7. DEVICE RE - ROUTING

ABOUT THIS CHAPTER

This chapter describes how input and output can be re-routed from/to alternative devices or files.

For further details of the commands mentioned in this chapter refer to Chapter 13.

CONTENTS

INTRODUCTION

Normal operation of the M20 system involves entering information via the keyboard and receiving responses on the VDU. In some cases, however, you may wish to use additional or alternative input and/or output devices; for example a printer can be used to get a hard copy of what is displayed on the video. This can be done on the M20 using "device re-routing parameters".

Device re-routing can be specified on two levels:

- local (for the duration of a single command only)
- global (for all subsequent PCOS commands entered during the current working session, or until respecified by another global device re-routing operation)

The input device may be re-specified as:

- the keyboard (standard)
- a hard disk or diskette file
- an RS-232-C port

The output device may be specified as:

- the VDU (standard)
- a disk or diskette file
- a printer
- an RS-232-C port

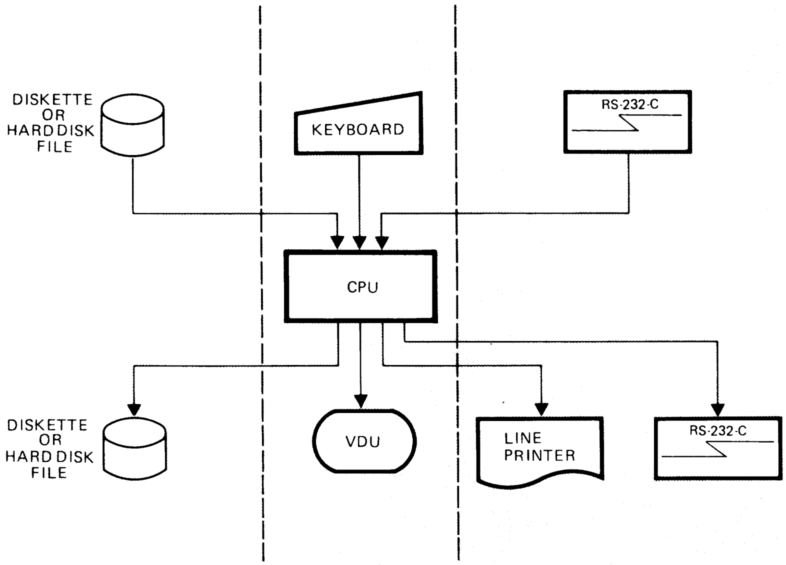


Fig. 7-1 Input/Output Devices

LOCAL DEVICE RE-ROUTING

This facility enables you to change the input and output devices for the current PCOS command, such that input can be received from devices other than the keyboard, and output can be routed to devices other than the VDU. After the command has been executed the re-routing specified is cancelled.

The implementation takes the form of a parameter to the command. This parameter specifies the name of the device whose I/O status is to be changed, preceded by two indicators. The first of these indicators specifies whether the device is to be cancelled as an input/output device (indicated by a "-" sign), or enabled as an input/output device (indicated by a "+" sign). The second indicator specifies whether the device is to supply the input (indicated by "S" for source), or receive the output (indicated by "D" for destination).

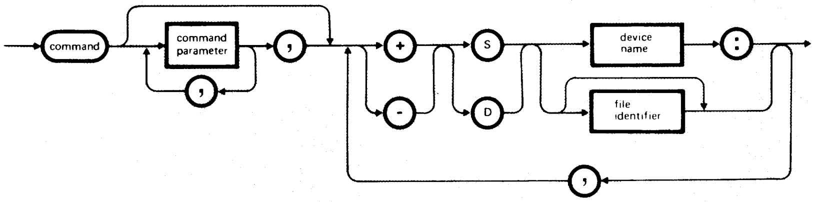


Fig. 7-2 Local Device Re-Routing

Where

SYNTAX ELEMENT	MEANING
command	a keyword of any PCOS command to be executed via the devices specified in the device re-routing parameters
command parameter	a parameter to be passed to the PCOS command in question
+	the device or file specified is to be enabled
-	the device or file specified is to be cancelled
S	the specified device is to be a source (for input to the CPU). It can be entered in upper or lower case

D	the specified device is to be a destination (for output from the CPU). It can be entered in upper or lower case
device name	a string of up to 13 printable ASCII characters, the first of which must not be a digit, denoting the device in question. This can be the device default name or any other name assigned to the device using the SDEVICE command. Note: A device name must be followed by a colon
file identifier	any file identifier. If the file does not exist it will be created on either the specified volume or the volume inserted in the last selected drive. Only one source and one destination file are permitted to be open at one time

Note

- + or -, S or D and device name or file name follow each other without any spaces in between, and constitute one parameter

- Each device re-routing parameter must be followed by a comma to separate it from the next parameter

- The position of the device re-routing parameter in the comand line is arbitrary. It is shown here at the end of the line merely for clarity

- +DPRT: may also be written as +PRT (in upper or lower case letters) see last example below

- If additional devices are enabled for input or output without disabling the currently active device(s) then devices become simultaneously active. The user must therefore exercise caution when re-routing input to avoid data from several devices becoming intermixed

Examples

IF you enter...	THEN...
vl 0:,-dcons:,+DPRT: /CR/	the directory of the diskette inserted in drive 0 is printed (because of +DPRT:). However, it is not displayed on the VDU (because of -dcons:)
ba +SBASIC.CMD /CR/	the M20 goes into the BASIC environment (see the BASIC command) and will take input from both the BASIC.CMD file and the keyboard. The CPU will no longer take input from the BASIC.CMD file when it goes back into the PCOS environment
ss +PRT /CR/	the global system environment parameters are not only displayed on the VDU, but also printed

GLOBAL DEVICE RE-ROUTING

Specification of global device re-routing causes the input and output of all PCOS commands executed thereafter to be re-routed. Any re-routing remains in operation until otherwise specified or until the system is reset. Furthermore, global device re-routing parameters can be made a permanent feature of the operating system by use of the PSAVE command.

Global device re-routing can be put through if device re-routing parameters, as described in the syntax diagram for local device re-routing (see Figure 7-2), are specified by themselves; that is, in the absence of a command keyword.

