
 1:          Pickles & Trout  Big-File  Backup Utility  -  ver 1.xxx
 2:
 3:                  ( Hit <BREAK> to quit, <F1> to restart )
 4:
 5:
 6:
 7:
 8:
 9:  Enter Source drive . . . . . (A-P): C
10:
11:  Enter floppy-disk Destination drive  (A-P): D
12:
13:  Enter name of file to back up
14:    - or -
15:  Press  CTRL-D  to  view  Source  directory: BIGFILE.EXT<enter>
16:
17:  Do you want each disk to be
18:   verified after having been written? (Y/N): Y<enter>
19:
```

Figure 6.9 Entering the Name of the File to Backup

Once you have finished the initial dialog, BFBACKUP will prompt you to mount an empty diskette on the destination drive as shown in Figure 6.10.

```
Insert a fresh disk in drive  D  and press <ENTER>: <enter>
```

Figure 6.10 Prompt for Mounting a New Diskette

Once you have mounted an empty diskette and pressed <enter>, BFBACKUP will begin the backup operation as shown in Figure 6.11. It first shows you how many diskettes you will need for the entire backup (Line 6.11-1) and then starts writing to the diskette. On Line 6.11-6 BFBACKUP keeps a running tally of how much of the file has been backed up.

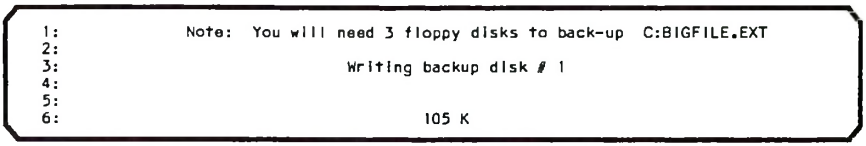

Figure 6.11 Writing to the First Diskette

After the first disk is entirely written, it will be read back and verified if you requested verification. The display during the verification pass is shown in Figure 6.12. BFBACKUP keeps a running tally of how much of the file has been verified on Line 6.12-5. After the entire diskette is verified the message on Line 6.12-7 is displayed to indicate that the diskette is finished and BFBACKUP waits for you to press <enter>.

Figure 6.12 Verifying the Data Written to the First Diskette

If, while making the verification pass, BFBACKUP detects an error, it will display the message shown in Figure 6.13.

Figure 6.13 Reporting a Verify Error

After you press <enter> in response to the message of Figure 6.13, BFBACKUP presents you with the options shown in Figure 6.14. You may retry writing the file on the same diskette or you may try another diskette. If you retry the same diskette, BFBACKUP will return to the display of Figure 6.11. If you elect to try another diskette, BFBACKUP will return to the display of Figure 6.10.

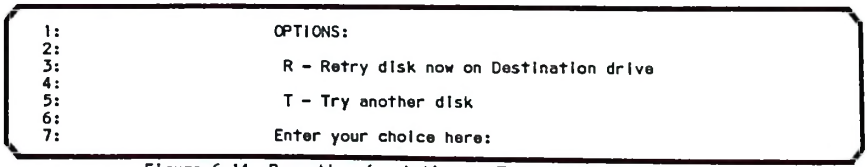

Figure 6.14 Prompting for Action to Take for a Verify Error

After the successful completion of a diskette, BFBACKUP will display the message shown in Figure 6.15 and wait for you to mount another diskette and press <enter>.

```
1:                    Remove disk from drive D.
2:
3:                 Insert a fresh disk in it's place.
4:
5:                    Press <ENTER> when ready: <enter>
```

Figure 6.15 Prompting for Another Diskette

If you do not change diskettes as requested in Figure 6.15 or if you should accidentally mount a diskette that had already been used in the backup operation, BFBACKUP will display the message shown in Figure 6.16.

```
1:              >>>> Disk on drive D is a completed backup disk.
2:
3:           Please remove it and insert a fresh disk in it's place.
4:
5:                    Press <ENTER> when ready: <enter>
```

Figure 6.16 Console Display If Diskette Is Not Changed

Once a new diskette is mounted, BFBACKUP will begin to write to it. While it is writing, it will display the message shown in Figure 6.17. Once again, BFBACKUP keeps a running tally of how much of the source file has been backed up on Line 6.17-5. If the backup is being verified, the display of Figure 6.17 with the word "Writing" replaced by "Verifying" will be shown during the verification pass. Displays of the form shown in Figure 6.17 will be presented for each additional destination diskette until the entire file is backed up.

```
1:                    Backing up file C:BIGFILE.EXT
2:
3:                    Writing backup disk # 2
4:
5:                          635 K
```

Figure 6.17 Console Display While Writing to the Second and Following Diskettes

If you should mount a diskette containing files when BFBACKUP asks for a new diskette, it will present the display shown in Figure 6.18. At this point you have the option of looking at the directory of the diskette, erasing all files on the diskette and continuing, or trying another diskette. On Line 6.18-12 the option to erase all files on the diskette and continue is selected. A listing of the directory will have the same form shown in Figure 6.8.

```
1:              >>>> Disk on drive D is not completely erased.
2:
3:
4:                   OPTIONS:
5:
6:                    D - Look at Destination disk directory
7:
8:                    K - Erase Destination disk and proceed
9:
10:                   T - Try another disk on Destination drive
11:
12:                  Enter your choice here: K<enter>
```

Figure 6.18 Console Display If the Diskette Is not Empty

Using BFBACKUP - Command Line Mode

The form of the command for the command line mode of BFBACKUP is shown in Figure 6.19. The "s:" would be replaced by the logical drive letter of the source drive. The "d:" would be replaced by the logical drive letter of the destination drive. The logical drive letters may be followed by a colon but it is not required. "filename.ext" would be replaced by the complete file name of the file to be backed up. Note that this file name may include a drive designation but it must agree with the source drive given previously in the command line. The "V" is optional and if present, causes a verification of the output file to be done. If no "V" is present, no verification is done. When BFBACKUP is executed in the command line mode, it begins execution at the point represented by Figure 6.10.

```
A>BFBACKUP s: d: filename.ext V<enter>
```

Figure 6.19 Executing BFBACKUP In the Command Line Mode

If BFBACKUP detects an error on the command line, it will check for the existence of an executing submit file. If it detects one, it will display the message shown in Figure 6.20. Since a command line error results in BFBACKUP not running, you may want to abort the execution of the submit file. If you respond affirmatively to the query in Figure 6.20, the submit file will continue. If you respond negatively, the submit file will be aborted and you will be returned to the command level of the system.

```
Command line error with SUBMIT file in progress.
Do you want to continue with the SUBMIT file? (Y/N) :
```

Figure 6.20 BFBACKUP Command Line Error During a Submit File

The following examples show command lines that could be used with BFBACKUP:

BFBACKUP A D DATA.DBF
Backs up the file DATA.DBF from logical drive A to diskettes on logical drive D. No verification is performed.

BFBACKUP B: F: HUGE.FIL V
Backs up the file HUGE.FIL from logical drive B to diskettes on logical drive F. The data written to each diskette is verified.

Possible Error Messages

Non-modular system, not compatible with this program.
This program has been designed to work with the modular version of P&T CP/M 2. It will not work with previous versions.

Destination drive is a hard disk
You cannot use a hard disk drive as a destination for the backup. Specify a floppy drive.

xxxxxxxx.eee has nothing in it
The file you specified ("xxxxxxxx.eee" will be replaced by the file name) to be copied has no data stored in it, hence it is not appropriate to use BFBACKUP.

Open Error on Source drive
Indicates that an error occurred while attempting to open a file on the source drive. BFBACKUP is restarted after an error of this type.

Open Error on Destination drive
Indicates that an error occurred while attempting to open a file on the destination drive. BFBACKUP is restarted after an error of this type.

Close Error on Source drive
Indicates that an error occurred while attempting to close a file on the source drive. BFBACKUP is restarted after an error of this type.

Close Error on Destination drive
Indicates that an error occurred while attempting to close a file on the destination drive. BFBACKUP is restarted after an error of this type.

Read Error on Source drive
Indicates that a disk error occurred while reading from the source drive. BFBACKUP is restarted after an error of this type.

Read Error on Destination drive
Indicates that a disk error occurred while reading from the destination drive. BFBACKUP is restarted after an error of this type.

Write Error on Source drive
Indicates that a disk error occurred while writing to the source drive. BFBACKUP is restarted after an error of this type.

Write Error on Destination drive
Indicates that a disk error occurred while writing to the destination drive. BFBACKUP is restarted after an error of this type.

Invalid drive name
Indicates that the logical drive letter just entered in the prompted mode is not in the range A - P.

Invalid file name
Indicates that the file name just entered in the prompted mode contains illegal characters.

Drive letter in file name doesn't match specified Source drive
Indicates that the drive designation given with the file name just entered in the prompted mode does not match the source drive previously entered.

Source and Destination cannot be the same drive
Indicates that you have specified the same logical drive for both the source and destination drives. This is not permitted when running BFBACKUP. This message can be given in either the prompted or command line mode.

That drive is not on the system
Indicates that the logical drive just specified for the source or destination drive in the prompted mode is not defined for this system.

Can't find file on Source disk
Indicates that a file with the specified name does not exist on the source drive. Display the directory to check if you are using the right name.

Not a valid response, please re-enter
Indicates that your response is not among the acceptable responses for the question. The acceptable responses are usually indicated in the question.

>>>> Error: Bad Source drive name.
Indicates that the logical drive letter entered on the command line for the source drive was not in the range A - P.

>>>> Error: Bad Destination drive name
Indicates that the logical drive letter entered on the command line for the destination drive was not in the range A - P.

>>>> Error: Bad file name
Indicates that the file name entered on the command line contained illegal characters.

>>>> Error: Drive letter in file name differs from Source drive name
Indicates that the drive designation used with the file name entered on the command line does not match the source drive specified previously on the command line.

>>>> Error: Bad Verify Option specifier
Indicates that characters other than "V" were found following the file name on the command line. Note that this message can be caused by specifying an extension longer than three characters for the file name.

>>>> Error: Source drive not on system
Indicates that the source drive specified on the command line is not defined for this system.

>>>> Error: Destination drive not on system
Indicates that the destination drive specified on the command line is not defined for this system.

>>>> Error: Source and Destination cannot be the same drive
Indicates that you have specified the same logical drive for both the source and destination drives on the command line.

>>>> Error: Destination drive is a hard disk
On the command line, you specified a hard disk drive as the destination for the backup. This is not allowed; specify a floppy drive.

6.4 Utility name: BFRESTOR

Purpose: To restore large files that were previously backed up by BFBACKUP.

General Description

The BFRESTOR utility program restores a file backed up by BFBACKUP. It reads the parts of the file from the diskettes on which it was saved and reassembles a copy of the original file. Checks are made to insure that the parts are reassembled in the proper order. You are allowed to change the name under which the restored file will be stored. BFRESTOR will optionally perform a read back verification of the data after it is written to the destination drive to insure data integrity.

BFRESTOR has two modes of operation: interactive and command line. In the interactive mode, BFRESTOR will enter into a dialog with the user to get information such as which drives to use for the restore operation, whether verification is to be done, etc. In the command line mode, this information is specified on the command line that executes BFRESTOR. This allows you to set up SUBMIT files that automatically initiate the restoration operation.

Note: the BFRESTOR program is only useful for restoring files which were backed up with the BFBACKUP program. It is not a general purpose file transfer utility.

Using BFRESTOR - Interactive Mode

At any time while BFRESTOR is running you may press the <break> key to return to the operating system. After you press <break>, BFRESTOR will ask you if you really want to quit. If you respond affirmatively, you will be returned to the command level of the system. If you respond negatively, you will be returned to the point at which you pressed <break>.

Similarly you may press the <F1> key to start over at the beginning of BFRESTOR. After you press <F1>, BFRESTOR will ask you if you really want to start over. If you respond affirmatively, BFRESTOR will start over just as if you had re-executed it. If you respond negatively, BFRESTOR will return you to the point at which you pressed <F1>.

Any time BFRESTOR asks a question requiring a "yes/no" answer, it will accept "Y", "y", "T", "t", and "1" as affirmative responses. "N", "n", "F", "f", and "0" will be accepted as negative responses. All other responses are not valid.

The CP/M command line shown in Figure 6.21 will execute BFRESTOR in the prompted mode.

```
A>BFRESTOR<enter>
```

Figure 6.21 Command Line to Execute BFRESTOR In Prompted Mode

When BFRESTOR is executed in the prompted mode, it enters into the dialog shown in Figure 6.22. On Line 6.22-7 BFRESTOR requests the logical drive on which the diskettes containing the file to be restored will be mounted. In this example logical

drive D was specified. On Line 6.22-9 BFRESTOR requests the logical drive to which the file will be restored. In this example logical drive C was entered. If you make a mistake in entering either of these drive letters, merely press <F1> to start over. BFRESTOR then asks if you want the file verified after it is restored. If you respond affirmatively, as shown on Line 6.22-12, the restoration operation will take about twice as long as if no verification is done.

```
 1:            Pickles & Trout  Big-File  Restore Utility  -  ver 1.xxx
 2:
 3:                    ( Press <BREAK> to quit, <F1> to restart )
 4:
 5:
 6:
 7:  Enter floppy-disk Source drive . . (A-P): D<enter>
 8:
 9:  Enter hard-disk Destination drive  (A-P): C<enter>
10:
11:  Do you want the file to be
12:   verified while being restored ?   (Y/N): Y<enter>
13:
```

Figure 6.22 Initial Dialog of BFRESTOR in Prompted Mode

After you have completed the initial dialog as shown in Figure 6.22, BFRESTOR will prompt you to mount the first of the backup diskettes (generated by BFBACKUP) on the drive you specified and press <enter> (Figure 6.23).

```
Insert the first backup disk in drive  D  and press <ENTER>: <enter>
```

Figure 6.23 Prompting for the First Diskette

If, in response to the prompt of Figure 6.22, you mount a diskette that was not generated by BFBACKUP, you will be presented with the display shown in Figure 6.24. At this point you have the option of looking at the directory of either the source or destination disks or of trying another source diskette. Viewing the directories can be useful for figuring out what diskette you actually did mount. The format of the directory display is similar to that shown in Figure 6.27 except that the file names fill the entire screen. In Figure 6.24, the user elects to try another diskette. After this option is selected, BFRESTOR returns to the prompt of Figure 6.23.

```
 1:              >>>>  Disk on drive  D  is not a valid backup disk.
 2:
 3:
 4:                    OPTIONS:
 5:
 6:                     S - View Source directory
 7:
 8:                     D - View Destination directory
 9:
10:                     T - Try another disk on Source drive
11:
12:                    Enter your choice here: T<enter>
```

Figure 6.24 Console Display if a Non-Backup Diskette is Mounted

If the diskette you mount was generated by BFBACKUP but is not the first diskette in the series for the file, you will be presented with a display as shown in Figure 6.25. This display tells you the name of the file on the backup diskette and the position of this diskette in the backup series (Line 6.25-1). Since there is nothing of interest in the directory of the backup diskette, the only options available to you

at this time are to look at the directory of the destination disk or try another source diskette. In Figure 6.25 the user elects to try another source diskette. After this option is selected, BFRESTOR returns to the prompt of Figure 6.23.

```
1:        Disk on drive  D  is backup disk # 2  for the file  BIGFILE.EXT
2:
3:
4:                  OPTIONS:
5:
6:                   D - View Destination directory
7:
8:                   T - Try another disk on Source drive
9:
10:                 Enter your choice here: T<enter>
```

Figure 6.25 Display If the Wrong Backup Diskette Is Mounted

If the mounted diskette is the first in the backup series, BFRESTOR proceeds to read it and present the display shown in Figure 6.26. On Line 6.26-6, BFRESTOR reports the name of the file that was backed up onto the diskette. On Line 6.26-8, the total size of the file is displayed and on Line 6.26-9 the amount of available space on the destination drive is displayed. If there is not enough space on the destination drive to restore the entire file, BFRESTOR displays the message "NOT ENOUGH SPACE" as shown on Line 6.26-9. On Line 6.26-11, BFRESTOR shows the name which will be used for the restored file. Note that it defaults to the same name the file had when it was backed up.

IMPORTANT NOTE: BFRESTOR restores the file to a temporary file on the destination drive. When the restore operation is completed, the temporary file is renamed to the correct file name. If another file with the same name exists on the drive, it is deleted just before the rename operation is performed. For the brief period of time after the restoration is complete but before the deletion is done, there are actually two copies of the file on the drive, the old copy and the newly restored copy in the temporary file.

This sequence of operations protects you from losing the original file on the destination drive if the restoration is interrupted before it is complete. This does have a drawback, however. If you are restoring a large file to a drive which already contains a copy of the file, BFRESTOR may tell you that there is not enough space available for the restoration. This occurs when there is not enough space for another copy (the temporary one) of the file. In this case you should delete the old copy from the hard disk before beginning the restoration to make room.

BFRESTOR then presents the options you have at this point (Lines 6.26-13 through 6.26-18) and requests that you enter the option you want on Line 6.26-20. In Figure 6.26 the user selects the option to look at the directory of the destination drive. While viewing the destination drive directory, you may delete files (if you wish) to make room for the file to be restored.