Note:

- + or -, S or D, and device name follow each other without any spaces in between
- Device re-routing parameters must be separated from each other by a comma
- +DPRT: may also be written as +PRT (in upper or lower case letters)
- If additional devices are enabled for input or output without disabling the currently active device(s) then devices become simultaneously active. The user must therefore exercise caution when rerouting input to avoid data from several devices becoming intermixed
- No more than one file for output and one file for input may be specified at any one time
- The keyboard can be disabled by specifying "-SCONS:". But, if this is done at the global level, control cannot be regained unless an external active device issues a "+SCONS:", or the system is reset

Examples

IF you enter...	THEN...
+scom:,+dcom: /CR/	the CPU will receive input from both the keyboard and the built-in RS-232-C communications port (assuming the latter has already been initialised). Output will be displayed on the VDU and re-routed to the RS-232-C communications port
-dcom:,+dprt: /CR/	the RS-232-C communications port (previously enabled) will be cancelled as a destination (for output) and the printer activated. Any other devices previously allocated as a source or a destination will remain active

+d1:fileA /CR/	"fileA" resident on the diskette inserted in drive 1 is enabled for ouput. If "fileA" does not exist on the volume it is created, and all the output will be both displayed on the screen and re-routed to "fileA"
+dprt: /CR/ +d1:output /CR/ -dprt: /CR/ -d /CR/	the first command enables the printer for output. The second enables the file named "output" resident on the diskette inserted in drive 1 to receive output. The third command cancels the printer and the fourth cancels the file. Note: When a file is cancelled the identifier is not checked and can therefore be omitted

DEVICE RE-ROUTING FROM A BASIC PROGRAM

When local device re-routing is specified while in BASIC, I/O redirection will continue until either a command is executed to explicitly turn it off, or BASIC is exited. Conversely, global device re-routing will remain active after BASIC is exited.

Examples

IF you enter...	THEN...
ba /CR/ EXEC "vl 1:,+D1:OUT" /CR/ . . . EXEC "-D1:OUT" /CR/	the first command enters the BASIC environment. The first EXEC statement performs a VLIST command on the diskette inserted in drive 1 and routes the volume list to the file named "OUT" on the same diskette as well as to the VDU. All subsequent output will also be routed to the file until it is cancelled by the second EXEC statement

ba /CR/ EXEC "vl 1:,+dprt:" /CR/ . . . SYSTEM /CR/	the system enters the BASIC environment. The EXEC statement causes the directory of the diskette inserted in drive 1 to be printed. All subsequent output is also directed to the printer until the printer is cancelled as an output device by the SYSTEM command
ba /CR/ EXEC "+prt" /CR/ . . . SYSTEM /CR/	the system enters the BASIC environment. The EXEC statement specifies the printer as an output device, but because "+prt" is specified globally, the following SYSTEM command will not cancel it

8. PROTECTION TOOLS

ABOUT THIS CHAPTER

This chapter describes the mechanisms by which a file or volume can have protection applied. For details about the commands mentioned in this chapter refer to Chapter 13.

CONTENTS

INTRODUCTION

The M20 offers password protection at both volume and file level. Write-protection may be applied to a diskette or a file, but not to the hard disk. Moreover, a BASIC program can be protected so that it cannot be listed, edited or saved again.

The following sections summarise the various protection mechanisms.

VOLUME PASSWORDS

IF you want to...	THEN...

assign a password to a volume	issue a VPASS command, specifying the password. For example
	
	`vp MYVOL:,MYPASS /CR/`
	
	OR
	
	if the volume already has a password this must be specified by the VPASS command, which, in this case, will change the password. For example
	
	`vp VOL1/OLDPASS:,NEWPASS /CR/`

access a volume that has a password (or a file saved on a volume that has a password)	enable that volume by specifying the volume password after the volume name or the drive number. This can be done in a BASIC or PCOS command or in an OPEN statement. Note: Once a diskette volume password has been specified it need not be re-specified until the diskette has been removed and another diskette has been referenced in the drive in which the diskette was inserted, or the current working session is terminated. Moreover, once a hard disk volume password has been specified, the hard disk will remain enabled until the end of the current working session

remove a volume password	issue a VDEPASS command, for example `vd MYVOL/MYPASS: /CR/` Note: You must know the password to use the VDEPASS command
hide a volume password	press /CTRL/ /G/. The cursor will change its shape and blink rate and the display of entered characters is suppressed (Hide mode). To return to normal display mode you must press /CTRL/ /G/ again, or /CR/

FILE PASSWORDS

IF you want to...	THEN...
assign a password to an existing file (that has no password)	issue an FPASS command, specifying the password. For example `fp V1:MYFILE,PASS001 /CR/`
create a new file and assign a password to it	issue an FNEW command, specifying the password. For example `fn 1:newfile/pass002,4 /CR/`
change the password to an existing file	issue an FPASS command, specifying the old password within the file identifier, and the new password as the second parameter. For example `fp 1:newfile/pass002,pass102 /CR/`

assign a password to a group of files	issue an FPASS command, specifying the group using wild card characters. The same password will be assigned to all files in the group. For example `fp 1:my*,mine  /CR/`

assign a password to a list of files	issue an FPASS command, specifying a list of files and, as the last parameter, the common password. For example
	
	`fp 1:myfile,yourfile,hisfile,`
	`herfile,ours  /CR/`

assign a password to a program file (that has none)	an FPASS command can be issued or, if in BASIC, the program can be stored on a volume using the SAVE command specifying the password. For example `SAVE "FILEABC/PASSABC"  /CR/`
access a file that has a password	specify that password after the file name. For example `fw FILE002/PASS002 /CR/`
remove a file password	issue an FDEPASS command, specifying the password. For example `fd V1:MYFILE/PASS001 /CR/`
hide a file password	press /CTRL/ /G/ simultaneously. The cursor changes its shape and blink rate and and the display of entered characters is suppressed (Hide mode). To return to normal display mode you must press /CTRL/ /G/ again, or /CR/

WRITE-PROTECTION

You can apply write-protection to a diskette or a file.

IF you want to...	THEN...
write-protect a diskette (that is, to prevent any writing to that diskette)	cover the write-protect notch with an aluminised label
remove write-protection from a diskette	remove the aluminised label
write-protect a file	issue an FWPROT command, specifying the file identifier. For example `fw 1:myfile /CR/`
remove write-protection from a file	issue a FUNPROT command, specifying the file identifier. For example `fu 1:myfile /CR/`

COPY-PROTECTION

Copy-protection can only be assigned by the supplier. It can be assigned at file or volume level. A copy-protected file can be copied only a specified number of times. A copy-protected volume, however, cannot be copied at all.

BASIC PROGRAM SECURITY

Besides password protection and write-protection the M20 offers a further level of security. In the BASIC environment a program can be saved using the SAVE command with the P option. In such a case, the saved program can no longer be:

- listed
- modified
- saved again

For example

```
SAVE "1:FILEAPROT", P /CR/
```

saves the file program FILEAPROT with the P option on the diskette inserted in drive 1.

9. VOLUME HANDLING

ABOUT THIS CHAPTER

This chapter provides an operational guide to the use of the volume directed commands.

Throughout this chapter the availability of commands is always assumed. That is, it is assumed that either a volume containing the command is present in one of the drives, or that the command in question is already resident in memory.

Additional information about the commands mentioned in this chapter can be found in Chapter 13.

CONTENTS

FORMATTING AND INITIALISING NEW VOLUMES

The operations of formatting a volume and initialising a volume are carried out by the VFORMAT and VNEW commands, respectively.

Formatting creates blank sectors on the tracks of a diskette or hard disk and checks for defective blocks on the hard disk only). A new diskette or hard disk must be formatted before use but note that Olivetti-supplied diskettes for the M20 are pre-formatted to a high precision and therefore do not require formatting. In fact to ensure reliability and to ensure that the same diskette can be used on any M20, you are advised not to re-format. Pre-formatted diskettes, however, are not initialised, and therefore require the VNEW command before they can be used.

Diskettes that are not supplied by Olivetti, however, will need to be formatted using the VFORMAT command. Once formatted in this manner, a diskette is initialised, ready for use.

Used M20 diskettes, whether Olivetti-supplied or not, should not be reformated. It is sufficient to initialise them using the VNEW command. The same is true for hard disks, except that after performing a VVERIFY destructive test it is necessary to re-format.

FORMATTING A DISKETTE OR A HARD DISK

Formatting New Diskettes

To format a new diskette you must place the unformatted diskette (which must not be write-protected) in an available drive (for instance drive 1), then enter

```
        vf 1: /CR/

        OR

        vf %s 1: /CR/  - for the special case of a 160 Kbyte diskette on
                         a 320 Kbyte drive
```

In either case the following message will be displayed:

```
        Warning - VFormat deletes all files.  Format disk? (y/n)
```

Respond by entering

```
        y /CR/
```

(or n /CR/ to abort)

and formatting begins. The message

```
        Formatting Track 0
```

appears on the screen, and the track number is subsequently incremented as each track is completed until all tracks are formatted. At this point the message

```
        Formatting Complete
```

is displayed and the PCOS prompt appears. The diskette is then formatted and ready for use.

Formatting Used Diskettes

Formatting will also work on used diskettes, although you are recommended to use the VNEW command. However, supposing you have a used diskette bearing the name "mydisk", you can reformat it by entering

```
        vf mydisk:  /CR/


        OR

        vf %s mydisk:   /CR/   - for  the  special  case  of  a 160Kbyte
                                 diskette on a 320 Kbyte drive
```

The M20 will respond with the message

```
        Warning - VFormat deletes all files. Format disk? (y/n)
```

Respond by entering

```
        y  /CR/
```

If the diskette is not enabled, that is, it is password protected and has not yet been accessed during the current working session, then the message

```
        Diskette appears password protected. Format disk? (y/n)
```

is displayed. Respond by entering

```
        y /CR/
```

and formatting procedes as for a new diskette. Once formatting is complete the diskette will be unnamed and have no password. The diskette is then ready for use.

Formatting Hard Disks

Hard disks are formatted in exactly the same way as diskettes except that the drive number is always 10. That is, you should enter

```
vf 10: /CR/
```

then the procedure continues as for a diskette.

No Interaction

The dialogue for any of the above formatting procedures can be skipped by including the no interaction flag (%n) in the command line. For example, if you enter

```
vf %n mydisk:  /CR/
```

then the diskette named "mydisk" is formatted. Formatting is complete when the PCOS prompt is displayed. The diskette is then ready for use.

INITIALISING A VOLUME

To initialise a volume you must enter the VNEW command along with the volume identifier and, if the disk or diskette is not enabled, the volume password. For example

```
vn mydisk/pass:  /CR/   - for a diskette having the name "mydisk"
                          and password "pass"
OR

vn %s mydisk:  /CR/     - for a diskette having the name "mydisk"
                          and that is a 160 Kbyte diskette on a
                          320 Kbyte drive
OR

vn 10:  /CR/            - for a hard disk
```

The M20 will respond with the message

```
Warning - vnew deletes all files. Initialise disk? (y/n)
```

Respond by entering

```
y /CR/
```

The initialisation process then begins. Its completion is indicated by the PCOS prompt.

The initialisation process removes any volume name.

Note that the above dialogue can be avoided by use of the no-interaction flag (%n). For example

```
        vn %n mydisk/pass:  /CR/
```

will perform the same function but without dialogue.

If you have forgotten the password of the diskette you will not be able to use the VNEW command.

LISTING A VOLUME

There are two commands available for listing the files contained in a volume. These are VLIST and VQUICK.

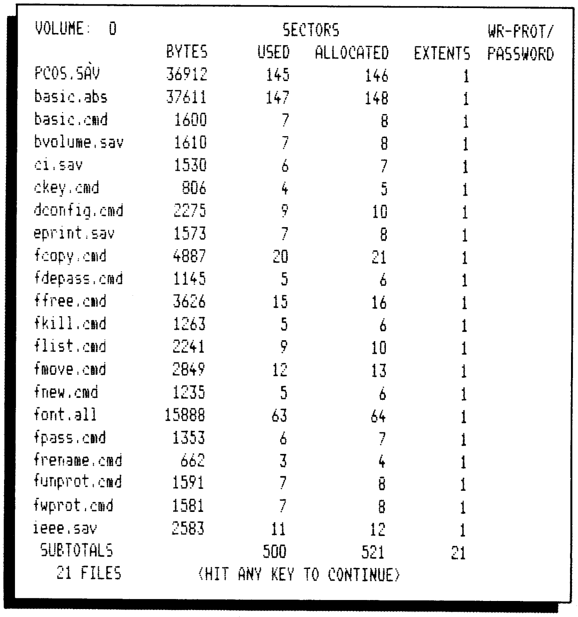
LISTING A VOLUME USING THE VLIST COMMAND

If you enter the VLIST command along with an appropriate volume identifier, a list of the files on the volume is generated. Displayed with each file name is information about that file: that is, the number of bytes occupied by the file, the number of sectors used, the number of sectors allocated, the number of extents used, and whether or not the file is write-protected or password protected.

For example, to examine the contents of the system diskette you must insert it in an available drive (for instance drive 1) then enter

```
        vl 1:  /CR/
```

A display such as the one shown in Figure 9-1 will result.



```
VOLUME:  0                          SECTORS                 WR-PROT/
                 BYTES      USED   ALLOCATED    EXTENTS  PASSWORD
PCOS.SAV         36912       145         146          1
basic.abs        37611       147         148          1
basic.cmd         1600         7           8          1
bvolume.sav       1610         7           8          1
ci.sav            1530         6           7          1
ckey.cmd           806         4           5          1
dconfig.cmd       2275         9          10          1
eprint.sav        1573         7           8          1
fcopy.cmd         4887        20          21          1
fdepass.cmd       1145         5           6          1
ffree.cmd         3626        15          16          1
fkill.cmd         1263         5           6          1
flist.cmd         2241         9          10          1
fmove.cmd         2849        12          13          1
fnew.cmd          1235         5           6          1
font.all         15888        63          64          1
fpass.cmd         1353         6           7          1
frename.cmd        662         3           4          1
funprot.cmd       1591         7           8          1
fwprot.cmd        1581         7           8          1
ieee.sav          2583        11          12          1
 SUBTOTALS                   500         521         21
   21 FILES          (HIT ANY KEY TO CONTINUE)
```

Fig. 9-1 Typical Volume List of the System Diskette

By repeatedly striking a key you will step through the list one screen at a time until you reach the end of the list. At this point the M20 will display some totals, after which the PCOS prompt will re-appear.