```
1:          Pickles & Trout  Big-File  Restore Utility  -  ver 1.xxx
2:
3:                    ( Press <BREAK> to quit, <F1> to restart )
4:
5:
6:      Disk on drive  D  is the  first  backup disk for the file  BIGFILE.EXT
7:
8:                       Total size of Source file  :  1456 K
9:                       Space on Destination disk C:     0 K  <-- NOT ENOUGH SPACE
10:
11:                        Output file-name:  C:BIGFILE.EXT
12:
13:                     OPTIONS:
14:
15:                      R - Restore file
16:                      C - Change output file name
17:                      D - View Destination directory
18:                      T - Try another disk on Source drive
19:
20:                     Enter your choice here: D<enter>
```

Figure 6.26 Console Display After First Diskette is Mounted

The directory display has the form shown in Figure 6.27. All files in the directory under the current user number are displayed including any files that have the "SYS" attribute set so that they do not show in a standard directory display. Only a limited number of directory entries are displayed at any one time since the rest of the space on the display is needed for other information. The size of the file being restored and the total available space on the destination disk are shown on Lines 6.27-13 and 6.27-14. The available options are displayed on Lines 6.27-17 through 6.27-22.

The "B" option moves you back to the beginning of the directory in case you want to go through it again. The "D" option moves you on to the next set of the directory entries. On Line 6.27-24, the user selects the "D" option to move on to the next set of directory entries.

```
1:  Directory (including "hidden" files) . . . .
2:
3:  C:  ASM      COM  :  CRT      DEF  :  DDT      COM  :  DUMP     ASM
4:  C:  DUMP     COM  :       .        :       .        :       .
5:  C:       .        :       .        :       .        :       .
6:  C:       .        :       .        :       .        :       .
7:  C:       .        :       .        :       .        :       .
8:  C:       .        :       .        :       .        :       .
9:  C:       .        :       .        :  DATIME   COM  :  DDCHECK  COM
10:
11: More directory entries follow . . .
12: ------------------------------------------------------------------------
13:                    Total size of Source file  :  1456 K
14:                    Space on Destination disk C:     0 K  <-- NOT ENOUGH SPACE
15:
16:
17:                     OPTIONS:
18:
19:                      B - Go to beginning of directory
10:                      D - View more directory entries
21:                      X - Delete a file
22:                      E - Exit directory listing
23:
24:                     Enter your choice here: D<enter>
```

Figure 6.27 Displaying the Destination Directory

The last set of directory entries is displayed as shown in Figure 6.28. Note that there are only 3 options available at this point since the "D" option (to display

more entries) has no meaning at this point. On Line 6.28-23 the user selects option "X" to delete a file. It is not necessary that the name of the file to be deleted appear on the console at the time the "X" option is selected nor must you be at the end of the directory to select the "X" option.

```
 1:  Directory (including "hidden" files) . . . .
 2:
 3:  C:  SETCCB    COM  :  FILE1    DAT  :  FILE2    DAT  :  FILE3    DAT
 4:  C:       .         :       .       :       .       :       .
 5:  C:       .         :       .       :       .       :       .
 6:  C:       .         :  BIGFILE  EXT  :
 7:
 8:  End of directory listing.
 9:  ------------------------------------------------------------------------
10:
11:
12:
13:                     Total size of Source file  :  1456 K
14:                     Space on Destination disk C:     0 K  <-- NOT ENOUGH SPACE
15:
16:
17:                       OPTIONS:
18:
19:                        B - Go to beginning of directory
20:                        X - Delete a file
21:                        E - Exit directory listing
22:
23:                       Enter your choice here: X<enter>
24:
```

Figure 6.28 Display of Last Set of Directory Entries

After the "X" option is selected, BFRESTOR asks you for the name of the file to be deleted as shown in Figure 6.29. On Line 6.29-17 you may just press <enter> to return to the previous display without erasing a file or you may enter a file name as shown.

```
 1:  Directory (including "hidden" files) . . . .
 2:
 3:  C:  SETCCB    COM  :  FILE1    DAT  :  FILE2    DAT  :  FILE3    DAT
 4:  C:       .         :       .       :       .       :       .
 5:  C:       .         :       .       :       .       :       .
 6:  C:       .         :  BIGFILE  EXT  :
 7:
 8:  End of directory listing.
 9:  ------------------------------------------------------------------------
10:                     Total size of Source file  :  1456 K
11:                     Space on Destination disk C:     0 K  <-- NOT ENOUGH SPACE
12:
13:
14:                       Enter name of file to delete
15:                       or just press <ENTER> to skip.
16:
17:                       File to delete:  BIGFILE.EXT<enter>
18:
```

Figure 6.29 Deleting a File

After you enter the name of the file to delete, BFRESTOR will try to erase it. If it is successful, it will return to the previous display with the menu of options. If the file does not exist, BFRESTOR will display an error message and ask for the file name again. If the File is set to "read only" status, BFRESTOR will present the display shown in Figure 6.30. You are informed that the file you specified is "read only" (Line 6.30-14) and asks you if you actually want it deleted (Line 6.30-16). If you respond affirmatively, BFRESTOR will delete the file and return to the menu

of options. If you respond negatively, BFRESTOR will take no action and return to the menu of options. In Figure 6.30, the user elects to erase the file.

```
1:  Directory (including "hidden" files) . . . .
2:
3:  C:  SETCCB    COM  :  FILE1    DAT  :  FILE2    DAT  :  FILE3    DAT
4:  C:       .         :       .       :       .       :       .
5:  C:       .         :       .       :       .       :       .
6:  C:       .         :  BIGFILE  EXT  :
7:
8:  End of directory listing.
9:  ------------------------------------------------------------------------------
10:                    Total size of Source file  :  1456 K
11:                    Space on Destination disk C:     0 K  <-- NOT ENOUGH SPACE
12:
13:
14:                      C:BIGFILE.EXT  is set to Read/Only status.
15:
16:                      Do you wish to delete it? (Y/N): Y<enter>
17:
```

Figure 6.30 Display if You Try to Delete a Read/Only File

As files are deleted, the display of available space on the destination disk is changed to reflect the space freed by the deletion. You may delete as many files as you wish to make enough space available to receive the file being restored. Note that after a file is deleted, you are returned to the beginning of the directory. Figure 6.31 shows the last directory display after just enough files have been deleted to make room for the file being restored.

```
1:  Directory (including "hidden" files) . . . .
2:
3:  C:  SETCCB    COM  :  FILE1    DAT  :  FILE2    DAT  :  FILE3    DAT
4:  C:       .         :       .       :       .       :       .
5:  C:       .         :       .       :       .       :       .
6:  C:       .         :
7:
8:  End of directory listing.
9:  ------------------------------------------------------------------------------
10:
11:
12:
13:                    Total size of Source file  :  1456 K
14:                    Space on Destination disk C:  1456 K
15:
16:
17:                      OPTIONS:
18:
19:                       B - Go to beginning of directory
20:                       X - Delete a file
21:                       E - Exit directory listing
22:
23:                       Enter your choice here: E<enter>
24:
```

Figure 6.31 Console Display After Deleting Files to Make Room

On Line 6.31-23, the user selects the "E" option to leave the directory display and return to the display of Figure 6.26. At this time you may decide that you want to restore the file under another name. To do this the "C" option in Figure 6.26 is selected. After selecting this option, BFRESTOR will prompt you for a new output file name as shown in Figure 6.32. You may change the output file name as many times as you like before beginning the restore operation.

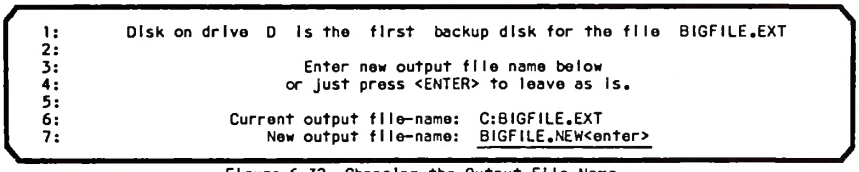

Figure 6.32 Changing the Output File Name

When the "R" operation is selected on Line 6.26-20, the restore operation is started. While the restoration is taking place, BFRESTOR presents a console display as shown in Figure 6.33. On Line 6.33-5 a running tally of the amount of data that has been transferred is displayed.

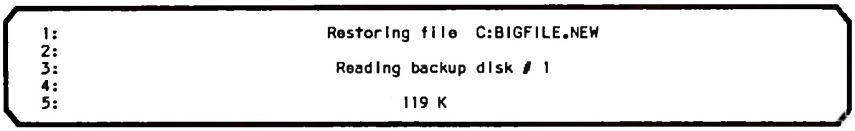

Figure 6.33 Restoring from the First Diskette

If you have requested a verification, BFRESTOR will make a second pass through the data that was transferred from this source diskette to check that it was copied accurately. While this verification is taking place, the display will look like Figure 6.34 up to Line 6.34-6. On Line 6.34-5 a running tally of the amount of the file that has been verified is displayed. After all operations for this diskette have been completed, the message shown on Line 6.34-7 is displayed.

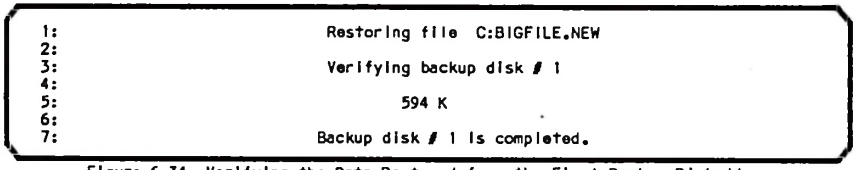

Figure 6.34 Verifying the Data Restored from the First Backup Diskette

You will be prompted for each diskette in the backup series with a display similar to that shown in Figure 6.23. After the last diskette in the series is completed, a display like that shown in Figure 6.35 will be presented. After acknowledging the display by pressing <enter>, BFRESTOR will start over again at the beginning. You may restore as many files as you wish and then press <break> to return to the command level of the system.

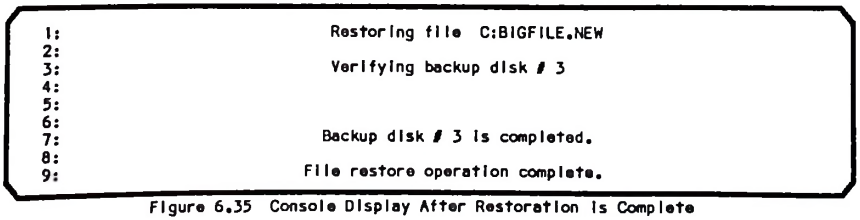

Figure 6.35 Console Display After Restoration is Complete

Using BFRESTOR - Command Line Mode

In the command line mode of operation, all of the information regarding the source and destination drives and file names is specified on the command line that executes BFRESTOR. Figure 6.36 shows the form of such a command line. The "s:" would be replaced with the logical drive letter of the source drive. Similarly the "d:" would be replaced by the logical drive letter of the destination drive. "filename.ext" is optional and, if present, will be used as the name for the output file. If it is not present, the file will be restored under the same name it had when it was backed up. The "V" is also optional. If it is present, a verification pass will be done otherwise no verification pass will be performed.

Figure 6.36 Command Line to Execute BFRESTOR In Command Line Mode

When the command line mode of BFRESTOR is used, the program begins at the point represented by Figure 6.23. When the restoration is completed, BFRESTOR will return immediately to the operating system if it was executed in the command line mode.

The following examples show command lines that could be used with BFRESTOR:

BFRESTOR D: C:
Restores a file previously backed up by BFBACKUP from diskettes to be mounted on drive D to drive C. The restored file will have the same name it had when it was backed up. No verification is done.

BFRESTOR E: A: DATABASE.BIG V
Restores a file previously backed up by BFBACKUP from diskettes to be mounted on drive E to drive A. The restored file will be named DATABASE.BIG regardless of the name it originally had. Read back verification is performed on the restored file.

Possible Error Messages

Non-modular system, not compatible with this program.
BFRESTOR is designed to work only with P&T CP/M 2.2m and subsequent releases. You have attempted to run the program on an earlier release.

Open Error on Source drive
Indicates that an error occurred while attempting to open a file on the source drive. BFRESTOR is restarted after an error of this type.

Open Error on Destination drive
Indicates that an error occurred while attempting to open a file on the destination drive. BFRESTOR is restarted after an error of this type.

Close Error on Source drive
Indicates that an error occurred while attempting to close a file on the source drive. BFRESTOR is restarted after an error of this type.

Close Error on Destination drive
Indicates that an error occurred while attempting to close a file on the destination drive. BFRESTOR is restarted after an error of this type.

Read Error on Source drive
Indicates that a disk error occurred while reading from the source drive. BFRESTOR is restarted after an error of this type.

Read Error on Destination drive
Indicates that a disk error occurred while reading from the destination drive. BFRESTOR is restarted after an error of this type.

Write Error on Source drive
Indicates that a disk error occurred while writing to the source drive. BFRESTOR is restarted after an error of this type.

Write Error on Destination drive
Indicates that a disk error occurred while writing to the destination drive. BFRESTOR is restarted after an error of this type.

Invalid drive name
Indicates that the logical drive letter just entered in the prompted mode is not in the range A - P.

Invalid file name
Indicates that the file name just entered in the prompted mode contains illegal characters.

Drive letter in file name differs from Destination drive name
Indicates that the drive designation given with the file name just entered in the prompted mode does not match the destination drive previously entered. This message may also be displayed in the command line mode if the drive designation given with the new destination file name does not match the logical drive letter entered for the destination drive.

Source and Destination cannot be the same drive
Indicates that you have specified the same logical drive for both the source and destination drives. This is not permitted when running BFRESTOR. This message can be given in either the prompted or command line mode.

That drive is not on the system
Indicates that the logical drive just specified for the source or destination drive in the prompted mode is not defined for this system.

Source drive is a hard disk.
Indicates that the source drive you have specified is a hard disk drive. BFRESTOR restores from diskettes only hence this is not allowed.

Not a valid response, please re-enter
Indicates that the response you entered is not among the acceptable responses for the question asked. The acceptable responses are usually indicated in the question.

***** VERIFY ERROR *****
This message is displayed if an error is discovered while doing the verification pass. After a verify error occurs, you are given the option of retrying the

restore operation starting with the current diskette in the backup series or aborting the restoration.

Disk on Source drive is an incomplete backup disk
An incomplete backup diskette can be caused by aborting the backup operation as the diskette was being generated. BFRESTOR detects this condition and displays this message. It typically means that you mounted the wrong diskette and should try another one. BFRESTOR will prompt you again for the diskette it wants.

Disk on Source drive is not a backup disk for filename.ext
Once you have started a restore operation, this message will be displayed if you mount a diskette which was not generated by BFBACKUP or is part of a backup series for another file. "filename.ext" will be replaced with the name of the file being restored. You have probably gotten your diskettes mixed up. BFRESTOR will prompt you again for the diskette it wants.

Disk on Source drive is a backup disk from another series.
This message indicates that the diskette you have mounted is from another series of diskettes generated by BFBACKUP while backing up the same file. This typically indicates that you have gotten your diskettes mixed up. BFRESTOR will prompt you again for the diskette it wants.

Disk on Source drive is not a valid backup disk.
This message indicates that the diskette you have mounted was not generated by the BFBACKUP program.

Disk on Source drive is backup disk # N
This message is given any time that a wrong disk in the backup series is found on the source drive. The N is replaced by the sequence number of the diskette which was mounted in the backup series. This can be caused by mounting a diskette out of order or one that has already been restored.

NOT ENOUGH SPACE
This message indicates that there is not enough space on the destination drive to restore the entire file. You should restore to another drive or delete files from the specified destination drive until there is enough space available.

Can't find file x:filename.ext
Indicates that the file "x:filename.ext" which you have requested to be erased could not be found on the destination drive. Check your typing to make sure you entered the file name correctly.

>>>> Error: Bad Source drive name.
Indicates that the logical drive letter entered on the command line for the source drive was not in the range A - P.

>>>> Error: Bad Destination drive name
Indicates that the logical drive letter entered on the command line for the destination drive was not in the range A - P.

>>>> Error: Bad file name
Indicates that the file name entered on the command line contained illegal characters.

>>>> Error: Drive letter in file name differs from Destination drive name
Indicates that the drive designation given with the destination file name on the

command line does not match the logical drive letter specified for the destination drive.

>>>> Error: Bad Verify Option specified
Indicates that characters other than "V" were found following the file name on the command line. Note that this message can be caused by specifying an extension longer than three characters for the file name.

>>>> Error: Source drive not on system
Indicates that the source drive specified on the command line is not defined for this system.

>>>> Error: Destination drive not on system
Indicates that the destination drive specified on the command line is not defined for this system.

>>>> Error: Source and Destination cannot be the same drive
Indicates that the command line specifies the same logical drive for both the source and destination drives. This is not permitted when running BFRESTOR.

>>>> Error: Source drive is a hard disk.
Indicates that the source drive specified on the command line is a hard disk drive. BFRESTOR restores from diskettes only hence this is not allowed.

6.5 Utility name: BKMOUNT

Purpose: **To prompt the user to mount a new diskette when backing up files with FILEBACK.**

General Description

The BKMOUNT utility program prompts the user to mount a new diskette on a given drive. It then waits until the user presses the <enter> key. After the user presses <enter>, BKMOUNT performs a disk system reset (CP/M BDOS function 13) and returns to the operating system via a warm boot. BKMOUNT is designed to be executed from a SUBMIT file when it is necessary to have the user mount a new diskette.

While this program is intended for use with the FILEBACK utility, it may also be of some use in making your own SUBMIT files.

Using BKMOUNT

The BKMOUNT program is executed with a command line of the form shown in Figure 6.37. The "z" in the command line should be replaced by the drive on which the new diskette is to be mounted.

```
A>BKMOUNT z<enter>
```

Figure 6.37 Command Line to Execute BKMOUNT

When BKMOUNT is executed, it will display the prompt shown in Figure 6.38 on the console. The "z" in the prompt will be replaced by the drive letter specified on the command line that executed BKMOUNT. After displaying the prompt line, BKMOUNT will wait for the <enter> key to be pressed. When <enter> is pressed, BKMOUNT will reset the disk system with BDOS function 13 and return to the system via a warm boot. This will cause any SUBMIT file which is in progress to resume.