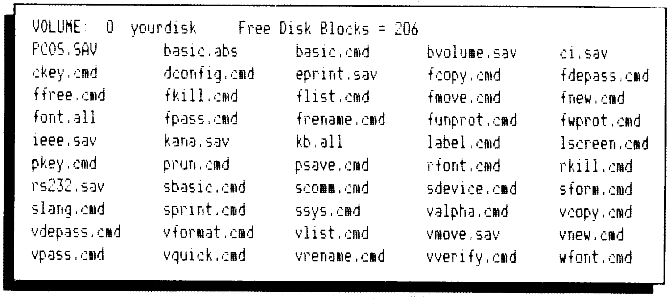
Use of the no-interaction (%n) flag with this command causes the list to scroll continuously until the end of the list.

LISTING A VOLUME USING THE VQUICK COMMAND

Entering the VQUICK command along with a volume identifier also generates a list of the files on the volume but without any additional information about the files. For example, if you want to list the files contained on the system diskette using the VQUICK command, first insert the diskette in one of the drives (for instance drive 0) then enter

```
vq 0:  /CR/
```

A display such as that shown in Figure 9-2 will result.



```
VOLUME  0  yourdisk    Free Disk Blocks = 206
PCOS.SAV        basic.abs       basic.cmd       bvolume.sav     ci.sav
ckey.cmd        dconfig.cmd     eprint.sav      fcopy.cmd       fdepass.cmd
ffree.cmd       fkill.cmd       flist.cmd       fmove.cmd       fnew.cmd
font.all        fpass.cmd       frename.cmd     funprot.cmd     fwprot.cmd
ieee.sav        kana.sav        kb.all          label.cmd       lscreen.cmd
pkey.cmd        prun.cmd        psave.cmd       rfont.cmd       rkill.cmd
rs232.sav       sbasic.cmd      scomm.cmd       sdevice.cmd     sform.cmd
slang.cmd       sprint.cmd      ssys.cmd        valpha.cmd      vcopy.cmd
vdepass.cmd     vformat.cmd     vlist.cmd       vmove.sav       vnew.cmd
vpass.cmd       vquick.cmd      vrename.cmd     vverify.cmd     wfont.cmd
```

Fig. 9-2 Typical Volume Quick list of the system disk

COPYING VOLUMES

There are two commands available for copying diskettes. These are:

VCOPY - for copying a diskette from one drive to another. That is, for copying diskettes on a dual-drive system.

VMOVE - for copying diskettes on a system with only one diskette drive

Note that by using these commands it is not possible to copy a volume of a given capacity onto a volume of a different capacity. Any attempt to do so will cause an error message to be displayed: that is, you can only copy a 160 Kbyte diskette onto another 160 Kbyte diskette, a 320 Kbyte diskette onto another 320 kbyte diskette, or a 640 Kbyte diskette onto another 640 Kbyte diskette. Copy operations between different volume sizes can only be performed using the FCOPY command (or FMOVE command on a single drive system).

CAUTION: IT IS IMPORTANT THAT YOU WRITE-PROTECT YOUR SOURCE DISKETTE BEFORE COPYING IT TO AVOID ACCIDENTALLY OVERWRITING IT.

COPYING VOLUMES ON A DUAL-DRIVE SYSTEM (VOLUMES OF EQUAL SIZE)

Before copying a diskette you must first load the VCOPY command into memory using the PLOAD command (if you are copying the system diskette this is not necessary). That is, insert the system diskette into one of the drives then enter

```
        pl vc  /CR/
```

Remove the system diskette and insert the write-protected source diskette in one of the drives (for instance drive 0) and the target volume (which must not be write-protected) in the other drive. Then enter

```
        vc 0:,1:  /CR/
```

and the M20 will respond with

```
        Warning- vcopy deletes all files. Copy disk? (y/n)
```

Respond by entering

```
        y  /CR/
```

A message such as the following will then be displayed

```
        Read block 0 to 144
```

which indicates that the M20 is reading blocks 0 to 144 from the source volume. After a time this message changes to

```
        Write block 0 to 144
```

which indicates that blocks 0 to 144 are being written to the target diskette. (The number of blocks read and written at once depends on the size of user memory.) When this operation is complete another "Read block" message appears for the next group of blocks. This process continues until all blocks have been copied to the target volume. At this point the message changes to

```
        VCopy complete
```

and the PCOS prompt appears.

Note that in the command line the volume identifiers may alternatively be specified by name, in which case the target volume assumes the same name as the source volume once the copy operation is complete.

If the source volume is password protected then the password need not be specified, but it will also be copied to the target volume. If the target volume is password protected then its password need not be specified either.

COPYING DISKETTES USING ONE DRIVE (VOLUMES OF EQUAL SIZE)

If you intend to copy a diskette on a single-drive system you must first insert the system diskette then make the VMOVE command memory resident via the PLOAD command (if you are copying the system diskette this is not necessary). That is, you must enter

```
        pl vm  /CR/
```

Now remove the system diskette.

The above step is unnecessary on a hard disk system, provided the VMOVE command resides on the hard disk.

Now insert the write-protected source diskette into the drive and enter the VMOVE command. If neither the source volume nor the target volume is password protected it is only necessary to specify the command name. Simply enter

```
        vm   /CR/
```

If the source volume is password protected then the full source volume identifier - including the password - must be specified. For example, if your source volume is named "mydisk" and has the password "pass" then you must enter

```
        vm mydisk/pass:  /CR/
```

Furthermore, if the target volume is password protected then both the source and target volumes must be specified in full. For example, if your source volume is named "mydisk" and has the password "pass", and the target volume is named "yourdisk" and has the password "pass1", then it is necessary to enter

```
        vm mydisk/pass:,yourdisk/pass1:  /CR/
```

Once you have entered the command line the M20 responds with the following message

```
        Warning- vmove deletes all files and PCOS.  Vmove disk? (y/n)
```

This means that not only does the VMOVE command overwrite everything in the target volume, but in doing so it also overwrites the M20's memory thereby deleting the operating system.

Respond by entering

```
        y  /CR/
```

At this point the M20 will fill all its memory space with data from the source diskette. When memory is full, a message will be generated asking you to insert the target diskette and hit any key. The M20 will then transfer all the data from memory onto the target diskette.

This process will be repeated a number of times (depending on the diskette capacity). At the end of the copy operation you can either make further copies or re-boot PCOS.

COPYING VOLUMES (OF DIFFERENT SIZES)

Where the source and destination volumes are of a different size the copy operation can only be performed using a file copy command. If the target volume is to contain only the contents of the source volume it must be blank, and formatted. For example, to copy the entire contents of a 160 Kbyte diskette inserted in drive 0 onto a 320 Kbyte diskette inserted in drive 1 enter

```
     fc 0:*,1  /CR/
```

The wild card character '*' specifies all the files on volume 0.

You must take care, however, when copying from a larger volume to a smaller volume that there is enough capacity on the diskette to accommodate all files.

This operation can be performed on a single drive system using the FMOVE command but note that wild card characters may not be used; each file must specified in a separate FMOVE command.

For further details on the FCOPY and FMOVE commands refer to Chapter 10.

NAMING AND PROTECTING A VOLUME

Volumes can be write-protected and/or password protected. Furthermore, it is often convenient to name a volume such that it can subsequently be identified by name rather than the number of the drive in which it resides.

WRITE-PROTECTION

Diskettes can be write-protected by fixing an aluminised label over the notch in the side of the diskette. Write-protection can be removed by simply removing the aluminised label. Hard disks cannot be write-protected.

PASSWORD PROTECTION

Password protection can be applied to a volume by means of the VPASS command. If you enter

```
        vp 1:,pword   /CR/
```

then the diskette inserted in drive 1 will be assigned the password "pword".

Similarly the (optional) hard disk can be password protected by specifying either the volume name or the drive number. In the latter case the drive number is always 10. For example, if you enter

```
        vp 10:,dpass
```

then the hard disk is assigned password "dpass".

Password protection has no effect on some commands, such as VCOPY, VLIST and VQUICK, but most operations require the diskette to be enabled.

Enabling a diskette implies specifying the diskette password as part of its volume identifier as a parameter to a command. For example, if you terminate the current working session then re-boot the system and insert into drive 0 the diskette that you previously password protected, then enter

```
        vl 0/pword:   /CR/
```

then the files contained on that volume are listed and the volume is enabled. The volume then remains enabled until either the volume is removed from its drive and another one inserted, or the current working session is terminated.

To remove password protection from a diskette use the VDEPASS command. For example, if you enter

```
        vd 0/pword:   /CR/
```

then password "pword" is removed from the diskette. But note that you must know the password to be able to delete it.

Do not assign a password to your system diskette since doing so prevents you from accessing any command on that diskette. Moreover, the VDEPASS command will also be disabled and hence you will not be able to enable the volume except from a back-up.

NAMING A VOLUME

There are three commands that enable you to name a volume. These are VFORMAT, VNEW and VRENAME.

The VFORMAT and VNEW commands enable you to name a volume at the same time as formatting or initialising by specifying the volume name as the second parameter to the command. For example, if you enter

```
vf 1:,datadisk  /CR/
```

then the new diskette inserted in drive 1 will be formatted and assigned the name "datadisk".

Alternatively, a used diskette (or a hard disk) can be re-initialised and assigned a name by use of the VNEW command. For example, if you enter

```
vn 1:,newname  /CR/
```

then the volume inserted in drive 1 is re-initialised and assigned the name "newname".

The third method of naming a diskette (or hard disk) simply assigns a name to an existing volume. This is done using the VRENAME command. If you enter

```
vr newname:,oldname  /CR/
```

then the volume you previously named "newname" will have its name changed to "oldname". Note that it is not necessary to specify an old volume name in this command, it is also valid to specify the diskette (or hard disk) by the drive number. For example, if the last operation was performed with the diskette in drive 1 it would have been equally effective to enter

```
vr 1:,oldname  /CR/
```

Note that if a volume is not enabled then its password must be specified if its name is to be changed using the VRENAME command.

ALPHABETISING A VOLUME

PCOS contains a facility that enables you to sort the files contained in a volume into alphabetical order. Any unused directory blocks on the volume are removed, thereby improving access time whenever the directory is scanned. To do this you simply enter the VALPHA command along with the volume identifier. For example, if the volume named "datadisk" resides in one of the drives you can alphabetise it by entering

```
va datadisk: /CR/
```

Note that if the volume is not enabled then you must also specify the password. Furthermore, the volume must not be write-protected.

10. FILE HANDLING

ABOUT THIS CHAPTER

This chapter provides an operational guide for the use of the file directed commands.

Throughout the chapter the availability of commands is always assumed. It is assumed that either a volume containing the command is inserted in one of the drives, or that the command in question is already resident in memory.

For a detailed description of the commands mentioned in this chapter, refer to Chapter 13.

CONTENTS

RENAMING FILES 10-15

CREATING FILES

File creation within the PCOS environment can be performed either by the FNEW command which creates empty files, or simply by copying existing files using the FCOPY or FMOVE command. Text files can be created using the Video File Editor (see Chapter 12). BASIC files can be created via the Video File Editor and also from the BASIC environment (see "BASIC Language Reference Guide").

CREATING AN EMPTY FILE

Creating an empty file requires the use of the FNEW command. Before this command can be executed the volume on which the file is to be created must be enabled and must not be write-protected.

To create an empty file simply enter the FNEW command along with the file identifier and the file size (in terms of the number of blocks to be made available to the file). For example

```
        fn 1:myfile,6 /CR/
```

causes an empty file named "myfile" to be created on the diskette inserted in drive 1, and 6 blocks to be allocated to it.

If the file size parameter is specified as zero, or not specified at all, then the file system allocates the number of blocks specified in the "extent size" global parameter. For example, if you enter

```
        ss ,,,4 /CR/

        .
        .
        .

        fn 1:newfile,0 /CR/
```

then the "extent size" parameter is set to 4 by the SSYS command, and an empty file named "newfile" is created on the diskette inserted in drive 1 and consequently allocated 4 blocks.

Note that the minimum number of blocks that can be allocated is 2, and that if you specify the file size as 1 then the file system automatically allocates 2.

If there is insufficient space on the diskette to accommodate the new file then a message

```
        ERROR 61 ----- disk filled
```

is displayed.