```
Mount an empty diskette on drive z and press <enter> when ready to continue
```

Figure 6.38 Prompt Given by BKMOUNT

Possible Error Messages

None

6.6 Utility name: CLEARDIR

Purpose: To wipe clean the directory portion of a logical drive on a hard disk.

General Description

The CLEARDIR utility program is designed to provide an easy method of clearing the directory area of a logical drive which is assigned to a hard disk drive. It completely obliterates whatever is in the directory area of the disk and pays no attention to user number or "read only" file status.

When you first begin using your Corvus hard disk system, you may find that any attempt to use a logical drive on a hard disk results in an error message indicating that the directory is full. This is caused by the data pattern that the Corvus format operation leaves on the disk. CLEARDIR will fill the directory with data that the system recognizes as empty directory entries, thus eliminating the problem.

Please note that the operation of CLEARDIR is irreversible. Once you have run the program on a disk, there is no way to recover the files on that disk.

Using CLEARDIR

Figure 6.39 shows a typical console dialog for using CLEARDIR. The command shown on Line 6.39-1 will execute CLEARDIR.

```
 1:  A>CLEARDIR<enter>
 2:                  Directory Erase Utility - ver 2.xx
 3:       (c) 1981, 1983 Pickles & Trout - all rights reserved
 4:
 5:         This program allows the quick erasure of the directory on
 6:         logical drives which are assigned to hard disk drives.  It
 7:         erases all files from all user numbers regardless if they
 8:         are read only entries or not.
 9:
10:             >>>> USE THIS PROGRAM WITH EXTREME CARE!! <<<<
11:
12:  Enter logical drive on which to clear directory (x to exit): E<enter>
13:
14:  You have requested that the directory on logical drive E
15:  be cleared.  Note that this operation will result in the
16:  LOSS OF ALL FILES ON LOGICAL DRIVE E.
17:  Do you want to continue (Y/N)? Y<enter>
18:
19:  Enter logical drive on which to clear directory (x to exit): X<enter>
20:  A>
```

Figure 6.39 Example of Using CLEARDIR

CLEARDIR first displays the messages shown on Lines 6.39-2 to 6.39-11. It then asks you for the logical drive which is to have its directory cleared. At this point you should enter the letter corresponding to the drive. In this example, drive E was specified. Since the effect of CLEARDIR is completely irreversible, it warns you that all of the files on the drive will be lost (Lines 6.39-14 to 6.39-16) and asks you if you want to continue (Line 6.39-17). If you respond negatively at this point, CLEARDIR will prompt you again for a drive letter. If you respond affirmatively,

CLEARDIR will proceed to erase all of the directory entries. You may clear the directories on as many logical drives as you wish merely by continuing to enter drive letters when asked for them. When you are finished clearing directories, enter an "X" as shown on Line 6.39-19 and CLEARDIR will return you to the command level of the system.

Possible Error Messages

Please respond with the letters A - P only.
This message is displayed if you enter an improper letter when specifying the drive which is to have its directory cleared.

That logical drive is not on the system.
This message is displayed if you specify a logical drive that is not defined on the system.

That logical drive is not a hard disk.
CLEARDIR will work only with hard disk drives. This message is displayed if you attempt to use it on a diskette disk drive.

Please respond with Y or N only.
This message is displayed if you respond with characters other than "Y" or "N" to the query about continuing with the clearing operation.

Error during directory write operation.
This message is given if a disk error should occur while CLEARDIR is actually writing to the directory area of the disk.

6.7 Utility name: DRIVEMAP

Purpose: **To report the Corvus disk block numbers assigned to all logical drives that are defined to be on a hard disk.**

General Description

In some cases it is important to know exactly which Corvus disk block numbers are assigned to a logical drive on a hard disk. This information allows you to use the Corvus Mirror program to back up data from individual logical drives. In is also useful when coordinating disk usage with other systems that are not running P&T CP/M 2.

The DRIVEMAP program displays the Corvus disk block numbers assigned to all logical drives on a hard disk. This information is taken from the system that is in the computer's memory when DRIVEMAP is executed.

DRIVEMAP allows you to direct its output to the console, system printer, a disk file or any combination of these. This feature can be helpful when documenting your system configuration.

Using DRIVEMAP

The general form of the command line to execute DRIVEMAP is shown in Figure 6.40.

Figure 6.40 Form of Command Line to Execute DRIVEMAP

The output defaults to the console if no output option(s) are specified. You may direct the output to any combination of these destinations by including combinations of output options on the command line. Any destination(s) you specify will be the only destination(s) for the output. For example, if you specify a disk file for the destination, no output will appear on the console. If you want output to both a disk file and the console you must specify both destinations. An output option is specified by a slash (/) followed by the destination name. The allowed output options are listed in Figure 6.41.

/CON:	sends output to the console (default if no output options are given).
/LST:	sends output to system printer.
/(valid file name)	sends output to the specified file. If a file with the same name already exists, it is replaced.
/(valid file name)+	appends output to the specified file. If the file does not already exist, it is created. The + may be preceded by a space or comma.

Figure 6.41 Output Options For DRIVEMAP

The output from DRIVEMAP has the form shown in Figure 6.42.

```
P&T CP/M 2 (Corvus) DRIVEMAP Utility - Ver 1.xxx
Copyright 1983 Pickles & Trout,    All Rights Reserved

      Logical drives are assigned to hard disk
              drives as follows:

+-----------+-----------+-----------+-----------+-----------+
| Logical   | Corvus    | First     | Last      | Block     |
| Drive     | Drive     | Block     | Block     | Count     |
|===========|===========|===========|===========|===========|
|     A     |     1     |        1  |     4096  |     4096  |
|-----------|-----------|-----------|-----------|-----------|
|     B     |     1     |     4097  |    19968  |    15872  |
|-----------|-----------|-----------|-----------|-----------|
|     C     |     1     |    19969  |    35856  |    15888  |
+-----------+-----------+-----------+-----------+-----------+
```

Figure 6.42 Output from DRIVEMAP

Possible Error Messages

>>>> Error: No Corvus drives on the system.
Displayed if you run DRIVEMAP on a system which has no logical disk drives assigned to the Corvus hard disk.

>>>> Error: Non-modular system, not compatible with this program!
Indicates that you have tried to run DRIVEMAP on an old version of P&T CP/M 2. It will only work with the modular Corvus version of P&T CP/M 2.

>>>> Error: Not a Corvus system.
Indicates that you have tried to run DRIVEMAP on a non-Corvus version of P&T CP/M 2. It will only work with the modular Corvus version of P&T CP/M 2.

6.8 Utility name: FILEBACK

Purpose: To generate SUBMIT files that will back up all files smaller than the capacity of a diskette from a hard disk to multiple empty diskettes. The SUBMIT program is then used to start the backup.

General Description

A problem in backing up files from a logical drive assigned to a hard disk is that there is usually more information stored on the hard disk than will fit on a single diskette. Using PIP with wildcards or FASTCOPY will result in the backup stopping with an error message after the destination diskette fills up.

FILEBACK generates SUB files to use PIP to backup all of the files stored under the current user number from one logical drive to multiple diskettes. It orders the files to be transferred in such a way as to fill each diskette as completely as possible before moving on to another diskette.

Since there is a limit to the size of a SUB file, FILEBACK will generate a series of SUB files, if necessary, and cause them to chain from one to another. The very last action of the last SUB file is to erase all of the SUB files that were used in the backup operation. FILEBACK will always create the SUB files on logical drive A and will give them names of the form shown in Figure 6.43.

```
BKUPx.SUB
x is replaced by letters A, B, C, ...
```

Figure 6.43 Form of SUB File Names Created by FILEBACK

FILEBACK includes a special option if you are using the P&T Advanced Command Processor (P&T ACP) on your system. If this option is used, the contents of the SUB file created by FILEBACK are modified to be consistent with the slightly different form of command lines recognized by the P&T ACP. Figure 6.44a shows the form of a SUB file normally created by FILEBACK. Figure 6.44b shows the form of a SUB file created by FILEBACK when the "/A" option is used.

```
XSUB
BKMOUNT D
PIP
D:=C:FILE.1[ovr]
D:=C:FILE.2[ovr]
D:=C:FILE.26[ovr]
D:=C:FILE.99[ovr]
.
BKMOUNT D
PIP
D:=C:SETUP.COM[ovr]
D:=C:FORMAT.COM[ovr]
        .
        .
        .
D:=C:SDTEST.COM[ovr]
D:=C:DDCHECK.COM[ovr]
D:=C:TRS2CPM.COM[ovr]
.
ERA A:BKUP?.SUB
```

Figure 6.44a Standard CCP

```
XSUB
BKMOUNT D
PIP
<D:=C:FILE.1[ovr]
<D:=C:FILE.2[ovr]
<D:=C:FILE.26[ovr]
<D:=C:FILE.99[ovr]
<.
BKMOUNT D
PIP
<D:=C:SETUP.COM[ovr]
<D:=C:FORMAT.COM[ovr]
        .
        .
        .
<D:=C:SDTEST.COM[ovr]
<D:=C:DDCHECK.COM[ovr]
<D:=C:TRS2CPM.COM[ovr]
<.
ERA A:BKUP?.SUB N
```

Figure 6.44b P&T ACP

Figure 6.44 Examples of the Contents of SUB Files Created by FILEBACK

FILEBACK can only backup files whose size is less than or equal to the capacity of a diskette. You can instruct FILEBACK to consider the destination diskettes to be either single or double sided (the diskettes are always considered to be double density). For single sided destination diskettes the maximum size file that can be backed up is 596 Kbytes while for double sided destination diskettes the maximum size file is 1210 Kbytes. Any file larger than these limits is reported on the console with a message indicating that BFBACKUP should be used to back it up.

The SUB files generated by FILEBACK make use of several system utility programs. Figure 6.45a lists the programs that must be resident on logical drive A in order to run the backup process with the standard CCP. Figure 6.45b shows the programs required to run the backup process if you are using the P&T ACP. Only the copy of PIP that was supplied with the system should be used in the backup process. It has been patched to provide a means for SUBMIT'ed files to exit PIP and return to the operating system. If unpatched copy of PIP is used, the backup process will not work properly.

```
XSUB
SUBMIT
BKMOUNT
PIP
```

Figure 6.45a Standard CCP

```
PNTSUB
BKMOUNT
PIP
```

Figure 6.45b P&T ACP

Figure 6.45 Files Required to Run the SUB Files Created by FILEBACK

The files transferred to the diskettes during the backup process will be saved as normal disk files. They may be accessed just like any other disk files. The files may be restored to a disk drive using standard file transfer utility programs like PIP and FASTCOPY.

Using FILEBACK

All instructions are given to FILEBACK on the command line that executes it. If no instructions are included on the command line, FILEBACK displays a brief summary of the information that should appear on the command line as shown in Figure 6.46.

```
A>FILEBACK<enter>
                Fileback version 1.xx
           Copyright 1982,83 Pickles & Trout

The command line to execute FILEBACK is as follows:
  FILEBACK d:filename.typ y: /zz /a
                             |  |  |  |
                             |  |  |  /A needed if using P&T ACP
                             |  |  /SS for single sided destination diskettes or
                             |  |  /DS for double sided
                             |  |    (optional - if omitted, defaults to /SS)
                             |  |
                             |  destination drive (cannot be A:)
                             |
                             source file specification, wildcards allowed
                             d: is equivalent to d:*.*

A>
```

Figure 6.46 Help Display of FILEBACK

The command line may have either 2, 3, or 4 parameters on it. The parameters must be separated from one another by at least one space. The first parameter specifies the logical drive from which the backup is to be done (the source drive) and an optional file name. The file name may be either a unique or a wildcard file name. If the file name is present, only file names matching it will be placed in the SUB file created by FILEBACK. If no file name is given with the source drive designation (e.g. "C:") all files will be transferred. This is the same as the wildcard specification "*.*".

The second parameter is the logical drive on which the diskettes will be mounted during the backup operation. The third parameter is optional and, if present, indicates the type of diskettes onto which the backup operation is to be performed. The fourth parameter is also optional and, if present, indicates that SUB files for the P&T ACP are to be generated.

Several examples of command lines for using FILEBACK are shown in Figure 6.47.

FILEBACK A: D:	Backup all files from drive A to drive D. The destination diskettes are single sided.
FILEBACK A: D: /DS	Backup all files from drive A to drive D. The destination diskettes are double sided.
FILEBACK A:*.COM E: /A	Backup all files with the extension "COM" from drive A to drive E. The destination diskettes are single sided and the P&T ACP is in use.
FILEBACK B:*.782 D: /DS /A	Backup all files matching "*.782" from drive B to drive D. Destination diskettes are double sided and the P&T ACP is in use.

Figure 6.47 Sample Command Lines for Using FILEBACK

The console display you might see for a typical usage of FILEBACK is shown in Figure 6.48.