CREATING A FILE BY COPYING

Files are created when using the file copy commands if the target file does not already exist. For example, if you enter

```
        fc 0:myfile,1: /CR/
```

on a dual drive system then the file named "myfile" on the diskette inserted in drive 0 will be copied onto the diskette inserted in drive 1. That is, the file named "myfile" will be created there, unless a file of that name already exists, in which case it will be overwritten. Furthermore, if you enter

```
        fc 0:myfile,yourfile /CR/
```

then file "yourfile" is created (if it does not already exist) and the contents of "myfile" are copied into it.

On single drive systems the FMOVE command must be used. In this case only one file can be specified in the command line. Assuming the source diskette is already inserted in the drive the file to be copied is written into user memory. You then have to remove the source diskette, insert the target diskette and copy the contents of user memory onto the target diskette, thereby creating a file of the same name. If, however, the file is too large to fit into user memory then the operation requires a number of passes.

Further operational details are provided in the section "Copying Files".

COPYING FILES

The PCOS command library contains two commands for copying files. These are:

FCOPY - for copying files on a dual drive system

FMOVE - for copying files from one diskette to another using a single drive system

COPYING FILES ON A DUAL DRIVE SYSTEM

First of all the volume containing the file(s) to be copied must be inserted in one of the drives. To copy a file you then simply need to enter the FCOPY command specifying the file to be copied (the source file) and the destination (the target volume or target file).

Copying One File At a Time

To copy "myfile" from a volume named "myvol" inserted in drive 1 to a volume named "copyvol" inserted in drive 0 enter

```
        fc myvol:myfile,copyvol: /CR/
```

and, provided "myfile" does not already exist on "copyvol" the M20 will respond

```
        COPY FILE 1:myfile TO 0:myfile
```

then the file is copied and the new file is given the same name as the source file.

If the source file has a password then this is also assigned to the new file. For example, to copy a file named "datafile" and having password "dpass" from the diskette inserted in drive 0 to the hard disk you would enter

```
        fc 0:datafile/dpass,10:
```

then, assuming a file named "datafile" does not already exist on the hard disk, the M20 would display

```
        COPY FILE 0:datafile TO 10:datafile
```

after which a copy of "datafile" with password "dpass" would be created on the hard disk.

The FCOPY command can also copy a file to an existing file thereby overwriting the target file. For example, to copy a file named "ifile1" having password "ipass" from volume "myvol" inserted in drive 0 to file "ifile2" with password "ind" on volume "yourvol", enter

```
        fc myvol:ifile1/ipass,yourvol:ifile2/ind  /CR/
```

and the M20 will respond

```
        COPY FILE 0:ifile1 TO 1:ifile2
        File already exists. Do you wish to overwrite (y or n)?
```

If you then respond by entering

```
         y  /CR/
```

then the target file will end up keeping the file name "ifile2" and password "ind", but its contents will be overwritten with those of "ifile1". The PCOS prompt appears when the copy operation is complete.

It is also possible to copy a file to another file within the same volume. For example, if you enter

```
        fc myvol:ifile1/ipass,copyfile/cpass  /CR/
```

then, after replying "y" to the subsequent prompt, the contents of "copyfile" are overwritten by those of "ifile1". However, "copyfile" still maintains its name and password.

Note: The dialogue in any of the above examples can be bypassed by using the no-interaction (%n) flag. For example

```
        fc %n myvol:newfile,copyvol: /CR/
```

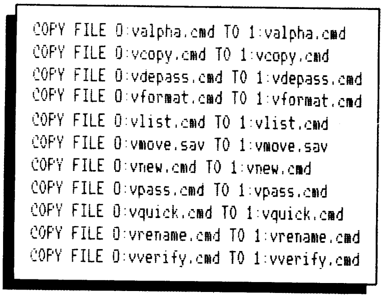
performs the copy operation without any intervening dialogue.

Copying Groups of Files using Wild Cards

A group of related files can be copied from one volume to another using the wild card facility. To do this you must enter the FCOPY command along with the group of files specified using wild cards, and the target volume. For example, you can copy all the volume-directed commands from the system diskette to another volume using one command by inserting the system diskette in one of the drives (for instance, drive 0), having the target volume inserted in the other and entering

```
        fc 0:v*,1:  /CR/
```

The M20 will then respond with the display shown in Figure 10-1.



```
COPY FILE 0:valpha.cmd TO 1:valpha.cmd
COPY FILE 0:vcopy.cmd TO 1:vcopy.cmd
COPY FILE 0:vdepass.cmd TO 1:vdepass.cmd
COPY FILE 0:vformat.cmd TO 1:vformat.cmd
COPY FILE 0:vlist.cmd TO 1:vlist.cmd
COPY FILE 0:vmove.sav TO 1:vmove.sav
COPY FILE 0:vnew.cmd TO 1:vnew.cmd
COPY FILE 0:vpass.cmd TO 1:vpass.cmd
COPY FILE 0:vquick.cmd TO 1:vquick.cmd
COPY FILE 0:vrename.cmd TO 1:vrename.cmd
COPY FILE 0:vverify.cmd TO 1:vverify.cmd
```

Fig. 10-1 Sample Display of a Copy Operation on a Group of Files

The copy operation is then complete. In this case none of the files already existed on the target volume. For each file that had existed on the target volume the prompt

```
        File already exists. Do you wish to overwrite (y or n)?
```

would be displayed.

Password protected files will not be copied by this procedure, unless the password is specified in the command line and is common to all files in the group.

Note: The dialogue in all the above examples can be bypassed by entering the no-interaction (%n) flag in the command line. In this case the system simply returns the PCOS prompt when the copy operation is complete.

COPYING FILES USING ONE DRIVE

Copying Files Diskette to Diskette

To copy files from one diskette to another on a single drive or hard disk system you must enter the FMOVE command along with the file name. For example, to copy "myfile" from the diskette named "mydisk" to the diskette named "archive" you would first enter

```
        fm myfile  /CR/
```

and the M20 will respond with

```
            Please put SOURCE disk in drive,
        then press any key (CTRL C to abort)
```

Respond by inserting the diskette named "mydisk" in the drive and hitting any key. A message such as the following will then be displayed

```
        File transfer will take two pass(s)
            Please put DESTINATION disk in drive,
        then press any key (CTRL C  to abort) (pass 1 of 2):
```

Respond as instructed and the file will be copied into user memory.

The above message indicates that it will require two passes to transfer the file. To transfer the file the M20 must first copy it into user memory, then ask you to insert the target diskette, and finally transfer the file to the target volume. In this case, however, the file is too large to fit into user memory, therfore the transfer requires two passes.

Now insert the target diskette "archive" and hit any key as instructed. A file named "myfile" will be created on the target diskette (if it does not already exist) and the contents of the user memory copied into it. When this is completed the following message will be displayed

```
        Please put SOURCE disk in drive,
    then press any key (CTRL C  to abort) (pass 2 of 2):
```

Respond by removing the diskette, inserting the source diskette and hitting any key, as instructed, then the remainder of the file will be copied into user memory. When this is completed the following message will be displayed

```
        Please put DESTINATION disk in drive,
    then press any key (CTRL C  to abort) (pass 2 of 2):
```

Respond as instructed and the remainder of the file will be copied from user memory onto the target diskette and the process is then complete. The M20 responds with the PCOS prompt.