```
 1:  A>FILEBACK B: D:<enter>
 2:                 Fileback version 1.xx
 3:          Copyright 1982,83 Pickles & Trout
 4:
 5:  Creating SUB files to backup *.* on drive B to drive D.
 6:  Destination diskettes are single sided.
 7:
 8:
 9:  You will require 4 blank double density, single sided diskettes to
10:    backup the files from drive B
11:
12:  Log onto drive A and type
13:     SUBMIT BKUPA
14:  to begin the backup process
15:
16:  A>
```

Figure 6.48 Example of Using FILEBACK with Single Sided Diskettes

Since no instruction about the type of destination diskette was given on Line 6.48-1, FILEBACK assumes they will be single sided. On Lines 6.48-2 and 6.48-3 FILEBACK gives an opening message indicating its version number. On Lines 6.48-5 and 6.48-6 FILEBACK summarizes what it is doing so that you can verify that it is doing what you want. After it is finished, FILEBACK tells you how many diskettes you will need to perform the entire backup (Lines 6.48-9 and 6.48-10). It then reminds you of what you must do to start the backup (Lines 6.48-12 through 6.48-14).

If you have the P&T ACP, you may begin the backup process at once by merely executing Line 6.48-13 directly from the screen. To do this press the comma key followed by <enter>. This will put you in the screen editing mode. Then use the <up arrow> key to move the cursor up to the line on the display corresponding to Line 6.48-13 and press <enter> again. P&T ACP will automatically execute the line to begin the backup.

After you are returned to the command level of the operating system, you may inspect the generated SUB files if you wish. The SUB files are generated assuming that the destination diskettes will be completely empty (i.e. newly formatted). If you alter the SUB files or if the diskettes you use are not completely empty, they will probably fill up during file transfer and cause PIP to report a "DISK WRITE ERROR".

Possible Error Messages

Invalid source drive
This message indicates that no source drive was specified or the drive letter specified for the source drive is not in the range of A - P.

Invalid source file specification
This message indicates that there is something wrong with the file name you specified for the source file. Typically this is due to more than one character appearing before the colon that separates the logical drive letter from the file name.

Invalid destination drive
This message indicates that no destination drive was specified or that the drive letter specified for the destination drive is not in the range of A - P.

Destination drive cannot be drive A
Since the SUB files and the intermediate file that are used by the SUBMIT process are stored on drive A, it is not possible for drive A to be the destination drive. You should use another logical drive as the destination.

Source and destination drive cannot be the same
This message indicates that you have specified the same drive letter for both the source and destination drives. This is not allowed since it would result in transferring files from a drive to itself.

ffffffff.xxx is too large for one diskette - use big file backup program.
This message is displayed when FILEBACK finds a disk file that is too large to fit on an empty diskette. "ffffffff.xxx" will be replaced with the name of the file that is too large. The BFBACKUP utility program can be used to backup files that are larger than the capacity of a single diskette. The file named in this message will not be included in the SUB file created by FILEBACK.

Error while creating BKUPx.SUB
This message is displayed if a disk system error occurs while creating the SUB file of the name shown. The "x" in the SUB file will be replaced by letter that changes depending on how many SUB files have already been created. This error is typically caused by insufficient space on logical drive A or its directory for another file.

Out of memory, too many files
FILEBACK begins by reading the names of all the files that match the specification given on the command line into memory. It then begins to create the SUB files to perform the backup operation. If you have a very large number (usually more than 1000) of files on the source drive, FILEBACK may run out of memory before it has read all of the file names. If you should get this error you should backup only a portion of the files on the disk at one time by specifying wild card file names on the command line of FILEBACK.

No files found
This message indicates that no files were found on the source drive that match the file specification given on the FILEBACK command line.

6.9 Utility name: HDCONFIG

Purpose: **To configure the disk storage allocation of the hard disk system.**

General Description

The HDCONFIG program allows you to modify the configuration of the disk storage on your hard disk system. It allows you to modify such parameters as the capacity, number of directory entries, allocation block size, etc. for each logical drive assigned to a hard disk. The program is fully discussed in Chapter 7, Configuring the System, hence it is not described here.

NOTES

7.1 Introduction

The Corvus Hard Disk version of P&T CP/M 2 comes configured as a floppy only system for a single diskette disk drive. In order to make use of the hard disk, the Corvus hard disk driver module and a disk table module must be included in the system when it is loaded. See Chapter 3 of this addendum for more instructions about doing this.

The hard disk driver module contains the software I/O routines for actually communicating with the hard disk while the disk table module defines how the disk drives on the system are to be used. For example, a single physical hard disk can be divided into several logical drives; the disk table module contains the information about how this is to be done.

Several typical disk table modules are included in the module library (BIOSMODS.PNT) which comes with the system. Descriptions of these modules are included in Appendix A. You should review these modules to see if one of the configurations meets your needs. If one does, you need merely select it for inclusion when the system is loaded (use the MODSEL utility program or the SM option of MENU to do this). If none of them are quite right, one may be close to what you need. It is usually easier to alter an existing module than it is to construct one completely from scratch.

The HDCONFIG utility program allows you to alter and create disk table modules. It will ask a number of questions about the drive configuration and will then create a module in the library which describes that configuration.

A word of caution is in order. It is a good idea to consider the disk drive configuration carefully in advance of doing a lot of work with the system. The way in which the hard disk drives are configured is of particular importance. It is desirable to settle on a configuration for the hard disk(s) on the system fairly quickly and then stick to it. Problems can arise when changing the hard disk configuration because when the configuration is changed, the new system may not be able to access data on the hard disk written by the old system.

If it is necessary to change the configuration of the hard disks, you must copy all files you want to save off of the hard disk onto diskettes before changing the configuration (see the descriptions of the FILEBACK, BFBACKUP, and BFRESTOR utility programs in Chapter 6). After you have reconfigured the system, you can copy the files back onto the hard disk. NOTE: you CAN NOT use the Corvus MIRROR for backing up the hard disk if you are changing the hard disk configuration.

Note that if all you do is change the logical drive letter assigned to a drive definition and do not change any of the parameters of the definition, you will not need to go through the trouble of copying files to diskettes. (However, it is always a good idea to make backups before changing anything to do with the system configuration just in case something goes wrong).

As an example, say that you started with a configuration that has logical drives A and B on floppy drives and logical drives C and D on a hard disk. Later you might decide that you want logical drives A and B on the hard disk and C and D on floppy drives. To do this you would use HDCONFIG to read the old module from

the module library and then swap (using the "S" command) the definitions of drives A and B with those for drives C and D. You would then have HDCONFIG create a new module with the altered assignments and would use MODSEL to include the new module in the system when it is loaded. Since all you did was change the logical drive assignments (rather than the definitions of the drives) you will still be able to access files that were stored on the hard disk with the original configuration.

7.2 Terms and Concepts

In order to specify the attributes of each logical drive defined on a hard disk, it is necessary to have an understanding of several terms and concepts related to hard disk drives and to the way in which CP/M operates on disk drives. This section discusses the information necessary to make decisions regarding the disk configuration.

CP/M deals with disk storage on the basis of logical disk drives which are given single letter names ranging from A to P. These logical drives are associated with a physical drive by the I/O routines in the operating system. There can be up to 4 physical drives connected to the system. The first hard disk drive (the primary unit) is physical hard disk 1. The first secondary unit is physical hard disk 2 and so forth. In the hard disk versions of P&T CP/M 2, the logical to physical drive assignments are under your control. Each physical diskette drive on the system can be associated with only one logical drive (except for systems with only 1 diskette drive) but a hard disk drive can be (and usually is) associated with more than one logical drive.

When you define a logical drive on a hard disk, you must also provide information which describes various attributes of the logical drive. Some of these attributes are: the location of the logical drive on the physical drive (when a physical drive is divided among two or more logical drives, there must be no overlap of the space assigned to the various logical drives), the number of directory entries the logical drive is to have, and others.

P&T CP/M 2 accesses Corvus hard disk drives in terms of tracks each storing 8192 bytes (8 Kbytes). Different versions of the Corvus hardware have a different number of tracks available on a drive. Figure 7.1 shows the number of tracks on each of the types of Corvus hardware currently available.

drive size	revision (series)	number of tracks
10 Mb	A	1183
6 Mb	B	701
10 Mb	B	1326
20 Mb	B	2403
6 Mb	H	721
10 Mb	H	1481
20 Mb	H	2241

Figure 7.1 Number of Tracks on Different Corvus Drives

CP/M reads and writes information to disks in 128 byte units called logical sectors (sometimes called CP/M sectors). Since it is impractical to keep track of disk storage on the basis of logical sectors, CP/M groups a number of logical sectors together to form an allocation block. The number of logical sectors in an allocation block (and hence its size) may vary from logical drive to logical drive. For a single density diskette, an allocation block contains 8 logical sectors making it 1 Kbyte in size. For a double density diskette (single or double sided), an allocation block contains 16 logical sectors for a size of 2 Kbytes. For a logical

drive which is defined on a hard disk, the size of the allocation block is under your control.

Space on a logical disk is assigned to disk files on the basis of allocation blocks. When a file is first created and has no information stored in it, it has no allocation blocks assigned to it. When the first logical sector of information is written to a file, the file is assigned an allocation block. Once an allocation block is assigned to a disk file, it cannot be used by any other disk file. Because disk storage is assigned in this way, small files can take up considerably more space than the amount of information they contain.

As more information is written to a file than can be contained in one allocation block, the file is assigned another allocation block. This process is repeated as the file grows in size.

In order to access a file, CP/M must keep a record of which allocation blocks are assigned to which file. This information is kept in a directory on each logical drive. When a file is created, an entry is made in the directory for the file. As a file becomes larger, it is possible that all of the information pertinent to that file cannot fit into a single entry in the directory. In this case, another entry in the directory is made to hold the additional information. Large files can end up using several directory entries.

It is important to keep in mind that the number of directory entries is not necessarily the number of files that can be stored on the logical disk since large files can require multiple directory entries. Typically one runs out of directory space when a logical disk drive has a lot of small files stored on it rather than several large files. For a single density diskette, there is space set aside for 64 directory entries. On a double density, single sided diskette, space is provided for 128 directory entries. A double density, double sided diskette has space for 192 directory entries. For logical drives defined on a hard disk, the number of directory entries is under your control.

In addition to the directory information, CP/M must also maintain information about which allocation blocks are in use on a logical drive and which are free. This information is kept in a table in the computer's memory called an allocation vector. The allocation vector contains one bit for each allocation block on the logical drive. The more allocation blocks there are on a disk drive, the larger its allocation vector will be. When a logical disk is first accessed after a warm boot (return to command mode) or a disk system reset, the entire directory on the disk is scanned to construct the allocation vector.

When a physical disk has removable media (such as a diskette drive or the cartridge in some hard disk drives), it is possible for you to change the media without taking the necessary steps (a warm boot or disk system reset) to inform the operating system of the change. If this were to happen, it could spell disaster because CP/M will treat the new disk as if it were the previously mounted disk and possibly write to areas of the disk that are already in use. CP/M provides a mechanism for checking for changed disks to avoid this sort of problem. When the mechanism is in effect, it keeps certain information about the contents of the directory in a table in the computer's memory called a directory check vector.

Whenever the directory for a logical drive is accessed, the information read from it is compared to what is kept in the directory check vector. If a discrepancy is found, CP/M assumes that a disk change has taken place and immediately sets the logical drive to "read only" status. When a logical drive has been set to "read only", any attempt to write to that drive results in an error message. The only

way to clear the "read only" status is by a warm boot or a disk system reset. The directory checking mechanism is always used for diskette drives, however it is optional for logical drives defined on hard disks since it is not needed on a fixed disk that cannot be changed.

Note that you might elect to include directory checking on logical drives assigned to a Corvus hard disk if you are using the Corvus Constellation. In this case, the directory checking will detect if another user on the system has written to a logical drive without you knowing about it. This is discussed in more detail in Chapter 9.

7.3 Tradeoffs

When configuring a hard disk system, there are a multitude of tradeoffs that must be considered in deciding on the configuration to use. The two variables that enter most prominently into the considerations are memory usage by the operating system and speed of access to disk drives. Unfortunately the items that must be considered are interdependent which makes it somewhat difficult to consider them one at a time.

The first thing to do when configuring a hard disk system is to try to define the use to which it will be put. For example if the hard disk is to be used primarily for large files, different choices will be indicated than if it were to be used for a large number of small files. Another consideration is the software that will be running on the drive. If the software requires a certain number of logical drives it may dictate how the hard disk is divided into logical drives. Occasionally a program will not be able to make use of a logical drive larger than a certain size; in this case, the size of the logical drives defined on a hard disk may be tailored to what the program requires.

In general, the following information will be useful in selecting the parameters for the logical drives that are defined on a hard disk:

1. The permissible sizes for the allocation blocks are 2, 4, 8, and 16 Kbytes.

2. The largest number of files that it is likely will be stored on a given logical drive defined on a hard disk. Keep in mind that a different number may used for each of the logical drives defined on a hard disk.

3. The number of small files (less than 16 Kbytes) that are likely to be stored on a logical drive.

4. The number of large files (greater than 200 Kbytes) that are likely to be stored on a logical drive.

5. The approximate average size of the disk files to be stored on a logical drive.

6. Any restrictions on logical drive size that may be imposed by programs that will be run on the system.

7. Any restrictions on the number of logical drives that may be imposed by programs that will be run on the system.

8. The maximum logical drive size allowed by CP/M 2 is 8 Mbytes (8192 Kbytes).

A good way to start the selection of parameters is to consider the allocation block size. Recall that the size of the allocation vector in memory is related to the allocation block size. Thus, the choice made for the allocation block size directly affects the memory required by the operating system. Larger block sizes are generally more useful when mostly large files are to be stored on a logical drive. If a large allocation block size is specified for a drive that has mostly small files, a lot of disk space is wasted since even very small files will be assigned one entire allocation block. A large allocation block reduces the memory required for the allocation vector since, for a given size of logical drive, there will be fewer allocation blocks than if a smaller allocation block is specified.

The choice of allocation block size is also affected by the size of the logical drive. If the logical drive is large, the space wasted by large allocation blocks may be of less importance than for smaller logical drives. If a small logical drive (about 1 Mbyte) is specified, it is usually used for storage of medium sized files such as programs and a smaller allocation block size is appropriate. The permissible sizes for the allocation block are 2, 4, 8, and 16 Kbytes.

Typically, for large logical drives (4 Mbytes or more) an allocation block size of 8 or 16 Kbytes is most appropriate. 8 Kbytes is a good "middle of the road" figure but, if the disk usage is heavily slanted towards large files, 16 Kbytes might be better. For smaller logical drives (1 to 4 Mbytes), the choice will most likely be between 8 and 4 Kbytes. In this case 8 Kbytes is still a good choice for most situations but if a lot of smaller files are going to be stored on the logical drive, 4 Kbytes might be a better choice.

2 Kbyte allocation blocks are typically only used on small logical drives (probably less than 1 Mbyte) since the memory overhead for the allocation vector is quite high for 2 Kbyte allocation blocks. For example, a logical drive of 5 Mbytes would require over 300 bytes for the allocation vector with 2 Kbyte allocation blocks but just about 80 bytes for 8 Kbyte allocation blocks. Your choice of allocation block size can have a significant affect on the amount of memory available to programs.

The other parameter that has major tradeoffs associated with it is the number of directory entries. For logical drives that are set up to check for disk changes, space for the directory check vector must be allocated in memory. The more directory entries there are, the more space will be required for the directory check vector. Of course, no check vector is allocated for a drive that does no checking for changed disks.

The number of directory entries on a logical disk drive also can have an affect on speed of access to the drive. As was previously noted, CP/M scans the entire directory when a logical drive is first accessed after a warm boot or disk system reset. The larger the directory, the longer it will take to make this scan. After the first access to a drive, the time required for directory operations is not highly dependent on the size of the directory. Of course, directory operations will slow down somewhat as the directory fills up but the additional time is usually small.

Although the time it takes to scan the entire directory may not seem very important (since it only occurs during the first access to a drive after a warm boot or disk system reset) it bears some consideration. If the usage of the system is such that a warm boot or disk system reset occurs only at the end of programs then the time taken for the directory scan is not too important. However, many programs do make use of the disk system reset feature in order to account for changed disks. If a program frequently resets the disk system, a large directory may degrade the program's performance.

Another instance when the time to scan the directory becomes a noticeable factor is when using SUBMIT files. Typically a SUBMIT file is used to execute several programs in succession. Since programs used with SUBMIT must terminate with a warm boot, the entire directory of one or more logical drives is scanned repeatedly in the course of running a SUBMIT file.

It is important to allot enough directory entries to a drive that future usage will not force the system to be reconfigured but specifying too large a directory can have performance penalties. Usually 256 directory entries are sufficient for a hard disk drive. In some cases where a large number of small files are being kept on the disk, 512 entries may be appropriate. It is only in rare circumstances that more than 512 directory entries will be needed on any one logical drive.

Another factor that can affect system performance is the definition of logical drives. When P&T CP/M 2 performs a warm boot (typically at the end of a program), it logs on logical drive A, and then logs on the current default drive if it is different from A. Usually less time is required to log on a hard disk drive than a diskette drive even though there are considerably more directory entries on a logical drive on the hard disk. System performance can be improved in some cases by defining logical drive A on a hard disk. It is important to keep in mind that if logical drive A is defined on a hard disk, the hard disk must be on-line and ready in order for the system to operate.

7.4 Single Diskette Drive Systems

If you have a system with a single diskette drive, you must decide how to handle it. You can configure the system to treat the single diskette drive as 1 to 4 logical drives. If you choose to have the system treat the drive as multiple logical drives it will issue messages on the system console for swapping diskettes in the drive (see Section 3.3 of the P&T CP/M 2 User's Manual). This technique allows you to run software that needs to access up to 4 diskette drives. With the hard disk system, this capability becomes somewhat less interesting because most of your work will probably be done on the hard disk.

Each logical drive assigned to the floppy drive will require about 150 bytes of memory. In order to minimize the memory requirements of the system, you should assign no more logical drives to the floppy drive than you actually need. About the only time you should need multiple logical drives assigned to the floppy is when a program must be able to access two or more diskettes. Remember that since you can reconfigure the system, you could have one system diskette configured for automatic swapping and another without it.

Since the Corvus hard disk version of P&T CP/M 2 uses a system diskette for warm booting, you will usually want a system diskette mounted on physical floppy drive 0. If you have only a single diskette drive, and since system diskettes are always double density, you may run into some trouble accessing a single density diskette. There are two solutions to this problem.

The first is to assign logical drive A to the hard disk. In this case you will be able to switch to a single density diskette after a warm boot has taken place. Just be sure to mount the system diskette before the next warm boot (this occurs when most programs return to the system command level); failure to do so will cause the system to issue an error message (see Section 9.4 of P&T CP/M 2 User's Manual).

If you must assign logical drive A to the physical floppy drive, the second solution is to assign a second logical drive to the floppy drive. Assigning multiple logical drives to the single floppy causes the system to prompt for disk swapping as previously described. If you always use this second logical drive for transfers to and from diskettes, the disk swapping mechanism of the system will insure that you can access any density floppy diskette.

If you indicate to HDCONFIG that you have only one physical diskette drive, it will ask you how many logical drives you want assigned to it. By responding with "1", the disk configuration you create will NOT allow drive swapping. Responding with "2" to "4" will allow swapping.

7.5 Getting Ready

Before using HDCONFIG, it is useful to spend a few minutes thinking over how the system is to be configured. Some points to keep in mind while deciding on the configuration are:

1. A logical drive defined on a hard disk must be entirely contained by that hard disk. Logical drives cannot span from one hard disk to another.

2. The location of logical drives on a hard disk is specified in terms of tracks. A single track holds 8 Kbytes.

The next page shows a hard disk configuration worksheet that can help in setting up the hard disk configuration. Extra copies of this worksheet are provided with the system documentation and you may make more if necessary. These worksheets are provided as a tool to help you organize your thinking about how to configure the disk system. An experienced user may not need to use them but it is a good idea to use them the first time or two you configure a system.

The following pages show hard disk configuration worksheets for typical disk configurations. Disk table modules for these configurations and others are included in the module library. Refer to Appendix A for a description of the disk table modules. If one of them suits your needs, you need merely select it (using MODSEL) for inclusion in the system. If one of them is close to what you want, you can edit it with HDCONFIG.

Corvus Hard Disk System Configuration Worksheet

logical drive	hard or floppy	physical drive #	maximum track #	begin at track	end at track	# of tracks	# of Kbytes	alloc block size	# of dir ents	dir check Y/N	initial access Y/N	initial access r/w - r/o

max track # for	6 Mb	10 Mb	20 Mb
rev A		1182	
rev B	700	1325	2402
rev H	720	1480	2240

Notes: Each track holds 8 Kb.

Maximum number of tracks allowed on a logical drive is 1024.

Corvus Hard Disk System Configuration Worksheet

logical drive	hard or floppy	physical drive #	maximum track #	begin at track	end at track	# of tracks	# of Kbytes	alloc block size	# of dir ents	dir check Y/N	initial access Y/N	initial access r/w - r/o
A	hard	1	2402	0	354	355	2840	4K	256	N	Y	r/w
B	hard	1	2402	355	1378	1024	8192	8K	512	N	Y	r/w
C	hard	1	2402	1379	2402	1024	8192	8K	512	N	Y	r/w
D	floppy	0										
E	floppy	1										

max track # for	6 Mb	10 Mb	20 Mb
rev A		1182	
rev B	700	1325	2402
rev H	720	1480	2240

Notes: Each track holds 8 Kb.
Maximum number of tracks allowed on a logical drive is 1024.

Configuration worksheet for a 20 Mbyte Rev B drive where the hard disk is divided into one small (2 Mb) and two large (8 Mb) drives with two floppy drives. Drives A, B, & C are defined on the hard disk and drives D & E are defined as floppies. (Module name: HDTB20-1)

Corvus Hard Disk System Configuration Worksheet

logical drive	hard or floppy	physical drive #	maximum track #	begin at track	end at track	# of tracks	# of Kbytes	alloc block size	# of dir ents	dir check Y/N	Initial access Y/N	Initial access r/w - r/o
A	hard	1	1480	0	456	457	3656	4K	256	N	Y	r/w
B	hard	1	1480	457	1024	1024	8192	8K	512	N	Y	r/w
C	floppy	0										
D	floppy	1										

max track # for	6 Mb	10 Mb	20 Mb
rev A		1182	
rev B	700	1325	2402
rev H	720	1480	2240

Notes: Each track holds 8 Kb.
Maximum number of tracks allowed on a logical drive is 1024.

Configuration worksheet for a 10 Mbyte Rev H drive where the hard disk is divided into one small (3.5 Mb) and one large (8 Mb) drives with two floppy drives. Drives A & B are defined on the hard disk and drives C & D are defined as floppies. (Module name: HDTH10)

Corvus Hard Disk System Configuration Worksheet

logical drive	hard or floppy	physical drive #	maximum track #	begin at track	end at track	# of tracks	# of Kbytes	alloc block size	# of dir ents	dir check Y/N	initial access Y/N	initial access r/w - r/o
A	hard	1	2240	0	559	560	4480	8K	256	N	Y	r/w
B	hard	1	2240	560	1119	560	4480	8K	256	N	Y	r/w
C	hard	1	2240	1120	1679	560	4480	8K	256	N	Y	r/w
D	hard	1	2240	1680	2240	561	4488	8K	256	N	Y	r/w
E	floppy	0										
F	floppy	1										

max track # for	6 Mb	10 Mb	20 Mb
rev A		1182	
rev B	700	1325	2402
rev H	720	1480	2240

Notes: Each track holds 8 Kb.
Maximum number of tracks allowed on a logical drive is 1024.

Configuration worksheet for a 20 Mbyte Rev H drive where the hard disk is divided into four approximately equal (4.4 Mb) with two floppy drives. Drives A, B, C, & D are defined on the hard disk and E & F are defined as floppies. (Module name: HDTH20-2)

7.6 Using HDCONFIG

HDCONFIG is a very user interactive program that allows you to specify the disk drive configuration for the hard disk system. To do this, it takes the information you enter and creates a disk table module. As much as is possible, it shows you the consequences of your choices as you make them and allows you to make changes as necessary. HDCONFIG will run on a floppy only system so it is not necessary to use MODSEL (or the SM option of MENU) to select modules to access the hard disk before running HDCONFIG.

IMPORTANT NOTE: HDCONFIG only adds a new disk table module to the disk table modules in the module library or modifies one of the existing disk table modules. In order for the configuration in a new disk table module to take effect, you must use MODSEL to select the new module for inclusion in the system. The configuration in the new module will take effect the first time the system is loaded (RESET) after the new module is selected for inclusion.

Although HDCONFIG is really quite easy to use, its interactive nature makes it difficult to document in a written form. If you are confused by the following descriptions, run the program while you read. You can try the various options and see the results. Running HDCONFIG will have no affect on your system and you can abort the program at any time. After practicing with HDCONFIG, you may not want to create a new module; just press <break> when HDCONFIG asks if you are ready to install the new module in the library. Even if you do create a module you don't want, no harm is done since the configuration contained in the module does not take effect until you use the MODSEL utility program to include the new module in the system when it is loaded.

The figures in this description illustrate the appearance of the console display at different points in the program. In some cases, the entire console display is shown. In other cases only the pertinent portion of the display is shown in order to save space. This is done, for example, when there is only one or two lines of text displayed in the center of the screen. Whenever HDCONFIG poses a question that requires a "yes/no" type of answer, it will accept several responses. The characters "Y", "y", "1", "T", and "t" are accepted as affirmative answers while the characters "N", "n", "0", "F", and "f" are accepted as negative answers. The responses to these questions will always be shown as "Y" or "N" in the figures.

Any time that HDCONFIG is waiting for input from the console, you may press the <break> key. HDCONFIG will take this to mean that you want to abort the program and return to the command level of CP/M. After you press <break>, HDCONFIG will query you to find out if you really want to exit the program. If you respond affirmatively to this query, HDCONFIG will return you to the command level of CP/M, otherwise HDCONFIG will resume where it was when you hit <break>.

You may also press <ctl-P> any time HDCONFIG is waiting for console input. This will result in the current screen display being printed on the system printer. This is useful for documenting the parameters you have specified for a particular drive configuration. Note that the system printer must be ready to accept characters when you press <ctl-P>. If it is not the system will hang up until the printer is ready.

The last 3 lines of the console display are used by HDCONFIG for displaying instructions and error messages. The figures in this section do not show any of these error messages since they disappear when corrective action is taken.

HDCONFIG has essentially three parts. In the first part you will be asked to supply information about the physical disk drives on your system. You are also able to read parameters from an existing disk table module in order to edit them. In the second part, you enter (or modify) the information that defines each of the logical drives you want on the system. The third part installs the new disk table module in the module library so that it will be available for selection by MODSEL.

Part 1 - Entering Physical Drive Information

The command line shown in Figure 7.2 will execute the HDCONFIG program. Note that HDCONFIG consists of a main program and 3 overlay files (HDCONFIG.001, HDCONFIG.002, and HDCONFIG.003). All four files must be present on the current drive when HDCONFIG is executed. An error message will be displayed if this is not the case and HDCONFIG will return immediately to the system command level.