Note that it is not possible to use the wild card facility.

Copying Files on the Same Diskette

This process is the same as for copying files on the same diskette on a dual drive system. For example, to overwrite the contents of a file named "oldfile" with the contents of "myfile" - both being resident on the same diskette - insert the diskette in the drive and enter

```
    fc myfile,oldfile  /CR/
```

and the M20 responds

```
    COPY FILE 0:myfile TO 0:oldfile
    File already exists. Do you wish to overwrite (y or n)?
```

Respond by entering

```
     y  /CR/
```

and the copy process is executed. The target file keeps its original name. Furthermore, a password protected target file will also keep its own password. For example, if you enter

```
    fc %n dfile/passa,efile/passb /CR/
```

then "efile" is overwritten with the contents of "dfile", but the target file maintains its name "efile" and password "passb". Use of the no-interaction (%n) flag bypasses the dialogue.

LISTING FILES

To list a text file simply enter the FLIST command followed by the file identifier. For example, to list the file named "workfile" that is resident on the the diskette inserted in drive 1 enter

```
        fl 1:workfile /CR/
```

and a display such as the following will be generated.

```
LISTING FILE  1:workfile
10 subname$="getvolname"
20 volname$="12345678901234"
30 call "bv"(subname$,@volname$)
40 print "the current volume is named"; volname$
50 end
```

Fig. 10-2 Sample Display of a Text File List

Moreover, any file can be listed in hexadecimal form, 16 bytes per line, by including the %h program flag in the command line. For example, if you enter

```
        fl %h 1:workfile  /CR/
```

then the following display will be generated (in 80 by 25 display mode):

```
LISTING FILE  1:workfile
0000:  31 30 20 73 75 62 6E 61 6D 65 24 3D 22 67 65 74   10 subname$="get
0010:  76 6F 6C 6E 61 6D 65 22 0D 0A 32 30 20 76 6F 6C   volname"..20 vol
0020:  6E 61 6D 65 24 3D 22 31 32 33 34 35 36 37 38 39   name$="123456789
0030:  30 31 32 33 34 22 0D 0A 33 30 20 63 61 6C 6C 20   01234"..30 call
0040:  22 62 76 22 28 73 75 62 6E 61 6D 65 24 2C 40 76   "bv"(subname$,@v
0050:  6F 6C 6E 61 6D 65 24 29 0D 0A 34 30 20 70 72 69   olname$)..40 pri
0060:  6E 74 20 22 74 68 65 20 63 75 72 72 65 6E 74 20   nt "the current
0070:  76 6F 6C 75 6D 65 20 69 73 20 6E 61 6D 65 64 22   volume is named"
0080:  3B 20 76 6F 6C 6E 61 6D 65 24 0D 0A 35 30 20 65   ; volname$..50 e
0090:  6E 64 0D 0A 0D 0A                                 nd....
```

Fig. 10-3 Sample Display of a Hexadecimal File List

The first four columns specify the byte address relative to the beginning of the file (in hexadecimal) of the first character in the line. Each such entry is followed by the hexadecimal codes for the 16 bytes subsequent to the displayed address. The right-hand columns show the corresponding ASCII characters, or '.' where the character is non-printable (octal characters 0 to 31 and 127).

The text or hexadecimal contents of password protected files can be displayed, but the password must be specified to do so.

The contents of more than one file can be displayed by specifying a list of files in the command line. Moreover, groups of files can be listed by use of the wild card facility. However, password protected files can be displayed in this manner only if the password is common to all the files in the group.

Remark

A text listing is only useful where the specified file is a text file, but a hexadecimal listing of any file can be obtained.

PROTECTING FILES

PCOS files can be password protected to limit access to only those who know the password. Furthermore, they can be write-protected to prevent accidental damage to the contents of a file.

PASSWORD PROTECTION

Assigning and Removing a Password to a File

To assign a password to a file, enter the FPASS command along with the file identifier and the password to be assigned. For example, to assign the password "pass" to the file named "mine" that resides on the hard disk simply enter

```
        fp 10:mine,pass /CR/
```

At some later time, you may wish to remove this password protection. To do so requires the use of the FDEPASS command.

To use the FDEPASS command simply enter the command mnemonic along with the file identifier, including the password. In this case enter

```
        fd 10:mine/pass /CR/
```

Note that you must know the password to be able to remove it.

Assigning and Removing a Password to a Group of Files

To assign a password to a group of files is the same as for a single file except that the wild card feature is used. For example, to assign the password "rst" to all files beginning with "data..." on a diskette named "datadisk" you would enter

```
        fp datadisk:data*,rst /CR/
```

and the M20 would respond

```
        Set Password on 0:data1
```

Then a "y" response will set the password of the file named "data1" and corresponding interactive messages will be displayed for each subsequent file in the group. The PCOS prompt appears when the operation is complete.

To subsequently remove this password you would enter

```
        fd datadisk:data*/rst /CR/
```

and an interactive prompt for each file in the group appears thus:

```
        Delete Password of 0:data1?
```

A response of "y" deletes the corresponding password and the prompt reappears for the next file in the group. The PCOS prompt appears when the operation is complete.

To remove passwords in this way the password must be common to all the files in the group. **Note**: The dialogue in the above examples will be suppressed if the no-interaction (%n) flag is used.

WRITE-PROTECTION

Applying and Removing Write-Protection to a File

To apply write-protection to a file enter the FWPROT command along with the file identifier. For example, to write-protect a file named "masterfile" on a the diskette inserted in drive 1 enter

```
        fw 1:masterfile /CR/
```

To subsequently remove write-protection requires the FUNPROT command to be entered along with the file identifier. For example, to remove write-protection from "masterfile" enter

```
        fu 1:masterfile /CR/
```

Applying and Removing Write-Protection to a Group of Files

To write-protect a group of files, For example, all files beginning with "index" on the diskette inserted in drive 1, enter

```
        fw 1:index* /CR/
```

then the M20 displays a prompt for each file in turn in the specified group asking you to confirm or decline write-protection

```
        Set WP 1:index1?
```

Respond "y" and write-protection is set for the file named "index1"

```
        Set WP 1:index3.5?
```

A "y" response sets write-protection for the file named "index 3.5", etc.

To remove write-protection from the above group of files, enter

```
        fu 1:index* /CR/
```

and the M20 responds with a sequence of interactive prompts requesting confirmation for each file in the group. The PCOS prompt is displayed when the operation is complete.