```
A>HDCONFIG<enter>
```

Figure 7.2 Command Line to Execute HDCONFIG

The first thing HDCONFIG does is to clear the console display and give you a warning message as shown in Figure 7.3. This message reminds you that the standard Corvus interface board is incompatible with the 68000 board set in a Model 16. If you have a Model 16 (or a Model II or 12 with the 68000 board set installed), you must either remove the 68000 board set or contact Corvus for a different interface board. If you get a modified interface board, you must contact Pickles & Trout for a patch to the system in order for it to work with the modified board.

```
Do not use this program if there is a 68000 board in the
system. There is a conflict between the data ports of the
68000 board and the Corvus interface that will prevent the
Corvus from working correctly.

Press <break> to exit program
```

Figure 7.3 Warning About 68000 Board Set

The next thing HDCONFIG does is to clear the console display and ask for the drive on which it can find the BIOSMODS.PNT file (Figure 7.4). This is the file that contains the library of modules. You should enter the logical drive letter of the drive on which your working system diskette is mounted. In this case, logical drive A is specified (which is typically the case when you are first configuring your system).

```
Enter drive on which to find BIOSMODS.PNT: A<enter>
```

Figure 7.4 Entering Drive for Library File

After HDCONFIG locates the BIOSMODS.PNT file, it asks you if you want to read parameters from an existing module as shown in Figure 7.5. You would respond affirmatively if you want to edit the configuration in an existing module. In this case, a negative response is entered indicating that an entirely new configuration is to be created.

```
Do you want to read parameters from an existing disk table module (Y/N)? N<enter>
```

Figure 7.5

After a negative response in Figure 7.5, HDCONFIG will ask you to indicate how many physical floppy drives are on the system as shown in Figure 7.6. In this example there are two physical floppy drives.

```
How many physical floppy drives are on the system:  2<enter>
```

Figure 7.6 Specifying Number of Floppy Drives

Next HDCONFIG will ask you for the number of hard disk drives on the system (Figure 7.7). You may enter any number from 0 to 4. If you enter 0, you will not be permitted to define any logical drives on a hard disk. A very typical response is 1 as is shown in the figure.

```
How many physical hard disk drives are on the system:  1<enter>
```

Figure 7.7 Specifying Number of Hard Disk Drives

After you have entered the number of hard disk drives, HDCONFIG will ask you to specify the type of each of them. Figure 7.8 shows the console display for entering this information. Selections 1 through 7 correspond to the various different drive and controller combinations currently available from Corvus. HDCONFIG attempts to access the Corvus drive to determine which type it is. If it successfully identifies the drive type, it will highlight the appropriate selection with reverse video and make that selection number the default for the query on Line 7.8-14.

You should check whether it has correctly determined the type of drive; if it has, you need merely press <enter> to accept the value. If the type is incorrect, you may enter any of the other selections before pressing <enter>. If the Corvus system is not installed or not turned on, HDCONFIG will not be able to determine the type of drive and will set 0 (corresponding to "other" drives) as the default value.

You will be presented with the same question for each of the physical hard disk drives on the system. If HDCONFIG cannot determine the types of drives 2, 3, or 4, it will set the default drive type to the type specified for the previous drive. Since it is most likely that they are all the same (although the system does allow you to have different sized physical drives if you wish), you need only enter the appropriate type for the first drive and press <enter> for all the rest.

```
 1:                    Type of Hard Disk Drive 1
 2:
 3:                      1. Corvus 10 Megabyte Rev A
 4:                      2. Corvus  6 Megabyte Rev B
 5:                      3. Corvus 11 Megabyte Rev B
 6:                      4. Corvus 20 Megabyte Rev B
 7:                      5. Corvus  6 Megabyte Rev H
 8:                      6. Corvus 10 Megabyte Rev H
 9:                      7. Corvus 20 Megabyte Rev H
10:
11:                      0. other
12:
13:          Drive type if found on system is shown reverse.
14:              Enter number for the type for drive 1 : 7
15:
```

Figure 7.8 Specifying Type of Hard Disk Drives

If you select the "other" option for a hard disk drive, HDCONFIG will ask you for the size of that drive. Corvus specifies drive size in 512 byte blocks hence HDCONFIG requests the number of blocks on the drive. Check your Corvus

documentation for the number of blocks available on your drive. Figure 7.9 shows how this information is entered.

```
1:                    Please supply the following information
2:                          about hard disk drive 0
3:
4:                    Number of blocks (1-65535):          11220<enter>
5:
6:
7:
8:                              Is this correct? Y<enter>
9:
```

Figure 7.9 Entering Information for "Other" Types of Hard Disk Drives

HDCONFIG will now show you a summary of the information you have entered. Skip ahead to Figure 7.13.

If you had given an affirmative response to the question in Figure 7.5 (indicating that you want to read parameters from an existing module), HDCONFIG would display the table of contents of the module library as shown in Figure 7.10. The names of all the disk table modules will be shown in reverse video. At the bottom of the display, you will be asked to enter the name of the module from which you want to read the parameters.

```
1:    P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
2:
3:
4:
5:
6:
7:                              Library Table of Contents
8:
9:                          Disk Table Modules are shown reverse
10:
11:    AUTOKEY  KEYXLATE SCRNDUMP ADM3     4FLOPPY  3FLOPPY  2FLOPPY  1FLOPPY
12:    TYPENAME COREINIT SIO      PPMIN    PPNORS   PPSTDR   PPSTD    M12CRT1
13:    STDCRT1  M12WB1   STDWB1   STDCORE1 CORV1    CORV2    HDTB10   HDTB20-1
14:    HDTB20-2 HDTH6    HDTH10   HDTH20-1 HDTH20-2 HDTH20-3 MYHD
15:
16:
17:
18:
19:
20:                    Enter name of disk table module to read:
21:      To view text portion of modules press <enter> without typing a name
22:
```

Figure 7.10 Table of Contents Display When Reading a Disk Table Module

If you enter a module name, HDCONFIG will read the disk parameters from it and proceed. Each disk table module has a text description associated with it to help you remember how it is configured. When HDCONFIG asks for the module to read, you may just press <enter> to read the text description of a module. In this case HDCONFIG will ask which module's description you want to read as shown in Figure 7.11.

```
Read text from which module : HDTH20-1<enter>
```

Figure 7.11 Entering Name of Module From Which to Read Text

HDCONFIG will then show you the text description associated with that module as shown in Figure 7.12.

```
P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx


                    SAMPLMOD - created 5/16/83
                          2 floppies
                      1 20 Meg Rev H Drive

 Drive    Alloc Blk    #Dir Ents    Start Track    #Tracks    Size
   A          4k          256            0           256      2 Mb
   B          8k          256          256           992      7.7 Mb
   C          8k          256         1248           993      7.7 Mb

   D -> phys floppy 0
   E -> phys floppy 1
```

Figure 7.12 Display of Text Description from a Module

After HDCONFIG has read a module or after you have finished entering the number and types of the hard and floppy disk drives, HDCONFIG will present you with a summary as shown in Figure 7.13.

```
            These Physical Drives are Present on the System

     1 hard disk drive                    2 floppy disk drives

   Drive # Trks  Blocks
      1    2241  35860

                          Is this correct? N<enter>
```

Figure 7.13 Summary of Physical Drives

If the parameters displayed are correct, you should enter an affirmative response to proceed to the configuration portion of the program. If you enter a negative response, you will be given the option of either changing the number (and types) of drives or reading another module as shown in Figure 7.14.

This option is very useful when adding drives to your system. For example, say that when you initially configured the system you had only a single hard disk drive but you have just added another. In this case you will probably want to keep the same configuration on the old drive and merely add some new logical drives on the new one.

The easiest way to do this is as follows: First read the disk table module containing the configuration for your single hard drive system. When HDCONFIG asks you if the physical drive summary is correct (Figure 7.13) respond negatively. Select option 2 to change the number of drives in Figure 7.14. You may now tell HDCONFIG that you have two hard disk drives instead of one. HDCONFIG will "remember" all the configuration information it read from the old module (so that you won't need to re-enter it) and will allow you to define new logical drives on the new hard disk. You can, of course, use the same procedure when adding floppy drives to the system.

```
1:
2:                    1. Read another module.
3:
4:                    2. Change number of drives.
5:
6:
7:                 Enter number of your selection:  1<enter>
```

Figure 7.14 Options if Physical Drive Summary is Not Correct

If the system has a single floppy drive, the summary of the physical drives will appear as shown in Figure 7.15. If you are entering a completely new configuration or are changing the number of drives in an old configuration, HDCONFIG will ask you (as shown) how many logical drives you want assigned to the single floppy drive. If you respond with 2, 3, or 4, a system using this configuration will issue disk swapping messages to emulate multiple diskette drives using the single physical drive. If you respond with 1, this swapping will not be done.

```
1:    P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
2:
3:
4:
5:
6:
7:              These Physical Drives are Present on the System
8:
9:       1 hard disk drive                    1 floppy disk drive
10:
11:      Drive # Trks  Blocks          note: Since this is a single floppy
12:          1   2241  35860         drive system, you may assign up to
13:                                      4 logical drives to the single
14:                                          physical floppy drive.
15:
16:
17:              How many logical floppies do you want? 2<enter>
18:
19:                        Is this correct? Y<enter>
20:
```

Figure 7.15 Physical Drive Summary for a Single Floppy System

Part 2 - Defining the Logical Drives

After you have accepted the physical drive summary and assuming that you did not have HDCONFIG read parameters from an existing module, it will next create a console display as shown in Figure 7.16. The top 5 lines of this display will remain in place as you enter the configuration information and will be updated to indicate any changes you make.

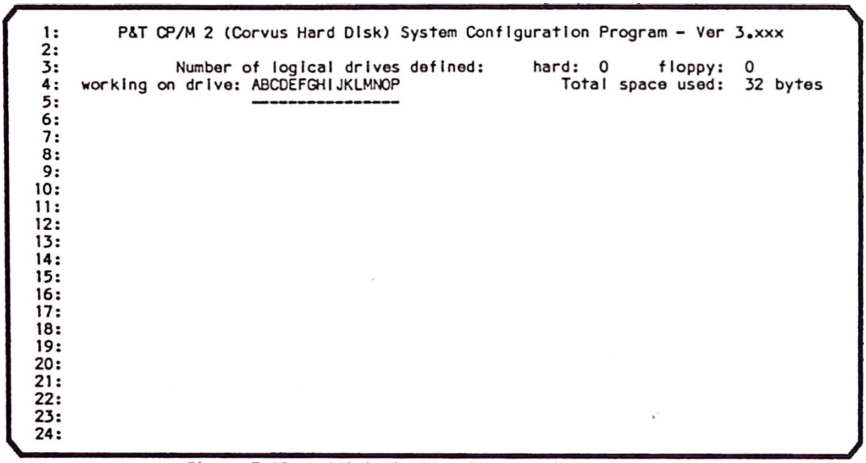

Figure 7.16 Initial Display with no Drives Defined

Line 7.16-3 shows the number of logical drives that have been specified on both hard and diskette drives. These numbers are updated as drives are defined and modified. The left half of Lines 7.16-4 and 7.16-5 form a status area that gives a visual indication of which logical drives have been defined. If the logical drive has been defined, either an "f", indicating a diskette (floppy) drive, or an "h", indicating a hard drive, will be displayed just below the logical drive letter. If the drive has not been defined, a hyphen will be displayed. The current logical drive is designated by displaying its letter and any symbol below it in reverse video. At the right end of Line 7.16-4, the total space used by the disk drive parameters is displayed.

The current logical drive can be changed by using the <left arrow>, <right arrow>, and <enter> keys. Pressing <right arrow> or <enter> will advance the current logical drive while pressing <left arrow> will back the current logical drive up by one. As the current logical drive is changed, the reverse video indicator on Lines 7.16-4 and 7.16-5 will be moved to the appropriate position. After about half a second, the logical drive's parameters will be displayed (if the drive is defined) or the bottom portion of the screen will be erased (if the drive is not defined). In addition to responding to <left arrow>, <right arrow>, and <enter> at this point, HDCONFIG will respond to several command keys.

Pressing one of "f", "F", or <F1> causes HDCONFIG to define a previously undefined drive as a diskette drive. Pressing one of "h", "H", or <F2> causes HDCONFIG to define a previously undefined drive as a hard disk drive. "E", "e", or <down arrow> indicates to HDCONFIG that you want to edit the parameters for the current logical drive. "-" or "." will tell HDCONFIG that you want to delete the definition of the current logical drive.

"m" or "M" indicates that you want to move the parameters for the current logical drive to another logical drive. If you press either of these keys HDCONFIG will ask you to which logical drive you want to move the current logical drive definition. You should enter a logical drive letter that does not have a definition yet. After entering the letter, HDCONFIG will move the current definition to that logical drive and will delete the definition from the current drive.

"s" or "S" indicates that you want to swap (exchange) the parameters for the current logical drive with those for another logical drive. If you press either of these keys HDCONFIG will ask you with which logical drive you want to swap the current logical drive definition. You should enter a logical drive letter that already has a definition. After entering the letter, HDCONFIG will exchange the current definition with that logical drive.

The "M" and "S" commands are useful when you want to change the logical drive assignments without changing any configuration parameters. "x" or "X" indicates that you have finished entering the definitions. <break> indicates that you wish to abort the program and return to the operating system. The various command keys are summarized in Figure 7.17.

while cursor is in status area

character	function
<left arrow>	move to previous logical drive
<right arrow>, <enter>	move to next logical drive
"-", "."	delete the definition of the current logical drive
"F", "f", <F1>	define the current logical drive as a floppy drive
"H", "h", <F2>	define the current logical drive as a hard disk drive
"E", "e", <down arrow>	edit the definition of the current logical drive
"m", "M"	move the current drive definition to another logical drive
"s", "S"	swap the current drive definition with one for another logical drive
"X", "x"	quit entering drive definitions and create new module

while entering or editing parameters

character	function
<enter>	Accept new value and/or move to next parameter. If cursor is at last parameter, it returns to the status area.
<up arrow>	Move cursor up to previous parameter. If cursor is at first parameter, it returns to the status area.

Figure 7.17 Summary of Command Characters Accepted by HDCONFIG

When a logical drive is first defined, HDCONFIG will display a set of default parameters for that drive. The console cursor will then move to the first of these parameters and wait for your input. If you want to keep the default value press, <enter> or <down arrow>. If you want to change the value, enter the new value and press <enter>. If you are on the first parameter and press <up arrow>, the cursor will return to the status area and the drive definition will be deleted. Note that this action is taken only the first time a drive is defined. If the drive has already been defined (hence you are editing its parameters) using <up arrow> when you are at the first parameter will not delete the drive definition but will merely return the cursor to the status area.

If there are several parameters, the <up arrow> and <down arrow> can be used to move up and down the list to make changes. To change a parameter, simply move the cursor to the parameter and type in the new value. Other numbers on the screen may change if they depend on the parameter you changed.

For a diskette drive, the total space used on Line 7.16-4 will be updated immediately after the drive is defined. For a hard disk drive, the number will be changed only after all of the drive parameters have been entered and accepted. If you edit a hard disk drive's parameters, the space it requires is subtracted from the total space figure when you begin editing. After you have finished editing the parameters, the space required by that drive (it may have changed based on your editing) will be added into the total again.

When you specify that a logical drive is to be defined as a diskette drive, HDCONFIG presents a display as shown in Figure 7.18. HDCONFIG always sets the physical drive number of a newly defined diskette drive to 0 as shown. If you wish to change the physical drive number, enter the number you wish and press <enter>. Note that the total space used number has been increased by 149 since 149 bytes are required for the parameters associated with a diskette drive.

```
 1:      P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
 2:
 3:           Number of logical drives defined:     hard: 0     floppy: 0
 4:  working on drive: ABCDEFGHIJKLMNOP                Total space used: 181 bytes
 5:                    f---------------
 6:
 7:                            Floppy Disk Drive Parameters
 8:
 9:                          Physical drive number:    0
10:
11:
12:
```

Figure 7.18 Display with One Floppy Drive Defined

The display shown in Figure 7.19 occurs after three diskette drives (A, B, and C) have been defined and a hard disk drive (E) has just been defined. The parameters shown are the default values for a newly defined hard disk drive. The "xxxx" on line 7.19-14 will be replaced by the actual number of tracks on physical hard disk drive 1. If another physical drive is selected, the number will be changed to the number of tracks on that drive. Note that many of the default values are not valid; you must change at least some of them to have a valid drive definition. If the cursor is at the first of the parameters and you press <up arrow>, the drive definition will be deleted and the cursor will return to the status area.

```
 1:      P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
 2:
 3:            Number of logical drives defined:     hard: 0     floppy: 3
 4:  working on drive: ABCDEFGHIJKLMNOP                 Total space used: 479 bytes
 5:                    fff-h----------
 6:
 7:                              Hard Disk Drive Parameters
 8:
 9:          Drive is initially accessible (Y/N):    Y
10:          Drive is initially read/write (Y/N):    Y
11:    Allocation block size in Kbytes (2,4,8,16):   0
12:                        Physical drive number:    0
13:
14:   Maximum number of tracks on drive 1 is xxxx
15:                         Beginning track number:   0
16:                              Number of tracks:    0
17:                   Disk size in Kbytes is 0
18:  Number of directory entries (multiple of   0):   1
19:                 Check for changed disks (Y/N):    N
20:                      Memory used =  39 bytes
21:
22:
23:
24:
```

Figure 7.19 Beginning to Define a Hard Disk Drive

Figure 7.20 shows the display after the user has set the initial drive access parameters and the allocation block size. Note that the 0 on Line 7.20-18 has changed to a 256 on Line 7.20-18. This results because the directory must occupy one or more complete allocation blocks on the disk. 256 directory entries will fit into an 8 Kbyte allocation block, hence the number of directory entries must be a multiple of 256 (in this example).

```
 1:      P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
 2:
 3:            Number of logical drives defined:     hard: 0     floppy: 3
 4:  working on drive: ABCDEFGHIJKLMNOP                 Total space used: 479 bytes
 5:                    fff-h----------
 6:
 7:                              Hard Disk Drive Parameters
 8:
 9:          Drive is initially accessible (Y/N):    Y
10:          Drive is initially read/write (Y/N):    N
11:    Allocation block size in Kbytes (2,4,8,16):   8
12:                        Physical drive number:    0
13:
14:   Maximum number of tracks on drive 1 is xxxx
15:                         Beginning track number:   0
16:                              Number of tracks:    0
17:                   Disk size in Kbytes is 0
18:  Number of directory entries (multiple of 256):   1
19:                 Check for changed disks (Y/N):    N
20:                      Memory used =  39 bytes
21:
22:
23:
24:
```

Figure 7.20 Display After Setting Access Flags and Allocation Block Size

Figure 7.21 shows the display after the user has specified the drive and range of tracks on that drive. Line 7.21-14 reminds you of the total number of tracks available on the drive so that you can make appropriate choices for the beginning track and number of tracks. After the range of tracks is specified, the storage capacity of the drive is calculated and displayed on Line 7.21-17. This number

includes any space that will be used by the directory, hence the usable storage on the drive will be a little less.

Note that for the same number of tracks, the amount of storage may vary a little for different allocation block sizes. This is due to the requirement that a logical drive contain an integral number of allocation blocks. A small amount of disk storage may be wasted if an integral number of allocation blocks do not fit exactly into the specified number of tracks. (Recall that a track stores 8 Kbytes.) The amount of waste will depend on the allocation block size. The amount of memory used (Line 7.21-20) has been increased to include the space taken up by the allocation vector.

```
 1:      P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
 2:
 3:            Number of logical drives defined:     hard: 0     floppy: 3
 4:  working on drive: ABCDEFGHIJKLMNOP                 Total space used: 479 bytes
 5:                    fff-h----------
 6:
 7:                              Hard Disk Drive Parameters
 8:
 9:           Drive is initially accessible (Y/N):    Y
10:           Drive is initially read/write (Y/N):    N
11:     Allocation block size in Kbytes (2,4,8,16):   8
12:                         Physical drive number:    0
13:
14:   Maximum number of tracks on drive 1 is xxxx
15:                           Beginning track number:   1
16:                                  Number of tracks: 511
17:                       Disk size in Kbytes is 4336
18:  Number of directory entries (multiple of 256):    1
19:                    Check for changed disks (Y/N):   N
20:                         Memory used = 107 bytes
21:
22:
23:
24:
```

Figure 7.21 Display After Entering Number of Tracks

Figure 7.22 shows the display after the user has specified the number of directory entries and whether or not to check for changed disks. After all of the hard disk parameters have been entered, HDCONFIG asks for confirmation that they are OK (Line 7.22-21). An affirmative answer results in the parameters being accepted and the cursor returning to the status area. A negative response results in the cursor being returned to the disk checking parameter on Line 7.22-19 so that the parameters may be modified.

```
1:      P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
2:
3:            Number of logical drives defined:     hard:  0     floppy:  3
4:  working on drive: ABCDEFGHIJKLMNOP                 Total space used: 479 bytes
5:                    fff-h----------
6:
7:                              Hard Disk Drive Parameters
8:
9:            Drive is initially accessible (Y/N):    Y
10:           Drive is initially read/write (Y/N):    N
11:     Allocation block size in Kbytes (2,4,8,16):   8
12:                        Physical drive number:     0
13:
14:   Maximum number of tracks on drive 1 is xxxx
15:                         Beginning track number:    1
16:                              Number of tracks:  511
17:                    Disk size in Kbytes is 4336
18:  Number of directory entries (multiple of 256):  256
19:                  Check for changed disks (Y/N):    N
20:                       Memory used = 107 bytes
21:                                    OK (Y/N)? Y<enter>
22:
23:
24:
```

Figure 7.22 Display with all Parameters Entered

Figure 7.23 shows the console display after the all of the parameters have been entered for a second hard disk drive. Note that the amount of space used for the disk parameters for this drive is less than for drive E because this drive is defined to have 16 Kbyte allocation blocks.

```
1:      P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
2:
3:            Number of logical drives defined:     hard:  1     floppy:  3
4:  working on drive: ABCDEFGHIJKLMNOP                 Total space used: 586 bytes
5:                    fff-hh----------
6:
7:                              Hard Disk Drive Parameters
8:
9:            Drive is initially accessible (Y/N):    Y
10:           Drive is initially read/write (Y/N):    N
11:     Allocation block size in Kbytes (2,4,8,16):   16
12:                        Physical drive number:     0
13:
14:   Maximum number of tracks on drive 1 is xxxx
15:                         Beginning track number:  512
16:                              Number of tracks:  512
17:                    Disk size in Kbytes is 4352
18:  Number of directory entries (multiple of 512):  512
19:                  Check for changed disks (Y/N):    n
20:                       Memory used =  73 bytes
21:                                    OK (Y/N)? Y<enter>
22:
23:
24:
```

Figure 7.23 Defining a Second Hard Drive

Part 3 - Installing the New Module in the Library

After all of the drives have been defined, you use the "X" command to tell HDCONFIG that you have finished entering drive information. HDCONFIG then makes a few last checks on the parameters that have been entered and, if no

errors are found, displays the module library as shown in Figure 7.24. The names of the disk table modules are displayed in reverse video.

```
1:      P&T CP/M 2 (Corvus Hard Disk) System Configuration Program - Ver 3.xxx
2:
3:            Number of logical drives defined:     hard: 3     floppy: 2
4:  working on drive: ABCDEFGHIJKLMNOP                Total space used: 615 bytes
5:                    hhhff----------
6:
7:                              Library Table of Contents
8:
9:                          Disk Table Modules are shown reverse
10:
11:       AUTOKEY  KEYXLATE SCRNDUMP ADM3     4FLOPPY  3FLOPPY  2FLOPPY  1FLOPPY
12:       TYPENAME COREINIT SIO      PPMIN    PPNORS   PPSTDR   PPSTD    M12CRT1
13:       STDCRT1  M12WB1   STDWB1  STDCORE1 CORV1    CORV2    HDTB10   HDTB20-1
14:       HDTB20-2 HDTH6    HDTH10  HDTH20-1 HDTH20-2 HDTH20-3 MYHD
15:
16:
17:
18:
19:
20:
21:
22:                  Enter name for modified disk table module: MYHD<enter>
23:       Name should not be in above list unless you intend to replace that module
24:
```

Figure 7.24 Entering Name for the New Disk Table Module

On Line 7.24-22, HDCONFIG asks you for the module name under which you want to save the configuration you have created. You may specify a new name or the name of one of the existing disk table modules. In the latter case, the existing module will be replaced by the new one being created. Since this deletes the existing module, HDCONFIG requires that you verify that a replacement is to take place by asking the question shown in Figure 7.25. If you respond affirmatively, HDCONFIG will continue. A negative response will cause HDCONFIG to reprompt for a module name.

```
MYHD      already exists, replace it (Y/N)? Y<enter>
```

Figure 7.25 Message If a Module Already Exists

After you have entered a name for the new disk table module, HDCONFIG will let you enter a description of the module. The display will appear as shown in Figure 7.26. You should enter any notes that will help you (at a later date) to remember what the configuration is and why it was created. A text area of 19 lines of 80 characters each is provided for entering your description. While you are entering the description, HDCONFIG functions as a simple text editor. As a convenience, HDCONFIG allows you to read the text description from any module in the library and then edit it. This is particularly useful when you are just making a few changes to an existing disk table module.

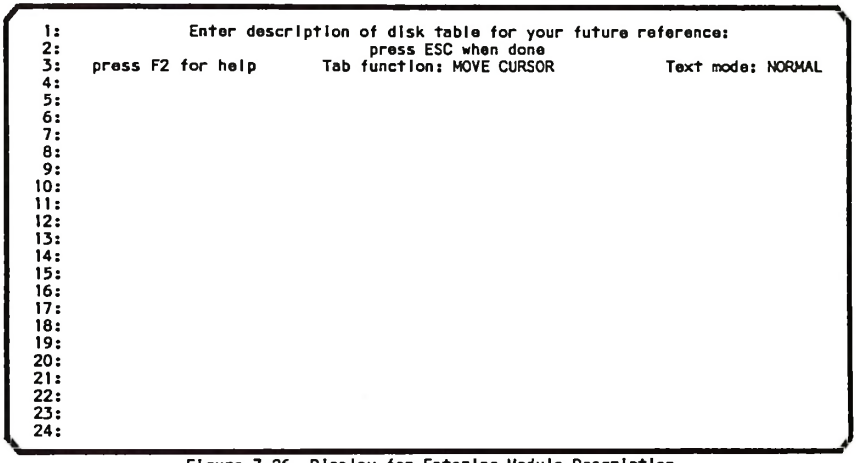

Figure 7.26 Display for Entering Module Description

Line 7.26-1 reminds you that you should be entering a description of the disk configuration you are generating so that you can refer to it in the future. Line 7.26-2 indicates that you should press the <esc> key when you are finished entering the description. Since you may not remember all the editing functions that are available, HDCONFIG has a help function available. By pressing <f2>, HDCONFIG will display a summary of the editing functions as shown in Figure 7.27. The left end of Line 7.26-3 reminds you that this help is available.

While accepting the description of the module, HDCONFIG has two modes, NORMAL and INSERT. While in the NORMAL mode, any character you type (except for control characters) will overwrite the character at the cursor position. The cursor is then moved to the next character position on the line. In the INSERT mode, characters you type are inserted into the text at the cursor location. All characters to the right of the inserted character are moved one space to the right to make room for the new character. Characters that move off the right end of the line will be lost.

```
HDCONFIG Text Editor Command Summary

Cursor Movement Commands

Cursor Left    - <ctl-d> or <left arrow>
Cursor Right   - <ctl-s> or <right arrow>
Cursor Up      - <ctl-w> or <up arrow>
Cursor Down    - <ctl-x> or <down arrow>
Cursor Home    - <ctl-q>
Start of Line  - <ctl-a> or <F1>
End of Line    - <ctl-f>
Start New Line - <enter>

Delete Text Commands

Delete Character at Cursor      - <ctl-t> or <ctl-k>
Delete Character left of Cursor - <backspace>
Delete Right to Next Space      - <ctl-y>
Delete Right to End of Line     - <ctl-u> or <ctl-j>

Text Mode (Insert-Normal) Toggle - <ctl-v>
Read Text from Existing Module for Editing - <ctl-r>

(press <enter> to continue)
```

Figure 7.27 HDCONFIG Editor Help Display

While entering the text description you can use the arrow keys on the keyboard to move the cursor around the screen. If you attempt to move the cursor out of the area allotted for the text, the bell (if there is one) will sound and the cursor will not be moved. As shown in Figure 7.27, <ctl-D>, <ctl-S>, <ctl-W>, and <ctl-X> are equivalent to <left arrow>, <right arrow>, <up arrow>, and <down arrow> for moving the cursor.

HDCONFIG also provides functions for moving the cursor more rapidly. Pressing <enter> will move the cursor to the beginning of the next line in the text area. Pressing <ctl-Q> will move the cursor to the upper left-most (home) position of the text area on the screen. <ctl-A> or <F1> will move the cursor to the first character position of the line. <ctl-F> will move the cursor to the first character position past the last displayed character on the line. This function is very useful for adding to the end of a line.

The function of the <tab> key differs depending on whether you are NORMAL or INSERT mode. In NORMAL mode, pressing the <tab> key performs a cursor movement function; it will move the cursor to the next tab position on the line. Tab positions are at every eighth column. In INSERT mode, the <tab> key will insert one or more spaces starting at the current cursor position and continuing until the cursor is positioned at the next tab stop. Thus in the INSERT mode, <tab> performs an insert function. The current function of the <tab> key is displayed as shown on Line 7.26-3 while you are entering your description.

There are also several editing functions that allow you to delete one or more characters from the text. Pressing <ctl-T> or <ctl-K> will delete the character at the current cursor position. After the deletion, any remaining characters to the right are moved to close up the resulting space. Pressing <backspace> will delete the character to the left of the cursor position and close up the space as for <ctl-T> and <ctl-K>.

The <ctl-Y> key will delete any characters from the cursor position up to, but not including, the next space. The cursor position is not changed. The character under the cursor is deleted and is not checked when looking for a space to terminate the

deletion. This means that if you want to delete an entire word, you should position the cursor at the space before the word and then press <ctl-Y>. If you want to delete only the ending of a word (the "ing" on "starting", for example), you should position the cursor to the "i" and press <ctl-Y>.

The <ctl-U> or <ctl-J> key will delete all characters from the cursor position to the end of the line. The cursor position is not changed.

The <ctl-V> key will cause HDCONFIG to toggle back and forth between NORMAL and INSERT mode. This means that if you are in NORMAL mode and press <ctl-V>, HDCONFIG will switch to INSERT mode and vice versa. Line 7.26-3 will be changed as you press <ctl-V> to remind you of which mode you are in and the function of the <tab> key.

The <ctl-R> key allows you to read text from an existing module in the library so that you can edit it. Note that if you read text from a module, that text will replace any text already present in the description. Thus, if you have entered some text and then decide to read the text from a module, the text that you have entered will be lost. If you want to edit the text from an existing module and add to it, you should read the existing text first and then make the additions.

If you press the <ctl-R> key, HDCONFIG will clear the display and show you the table of contents of the module library as shown in Figure 7.24. It will then ask the question shown in Figure 7.28. You may then enter a module name or just press <enter> to return to editing. After you enter a module name, HDCONFIG will read the text description of that module, place it in the text area of the editing display (Figure 7.26) and allow you to start editing it.

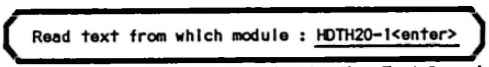

Figure 7.28 Entering Module Name for Reading Text Description

Figure 7.12 shows a typical description for a disk table module.

After you have finished entering the text description for the module, press the <esc> key. HDCONFIG will then ask you if you are ready to install the new disk table module in the module library as shown in Figure 7.29. This is your last chance to abort the program by pressing <break> or entering a negative response. If all is OK and you want to continue, enter an affirmative response. HDCONFIG will then install the module in the library, displaying the message shown in Figure 7.30 while doing so.

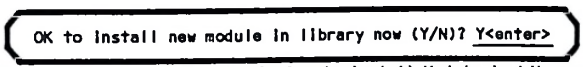

Figure 7.29 Prompt for Permission to Install Module in Library

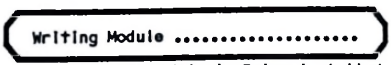

Figure 7.30 Display While Module is Being Installed in Library

After the module has been written to the library, HDCONFIG will return to the command level of the system.

REMINDER: For this new disk table module to be used by the system, use MODSEL or the SM option of MENU to select it for inclusion in the system. Then reset the computer so that the system (including the new module) is loaded into memory.

7.7 Possible Error Messages

General

Not a valid response, please re-enter
This is a general error message which indicates that something is wrong with the response that you have just given. The message appears by itself when the reason for it is obvious (like answering "Q" to a Y/N question) or in conjunction with other error messages if additional explanation is needed.

Not enough memory to run HDCONFIG
HDCONFIG requires about 48 Kbytes of memory to execute. If you have reserved a lot of memory above the operating system, you may encounter this message; under normal circumstances it should not occur. If you get this message you should run HDCONFIG on a system that reserves less (or no) memory above the system.

**HDCONFIG overlay files are not present on the current disk drive.
All of these files must be on the current drive when you execute HDCONFIG:**
HDCONFIG.COM, HDCONFIG.001, HDCONFIG.002, HDCONFIG.003
If you see this message, you probably tried to execute HDCONFIG from another drive. You must make the drive with all of the HDCONFIG files current before executing the program. This can also be caused by not copying all of the HDCONFIG files when making a working disk. HDCONFIG will return directly to the command level of the system after this error.

While Entering Physical Drive Information

Enter a drive letter A to P
Indicates that an invalid letter was entered in response to a query. Re-enter using one of the valid letters.

Drive X: does not exist
Indicates that the logical drive letter you entered, while valid, corresponds to a drive that does not exist on the system.

Cannot find X:BIOSMODS.PNT
Indicates that the module library (BIOSMODS.PNT) could not be found on the drive you specified (X). Make sure you entered the correct drive letter.

Error re-opening library
HDCONFIG closes and reopens the library file several times during execution. This message indicates that one of these reopen operations failed. The most common cause of this error is changing the disk on which the library is stored. It can also be caused by hardware problems.

**X:BIOSMODS.PNT has no more room for additional modules.
You will be required to replace an existing module with
the new module you are creating.
<break> will exit the program.**
If the module library is full, you cannot add a new module to it. This message is given at the beginning of HDCONFIG as a warning that, if you continue, you must replace one of the existing modules. It is very unlikely that you will ever run into this problem since the module library can hold up to 79 modules.

Disk read/write error (N)
This message indicates that some error occurred while HDCONFIG was reading or writing to the module library file. The "N" will be replaced by a number that indicates what might have caused the error as follows:

1,4,6 May indicate that there is something wrong with the module library file. Something may have been damaged. Try using a copy from a backup disk.

3 May indicate a hardware problem or that the disk has been changed.

5 The directory on the disk is full and another entry is needed. You should abort HDCONFIG, delete any unnecessary files from the disk and then run HDCONFIG again.

xxxxxxxx has no text
This message is displayed if you try to read the text description from a module that has none.

xxxxxxxx not in library, enter another name
You have entered the module name "xxxxxxxx" for some operation and HDCONFIG could not find it in the library. Check the table of contents to be sure you entered the name correctly.

xxxxxxxx is not a disk table module, enter another name.
When HDCONFIG asked for a disk table module name for some operation, you entered the name of a module that is not such a module. Check the table of contents display; the disk table modules are shown in reverse video.

While Defining Logical Drives

All physical floppy drives are assigned
This message is displayed if you attempt to assign a logical drive to a diskette drive after all of the physical diskette drives have been assigned to logical drives.

This logical drive is not defined as a physical floppy drive
This message is displayed if you attempt to define a logical drive on a diskette drive when it was already defined on a hard drive. If you want to change a logical drive from a hard to a diskette drive, you must first delete its definition as a hard drive and then redefine it as a diskette drive. Alternately, you can use the "S" command to swap the floppy definition with a hard disk definition for another logical drive.

You may specify only physical drive 0 on a single floppy system
If you indicated that you you have only a single diskette disk drive you may only specify physical floppy 0 for a logical drive defined as a floppy.

Physical drive N is already defined as logical drive X
Indicates that you have tried to redefine a physical diskette drive on a system that has multiple physical diskette drives. "N" will be replaced by the physical drive number specified and "X" will be replaced by the logical drive letter to which it is already defined. You may change a definition using the Move and Swap commands.

You indicated that there were only N floppy drives on the system
HDCONFIG assumes that all physical diskette drives on the system will have consecutive physical drive numbers starting with 0. At the beginning of HDCONFIG you indicated that there were "N" physical diskette drives on the system. This message indicates that you entered a physical drive number greater than N-1 for a diskette drive.

You have indicated that there are only N hard disk drives on the system
HDCONFIG assumes that all physical hard disk drives on the system will have consecutive physical drive numbers starting with 1. At the beginning of HDCONFIG you indicated that there were "N" physical hard disk drives on the system. This message indicates that you entered a physical drive number greater than N for a hard disk drive.

You have specified that there are no hard drives on the system
This message is given if you attempt to define a logical drive on a hard disk when you originally indicated that there were no hard disk drives on the system.

Drive A: must be accessible
Logical drive A: must be accessible at all times (it may be read only, if you wish). This messages indicates a drive A: is defined on a hard disk and that you tried to make it initially inaccessible. This would result in an inoperative system, hence HDCONFIG does not allow you to make the definition.

These specifications exceed the number of tracks on this drive
Indicates that the beginning track number and number of tracks you specified for this hard disk drive combine to exceed the number of tracks on the hard disk.

This logical drive overlaps the tracks defined for logical drive X
HDCONFIG does not allow you to specify a track range for a hard disk logical drive that will conflict with another hard disk logical drive. This message indicates that such a conflict was found with the logical drive "X".

The number of directory entries must be a multiple of NNN
The directory must occupy one or more complete allocation blocks on the disk. As a result the total number of directory entries must be a multiple of the number of entries that will fit in one allocation block. This message indicates that the number of entries you specified is not such a multiple. "NNN" will be replaced by the number of entries that will fit into one allocation block.

Too many directory entries for allocation block size
A maximum of 16 allocation blocks may be used for directory space. This puts an upper limit on the number of directory entries that may be specified for a logical disk drive. For example, a 4 Kbyte allocation block will hold 128 directory entries. Hence a drive with 4 Kbyte allocation blocks will be limited to 2048 directory entries. You will probably never run into this limit since it is highly unusual to require more than 512 directory entries on a drive.

This logical drive is not defined as a physical hard disk drive
This message is displayed if you attempt to define a logical drive on a hard disk drive when it was already defined as a floppy drive. If you want to change a logical drive from a diskette to a hard drive, you must first delete its definition as a diskette drive and then redefine it as a hard drive. Alternately, you can use the "S" command to swap the hard disk definition with a floppy definition for another logical drive.

This logical drive is not defined - cannot edit
Indicates that you have attempted to edit the drive parameters of a logical drive that has not yet been defined.

You have not defined a logical drive for physical floppy drive 0
Physical diskette drive 0 must be defined as a logical drive. (Other physical diskette drives need not be defined as logical drives.) This message is given if you attempt to have HDCONFIG install the new module in the library before defining a logical drive for physical diskette drive 0.

You must define N logical drives on the floppy drive
If you have a single floppy system and indicated that you want to define more than one logical drive to the floppy, HDCONFIG will issue this message if you fail to define enough logical drives to the floppy. After pressing <enter> you may define additional logical drives.

Logical drive A is not defined
This message indicates that you have failed to define logical drive A. Logical drive A is required in order for CP/M to function. After pressing <enter> you may make the necessary definition.

WARNING: you have not defined logical drives for all of the physical floppy drives, is this correct (Y/N)?
HDCONFIG issues this warning if you do not define logical drives for all the physical floppy drives you initially indicated are present on the system. While this situation is not an error, it may indicate that you have made a mistake in defining the logical drives.

The range of tracks you have specified gives a drive larger than 8192 Kbytes
The maximum logical drive size that CP/M 2 can accommodate is 8192 Kbytes. You should reduce the number of tracks for the drive until its size is less than or equal to 8192 Kbytes.

Cannot move to a defined drive, use swap
You have tried to use the Move command to move the definition of one logical drive to another that is already defined. You cannot do this. You should either delete the definition of for the logical drive you are trying to move to or use the Swap command to exchange the definitions.

Cannot swap with an undefined drive, use move
You have tried to use the Swap command to move the definition of one logical drive to another that is not defined. Swap can only be used when both logical drives involved are already defined. To move a definition to an undefined logical drive, use the Move command.

Cannot move or swap an undefined drive
You have tried to use the move or swap command when the cursor is positioned at a logical drive that has not yet been defined. You must move the cursor to a logical drive that is defined before using either of these commands.

Workspace is exhausted, install the partial module in the library and then re-execute HDCONFIG, read the parameters from the module and continue.
Although it is very unlikely, if you make a lot of changes while entering your disk configuration, HDCONFIG may run out of workspace in memory. Under these conditions, HDCONFIG will display this message any time you attempt an operation that requires more workspace. If this occurs, you should install your partially completed module in the library. You can then run HDCONFIG again,

read the parameters out of the partial module, and continue to make your changes and additions.

While Installing Module in Library

Error re-opening library
See explanation under **While Entering Physical Drive Information** above.

Disk read/write error (N)
See explanation under **While Entering Physical Drive Information** above.

xxxxxxxx has no text
See explanation under **While Entering Physical Drive Information** above.

xxxxxxxx not in library, enter another name
See explanation under **While Entering Physical Drive Information** above.

xxxxxxxx is not a disk table module, enter another name
See explanation under **While Entering Physical Drive Information** above.

Invalid module name, please re-enter
Indicates that the name you specified for the new module contains invalid characters (control characters and the special characters not allowed in a CP/M file name). You should invent another name that does not use any of these characters.

xxxxxxxx already exists, replace it (Y/N)?
Indicates that the module name you have specified for the new disk table module already exists in the module library. In this case HDCONFIG wants confirmation that you really want to replace it before doing so. A positive response will allow the replacement to proceed while a negative response will cause HDCONFIG to ask for another name.

Table of contents is full. xxxxxxxx is not in table of contents.
Enter the name of the module to replace with xxxxxxxx:
This message will occur only in the rare case that the library file is full (it will hold 79 modules). In this case you must replace an existing module with the new module you are creating. "xxxxxxxx" will be replaced with the name you have specified for the new module. If you had specified the name of a module that already exists in the library for the new module, the existing module will be replaced. If no module with the name you specified appears in the library, HDCONFIG will prompt with this message for the name of an existing module that may be deleted to make room for the new one.

xxxxxxxx not in library, enter another name
This message occurs if, in response to the previous message, you directed HDCONFIG to replace a module that does not appear in the library. "xxxxxxxx" will be replaced with the module name that you specified to be replaced.

Cannot replace a module that is not a disk table
You have directed HDCONFIG to replace a module that is not a disk table with the disk table module you are creating. This is not allowed; specify a disk table module (names shown in reverse video) or use a new name.

Cannot replace a permanent module, use another name
You have given HDCONFIG the name of a permanent module to be replaced by

the disk table module you are creating. Permanent modules cannot be replaced or deleted from the library. You should try another module or use a new name.

Error while closing BIOSMODS.PNT

This message indicates that an error occurred while HDCONFIG was trying to close library file. This may occur if a disk is changed while HDCONFIG is running.

NOTES

8.1 System Error Messages

This section describes the error messages associated with the hard disk that can be given by the Corvus Hard Disk version of P&T CP/M 2.

Hard disk lock out error, code = HLxx
This message is displayed when an attempt is made to access a logical drive on the hard disk to which access is currently not available. "xx" is a hexadecimal representation of the logical drive number (00=A...0F=P) to which access was attempted. See Chapters 5 and 7 and Section 6.2 for information about limiting access to logical drives on the hard disk.

Hard disk write prot error, code = HP00
This message is displayed if an attempt is made to write to a logical drive on the hard disk which has been set for "read only" access. See Chapters 5 and 7 and Section 6.2 for information about limiting access to logical drives on the hard disk.

Hard disk not ready error, code = HNxx
This message is displayed when a read or write access is made to a hard disk drive that is not on-line and ready. This is typically caused by the drive being turned off or trying to access a nonexistant drive. You can use the ERROR program to get a further explanation of the error code.

Hard disk - - - - - error, code = xxxx
This is the form of the message given for a general hard disk error. The "- - - - -" will be replaced by a word or phrase describing the type of error (eg. read, write, etc.). "xxxx" will be replaced with an error code that indicates the nature of the error. You can use the ERROR program to get an explanation of the error code.

NOTES

9.1 Introduction

The Corvus version of P&T CP/M 2 provides all the necessary support for a single user CP/M system to access the Corvus hard disk system. If multiple micro-computers are to be connected to the Corvus system via the Constellation some precautions must be taken in order to avoid problems caused by multiple users accessing a single logical drive.

This chapter discusses some of the problems that exist in multicomputer access to the hard disk and describes some strategies for dealing with them.

First it is necessary to understand the distinction between logical and physical drives. A **physical drive** is the actual piece of hardware that stores the information. The Corvus controller is capable of running up to 4 physical drives, although many systems will be configured with only one. Each physical drive may be divided into one or more **logical drives** (sometimes called **pseudo drives**). In making this division, the disk is divided up into regions each of which is referred to as a separate logical drive. For example, a common way of dividing a 10 Mbyte drive is to treat it as two 5 Mbyte logical drives.

The simplest and safest way of dividing a physical drive among multiple users is to assign to each user one or more separate logical drives on the physical drive. Since the logical drives do not overlap, there is no way that one user can interfere with another. Unfortunately much of the motivation for sharing a disk drive is to allow several users to share common programs and data files so this simple technique is not very attractive.

When users share a common logical disk, problems usually arise only when more than one user is writing to the disk. A discussion of the nature of these problems will follow shortly. The potential problems may be avoided if the shared areas of the disk are treated in such a manner that only one user may write to it.

This user would be the "system administrator" and would be responsible for placing programs or other files on the disk to be used by others. A situation where this approach might be attractive is where a common drive is used for frequently used utility programs such as editors, assemblers, and compilers so that copies of these programs do not need to be maintained on each logical drive. Another situation where this arrangement could be useful is in a case where a data file is accessed by multiple users but only one user has the authority to modify it. This second case is subject to a few restrictions which will be discussed shortly.

All manner of problems arise when multiple users must have read/write access to a common logical drive. In order to understand the problems, it is necessary to understand how CP/M uses disk storage. Section 7.2 presents a discussion of this topic. You should refer to this section for an explanation of what allocation blocks are and how they are used.

In particular, you should be aware that CP/M keeps a map of the free allocation blocks for an active disk drive in the computer's memory. This map (the allocation vector) is initialized when the drive is first activated (logged on) and is updated as allocation blocks are assigned to files (as files are written) or released (as files are deleted). The integrity of this table is vital because if it is altered, the system may think an allocation block is free for use when, in fact, it has been already

assigned to another file.

You should also be aware that when a disk file is opened or created under CP/M, information about what allocation blocks have been assigned to the file is copied from the disk directory to the computer's memory. It is held in memory until something occurs that forces it to be written back out to the disk directory. The directory information is written back to the disk when a file is closed and under certain other circumstances that are not of great importance to this discussion. The important point is that the directory may not be in complete agreement with the actual usage of the disk until all open files that have been written to are closed.

9.2 Blocking and Deblocking

CP/M always deals with a disk on the basis of **logical sectors** (sometimes called CP/M sectors) of 128 bytes each. Whenever CP/M reads or writes to a disk, it is done in "chunks" of 128 bytes. However many disk drives allow more storage and faster access as the sector size is increased; hence it is often desirable actually to use larger sectors on the disk. The actual sectors on the disk drive are often referred to as **physical sectors.**

In order for CP/M to work with disk drives that have physical sectors larger than 128 bytes, it is necessary for some additional work to be done in the disk I/O drivers to make the disk look like it has 128 byte sectors. This is done by reading a physical sector from the disk into a special buffer and then transferring only a 128 byte portion of the physical sector to CP/M. For example, during a write operation, the physical sector is read from the disk into a special blocking buffer and the 128 bytes of the logical sector are transferred into the buffer at the appropriate location.

In order to optimize disk access the disk I/O routines do not always write the contents of the blocking buffer back to the disk immediately after a logical sector is "written" from CP/M. Since disk accesses are frequently sequential in nature, considerable improvement in performance is realized by holding the contents of the buffer in memory until it is determined that the buffer is needed to access another area of the disk. When this situation occurs and if new data has been placed in the buffer by a write operation, the buffer is written back to the disk before the buffer is reused.

In normal operation this deferring of the actual write operation to the disk does not cause a problem since the disk drivers make sure the buffer is clear before using it. However, since the write to the disk may be deferred, the data "written" out by a program may not immediately appear on the disk. In some circumstances, this may cause a problem.

For example, if multi-user access to a disk is used to provide communication between users, one user may "write" something to the disk but since the information gets held in the blocking buffer (and may not get immediately written to the disk) a second user may not see the information which the first user thinks he has written. If the communication is for the purpose of synchronizing programs or other time dependent operations, difficulties can result.

When it is important that data "written" out by a program be immediately placed on the disk the program must take special care to insure that the blocking buffer is written to disk as soon as possible after the output to the disk file was done.

One way of doing this is as follows: the program should open the file to which it will be writing and open a second dummy file that is used solely for the purpose of forcing the buffer to be written out to the disk. The dummy file should have at least a small amount of readable data in it. After writing to the output file, the program should immediately read something from the dummy file. What is read is unimportant. The effect of this procedure is to create a situation where the disk I/O drivers need to use the buffer for accessing another physical sector on the disk immediately after writing to the output file. Since a write to the buffer has been done and since the buffer is needed to read another physical sector, the buffer will be written to the disk.

9.3 Semaphores

When planning the approach to take to a multi-user system with shared disk drives, it is important to be aware of a few of the operational features of the Corvus Constellation. The Constellation is a multiplexer that switches rapidly between several different users of the Corvus disk system.

In order to assure that a single user cannot jam up the system by hanging onto the disk controller for a long period of time, the Constellation imposes certain limits upon the time allotted to each user. Each user is permitted a maximum number of disk transfers and a maximum time between successive transfers before the Constellation moves on to the next user. See your Corvus documentation for information about these limits. With the typical limits imposed by the controller, virtually no program that uses standard operating system calls can make successive disk transfers within the allotted time. You must assume that only one transfer will occur before control of the disk system passes on to the next user.

A typical type of disk operation, that of updating a record in a data file, involves reading the data from the disk, modifying it, and writing it back to the disk. Since it is unlikely that the program making the update will retain control of the disk system between the read and write operation the possibility exists of another program accessing the same record before the first program had completed its modification. Obviously this is an undesirable situation that may cause problems.

The Corvus disk system provides a means of communication between users of the system in the form of **semaphores.** These semaphores can be thought of as a series of named switches which can be set and cleared by any user. A unique feature of the semaphores is that only one user can access a semaphore at a time. When a user attempts to set or clear a semaphore, there is no chance of another user accessing the semaphore before the first user is done with it. This is not the case with disk files as will be discussed in a subsequent section.

Each semaphore is identified by an 8 character name. The names may consist of any 8 characters and are generally chosen by the programmer setting a system up for multiple user access. Typically all programs that need to communicate with one another for a specific purpose will do so by means of a specific semaphore name. For example, a semaphore used by various programs to provide a means of locking other users out from using a given logical disk drive may be named DISKLOCK.

A user may request a given semaphore to be set (sometimes referred to as locking the semaphore) or cleared (sometimes referred to as unlocking the semaphore). When a semaphore operation is performed, a code is returned to the program that requested the operation which indicates the status of the semaphore previous to the requested operation. When a set operation is requested, the disk controller will set the semaphore if it was previously clear and return a code indicating that the

semaphore was previously clear. If the semaphore is already set when a set operation is requested, no action is taken and the disk controller returns a code indicating that the semaphore was previously set.

If a user wants to set a semaphore, he requests a set operation on it and if the returned code indicates a previously cleared condition, it means that he has succeeded in setting the semaphore and that he should continue on that basis. If the returned code indicates that the semaphore was previously set, it means that the set operation was ignored. Typically a program would attempt to set a semaphore and, if it was already set, assume that another program had set it. In such a case, the program might continually retry to set the semaphore until it succeeds and then continue on with what it wants to do.

A note of caution is in order regarding semaphores: The clear operation can be performed by any user (not just the one who set the semaphore) at any time. Thus it is necessary that all the programs that use a semaphore "play by the rules". That is, a given program should not indiscriminately clear a semaphore. It was probably set by someone else and having it unexpectedly cleared may cause problems. A good rule to follow is that a program should clear only a semaphore that it had successfully set. Since there is no hardware to enforce a semaphore to be cleared by a particular user, close attention should be paid to the parts of programs that set and clear semaphores to insure that they work properly. A semaphore that is set by one program and then cleared by another can cause errors in a system that are difficult to track down.

In order to use the semaphores, you must write interface routines that communicate directly with the Corvus disk controller. See the Corvus documentation for information on doing this. P&T CP/M 2 does not interfere with the Corvus Semaphores in any way.

9.4 Pipes

The Corvus controller (rev B and rev H) provides a second method of communication between different computers connected through the Constellation called Pipes. These pipes allow actual messages to be passed from one computer to another. In designing multi-user access software, the pipes may be of use for passing information about what one computer is doing to other computers on the system.

As with semaphores, you must write interface routines that communicate directly with the Corvus disk controller in order to use pipes. See the Corvus documentation for information on doing this. One special note is necessary: The pipe mechanism requires an area on the hard disk to store the messages until they are read by someone. You must insure that the area used for pipes does not conflict with the areas assigned to logical drives on the hard disk. The best way to do this is to decide on what area on the hard disk is to be used for the pipes and then custom configure your system (see Chapter 7) such that no logical drive is assigned to that area.

9.5 Pre-allocated, Fixed Length Files

The most straight forward approach to multiple user read/write access to a logical drive is to preallocate all files that will be written to. For purposes of example let's assume a single data file on a shared logical disk which which is to be randomly read and written by multiple users.

In the process of setting up the programs which make use of this file, a decision as to the maximum length of this data file must be made. Once the maximum length is decided, a special initialization program is written to create the file. This program creates the file and then writes out to the file the total number of records that the file is to hold. Depending on the programs that are to make use of the file, each record might contain such items as a flag to indicate whether the record has had valid data written to it or not. When the file is first created, all records that are written to it would have this flag set to indicate no valid data in the record. As the file is used and data is written to it, the flag in each record would be changed as data is written to it to indicate that the record now contains valid data.

The advantage of this approach is that the directory entries on the disk drive and the allocation blocks that are in use will never be changed by use of the file. The directory entry (or entries) for the file and the allocation blocks assigned to it are fixed when the file is first created. Since none of the programs that use the file will be extending it (remember the maximum file size was decided beforehand and it is the duty of each program using the file to live within the specified size), no changes will be made to the blocks allocated to the file or its directory entry (or entries). As long as no attempt is made to increase the size of the file, many programs can access the same file and read/write to it without problems.

One note of caution is in order, however. It is the responsibility of the programmer to be certain that one user does not make use of a record from the file while another user is in the process of modifying that record. One way of doing this is to add another flag to the record that is set whenever a user is in the process of updating that record. Other users' programs must check that flag before using the data in that record to determine if the data is stable or not. When such flags are used, it is necessary to take precautions to insure that the flags are actually set before another user has a chance to check them.

9.6 Drive Lockout

In the case that predefining the disk files on a shared drive is not desirable it is possible to have read/write shared access to a logical drive if it is acceptable to have only one user have access to a drive at a time. This is an application for the semaphores which are part of the Corvus controller.

The general methodology for using the semaphores is as follows: All of the programs and/or users of a shared drive will agree on a particular semaphore name for the purpose of gaining access to the shared drive. When a program wants access to the drive, it will attempt to set the semaphore. If the semaphore is already set, another user is in possession of the drive, and the program will have to try again later to gain access to the drive. If the semaphore was not set, the program now has possession of the drive and may perform any operation on it.

When a program is finished on the drive, it should close any files it has been writing to and clear the semaphore. This will allow other uses to gain access to the drive.

The checking of the semaphore can be done by a program but this will require making some modifications to the program. Changing the program is probably the best approach if it can be done, but in some cases (especially when using purchased software) the program cannot be modified. In this case a simple program could be written which will allow a user to check semaphores directly.

For example, assume a user wants to use a text editor to modify a file that is kept on a shared drive. Since the text editor knows nothing about semaphores and the like and it assumes a "standard CP/M" system where there is only one user, it expects to have sole possession of the drive. In this case the user would first execute a special sign-on program that would check the semaphore for the drive to which access is desired. If the drive is available it would report to the user that he now has access to the drive and may make use of it. If the drive is currently in use, the program would inform the user of that fact and suggest that he try again at a later time. The sign-on program could even make use of the "read only" and "no access" features that have been added to P&T CP/M 2 (see Chapter 5 of this addendum and Chapter 16 of the P&T CP/M 2 User's Manual) to insure that the user cannot make use of the drive until he has successfully logged on to it.

Note that with this technique of multiple user access, only one user has access to the drive at a time. While the one user has possession of the drive, all other users are not allowed to write to it. It is possible to allow other users to have read access to the drive while one user has possession of it if they are making use of files that the user in possession of the drive is not modifying. The period of time a given user retains possession of the drive depends on the program he is running and what he is doing but typically might run from a few minutes to hours.

Also note that before relinquishing possession of the drive, the user or program should close all files that have been created and/or written to. This will insure that the directory is completely up-to-date. If this is not done the directory will not accurately reflect which allocation blocks on the disk are in use and which are free. If an allocation block is assigned to a file generated by one user, and the directory does not get updated to reflect the assignment, another user might use the same allocation block and thereby destroy the data stored there by the first user.

After gaining access to a disk drive, it is necessary that a disk system reset be done by the user or program that gained access to the drive. This forces the system to rebuild the allocation vector for that drive which will insure that any changes that might have been made by other users will be reflected in the vector. If the user gained access to the drive manually via a sign-on program (as described above) he need merely perform a warm boot (break or control-C) to insure the system is reset (if the sign-on program makes a normal return to CP/M it will do so by means of a warm boot and no further action will be required on the part of the user).

Most high level languages provide some means of performing this reset function (many versions of BASIC have a function called RESET for this purpose) but frequently they will require that all files be closed before the function is performed. For assembly language programs, BDOS function 13 can be used to reset all disks on the system or BDOS function 37 can be used to reset only the drive that is being accessed.

9.7 File Lockout

It is possible, by the use of semaphores, to provide file lockout within a shared drive. The file to be shared in the manner should be preallocated as described in Section 9.5 to insure it can be written to without affecting other users who are accessing the same drive while one user is writing to the file.

The technique for using the semaphores is very similar to that described in the previous section for drive lockout. All the programs that will make use of the

shared file will agree on the name (perhaps the same as the file name) for a semaphore that will be used to control access to the file. If a program wants to make use of the file, it will attempt to set the semaphore to indicate that the file is in use and that other programs should not attempt to access.

If the code returned by the attempt to set the semaphore indicates that the semaphore was previously not set, the program now has possession of the file and may make use of it. If the code indicates that the semaphore was previously set it means that another user has control of the file and that the program should not use the file at this time. In this case, the program should repeatedly try to set the semaphore until the returned code indicates that the semaphore was previously clear.

With this technique, a program should not even try to read from the file in question when the semaphore indicates that another program is using it since there is no way of knowing what the other program is doing. Reading from the file while another program is writing to it may result in getting data that is in the process of being updated and is not currently valid.

In order to minimize the time a program spends waiting for the file to become available, each program that is using the file should retain possession of the file for as short a time as possible. For example, a data file could be constructed in which each record has an internal flag which indicates whether the record is currently being updated by a user or not.

If a program wants to update a certain record, it would gain access to the file (by setting the semaphore as described above) read the record to make sure that no other user is updating the record, and then write the record back with the flag set. After writing the flagged record back to the disk, it would release access to the file while actually processing the record by clearing the semaphore. When another program finds a record that is in the process of being updated it would, most likely, not make use of the information in the record until the update is complete.

Special note: When using flags within files to communicate between programs, be sure to take precautions to be sure that the disk I/O driver's blocking buffer is written back to the disk before releasing the file to another user. If this is not done, the flag may not actually appear in the file on the disk before another program gains control of the drive. For a discussion of how to accomplish this see Section 9.2.

This technique of file sharing has the disadvantage that only one user can have access to the file at a time. This holds for both reading and writing since in order to read from the file a user must first insure that no other user is writing to the file. The only way to find out if another user is writing to the file is to attempt to set the semaphore associated with the file. If the semaphore was previously clear indicating that no user is writing to the file, it will still end up being set and it must remain set while the program reads data from the file to insure that another user does not start writing to the file while the first user is reading from it.

9.8 Record Lockout

In some cases, it might not be desirable for a program to lock out an entire file from access by other programs for the entire time it needs to access the file. In this case some mechanism needs to be developed to provide access control on a record-by-record basis. This cannot be done directly via the semaphores in the

Corvus controller however the pipes may prove useful.

One way of providing this type of access control without using pipes is to have a small disk file that contains only control flags for a larger data file. This flag file might contain information about each record in the data file indicating whether it is in use or not and which program is using it if it is in use. When a program wants to have control of a record in the data file it would check the flags associated with that record in the flag file. If the flags indicate that the record is available, the program would set them to indicate that it has control of the record. Access to the control file should be regulated with a semaphore as described in the previous section.

Using record lockout is probably the most difficult form of multi-user access under CP/M 2. It requires very careful planning and programming in order to work properly.

A.1 Additional Modules Included with CORVUS Hard Disk System

In addition to the modules included with the floppy disk version of P&T CP/M 2, the following modules are included in the module library that comes with the Corvus hard disk version. You may select from among them to customize the system to your needs. See Chapters 6 and 7 of the P&T CP/M 2 User's Manual for more information about selecting modules.

The descriptions of the hard disk parameter modules given here are brief and intended to help you decide which ones merit further investigation. If you want more complete information about the disk configuration, the MODSEL program will allow you to view the complete text description associated with each module.

CORV1
Contains the I/O routines which enable the system to access a revision (series) A Corvus hard disk. If you have a revision B or H Corvus system you should use CORV2. You must select either CORV1 or CORV2 for inclusion in the system if you want to use the hard disk.

CORV2
Contains the I/O routines which enable the system to access a revision (series) B or H Corvus hard disk. If you have a revision A Corvus system you should use CORV1. You must select either CORV1 or CORV2 for inclusion in the system if you want to use the hard disk.

HDTB10
Contains a disk parameter table for use with a single 10 Mbyte Rev. B hard disk and two floppy drives. Logical drives A and B are defined on the hard disk. Drive A is about 2.3 Mb in size with 256 directory entries and 4 Kb allocation blocks. Logical drive B is 8 Mb in size with 512 directory entries and 8 Kb allocation blocks. Logical drives C and D are defined on physical floppy drives 0 and 1, respectively.

HDTB20-1
Contains a disk parameter table for use with a single 20 Mbyte Rev. B hard disk and two floppy drives. Logical drives A, B and C are defined on the hard disk. Drive A is about 2.7 Mb in size with 256 directory entries and 4 Kb allocation blocks. Logical drives B and C are 8 Mb in size with 512 directory entries and 8 Kb allocation blocks. Logical drives D and E are defined on physical floppy drives 0 and 1, respectively.

HDTB20-2
Contains a disk parameter table for use with a single 20 Mbyte Rev. B hard disk and two floppy drives. Logical drives A, B, C, and D are defined on the hard disk and divide it into four equal parts each with 4.7 Mb, 256 directory entries, and 8 Kb allocation blocks. Logical drives E and F are defined on physical floppy drives 0 and 1, respectively.

HDTH6
Contains a disk parameter table for use with a single 6 Mbyte Rev. H hard disk and two floppy drives. Logical drives A and B are defined on the hard disk and divide it into two equal parts each with 2.8 Mb, 256 directory entries, and 4 Kb allocation

blocks. Logical drives C and D are defined on physical floppy drives 0 and 1, respectively.

HDTH10
Contains a disk parameter table for use with a single 10 Mbyte Rev. H hard disk and two floppy drives. Logical drives A and B are defined on the hard disk. Drive A is about 3.5 Mb in size with 256 directory entries and 4 Kb allocation blocks. Logical drive B is 8 Mb in size with 512 directory entries and 8 Kb allocation blocks. Logical drives C and D are defined on physical floppy drives 0 and 1, respectively.

HDTH20-1
Contains a disk parameter table for use with a single 20 Mbyte Rev. H hard disk and two floppy drives. Logical drives A, B and C are defined on the hard disk. Drive A is 2 Mb in size with 256 directory entries and 4 Kb allocation blocks. Logical drives B and C are about 7.8 Mb in size with 512 directory entries and 8 Kb allocation blocks. Logical drives D and E are defined on physical floppy drives 0 and 1, respectively.

HDTH20-2
Contains a disk parameter table for use with a single 20 Mbyte Rev. H hard disk and two floppy drives. Logical drives A, B, C, and D are defined on the hard disk and divide it into four equal parts each with 4.4 Mb, 256 directory entries, and 8 Kb allocation blocks. Logical drives E and F are defined on physical floppy drives 0 and 1, respectively.

HDTH20-3
Contains a disk parameter table for use with a single 20 Mbyte Rev. H hard disk and two floppy drives. Logical drives A, B, and C are defined on the hard disk and divide it into three equal parts each with 5.8 Mb and 8 Kb allocation blocks. Drive A has 512 directory entries and drives B and C each have 256 directory entries. Logical drives D and E are defined on physical floppy drives 0 and 1, respectively.

Index

PICKLES & TROUT®

P.O. BOX 1206, GOLETA, CA 93116