Note: The dialogue in the above examples will be suppressed if the no-interaction flag (%n) is used.

FREEING UNUSED FILE BLOCKS

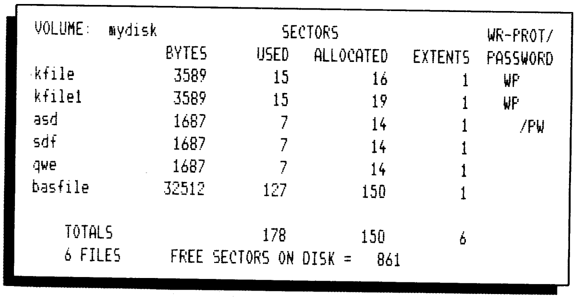
This section describes how to free blocks that are allocated to files but not used.

To free unused blocks of a file, group of files or an entire volume requires the FFREE command. Before this command can be executed the volume containing the files to be worked on must be present in one of the drives. It must also be enabled and must not be write-protected.

Before performing the FFREE command take a look at the contents of the volume to be worked on. For example, if you enter

```
vl 1:  /CR/
```

then a volume list of the contents of the diskette inserted in drive 1 will be displayed and will look something like the following:



```
VOLUME:  mydisk                   SECTORS                     WR-PROT/
                  BYTES      USED    ALLOCATED    EXTENTS   PASSWORD
kfile              3589        15           16          1      WP
kfile1             3589        15           19          1      WP
asd                1687         7           14          1        /PW
sdf                1687         7           14          1
qwe                1687         7           14          1
basfile           32512       127          150          1

    TOTALS                    178          150          6
    6 FILES        FREE SECTORS ON DISK =   861
```

Fig. 10-4 Sample Volume List before Freeing Unused Blocks

To free unused blocks in the entire volume perform the following:

Enter

```
ff 1:  /CR/
```

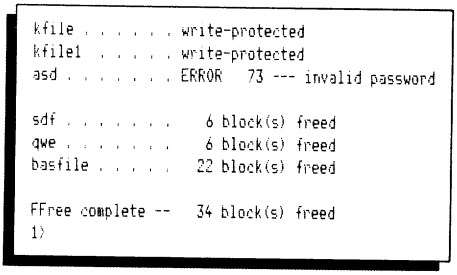
And the M20 will respond

```
You may not change disks while FFree in progress.  Continue?
```

Respond by entering

```
y  /CR/
```

and the display shown in Figure 10-5 will result.



```
kfile . . . . . . . write-protected
kfile1  . . . . . . write-protected
asd . . . . . . . . ERROR   73 --- invalid password

sdf . . . . . . . .    6 block(s) freed
qwe . . . . . . . .    6 block(s) freed
basfile . . . . . .   22 block(s) freed

FFree complete --     34 block(s) freed
1>
```

Fig. 10-5 Example of a Display Produced by an FFREE Command

The unused blocks allocated to "kfile" and "kfile1" are not freed because these files are write-protected. "asd" did not have unused blocks freed because it is password protected and did not have its password specified.

To free the unused blocks from file "asd" another FFREE command is necessary specifying the complete file identifier including the password. That is, if you enter

```
        ff 1:asd/pass  /CR/
```

M20 will display

```
        You may not change disks while FFree in progress.  Continue?
```

Respond by entering

```
         y  /CR/
```

then the unused blocks will be freed and the M20 will display

```
        asd . . . . . . . .  6 block(s) freed

        FFree complete --    6 block(s) freed
        1>
```

To free the unused blocks in files "kfile" and "kfile1" you must first remove the write-protection by entering

```
        fu 1:k*  /CR/
```

then, after answering "y" to the subsequent confirmation prompts, free the unused blocks by entering

```
        ff 1:k*  /CR/
```

M20 will respond

```
        You may not change disks while FFree in progress.  Continue?
```

respond by entering

```
         y  /CR/
```

and the M20 will display

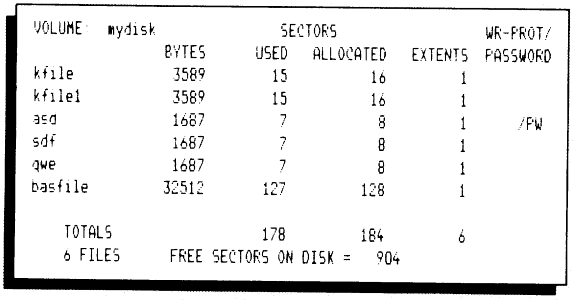
```
        kfile  . . . . . . .   0 block(s) freed
        kfile1 . . . . . . .   3 block(s) freed

        FFree complete --      3 block(s) freed
        1>
```

The total effect of freeing the unused blocks can be seen by examining a volume list and comparing it to to the previous one. That is, if you enter

```
        vl 1: /CR/
```

then the following will be displayed:



```
VOLUME:  mydisk                  SECTORS                      WR-PROT/
                   BYTES      USED   ALLOCATED   EXTENTS   PASSWORD
kfile               3589        15          16         1
kfile1              3589        15          16         1
asd                 1687         7           8         1      /PW
sdf                 1687         7           8         1
qwe                 1687         7           8         1
basfile            32512       127         128         1

   TOTALS                      178         184         6
   6 FILES       FREE SECTORS ON DISK =    904
```

Fig. 10-6 Sample Volume List after Freeing Unused Blocks

Finally, write-protection should be re-applied to files "kfile" and "kfile1".

DELETING AND RECOVERING FILES

Deleting a file implies freeing the disk space occupied by that file for use by other files. The actual file data, however , is not deleted until the space it occupied is actually overwritten. This makes it possible to recover a deleted file so long as the file has not yet been overwritten and the volume has not been alphabetised.

DELETING FILES

To delete a file or group of files requires the FKILL command. Before this command can be executed the following conditions must apply:

- The volume containing the file or files to be deleted must be inserted in one of the drives. It must also be enabled and not write-protected

- The file or files to be deleted must not be write-protected

To delete a file from a volume requires you to enter the FKILL command along with the file identifier including any necessary volume identifier and/or password. For example, to delete a file that has the name "myfile" and password "mine" from the diskette inserted in drive 1 enter

```
        fk 1:myfile/mine /CR/
```

A list of files can be specified for deletion. For example, to delete the (unprotected) files "myfile" and "yourfile" from the diskette inserted in drive 0 enter

```
        fk 0:myfile,yourfile  /CR/
```

A killed file is not actually deleted from the diskette, but it can no longer be accessed by any command other than the RKILL command. Furthermore, its name will no longer feature in a volume list as it is erased from the directory, and the space it occupied will be available for other files.

Groups of files can be deleted using the wild card feature. For instance to delete from the diskette inserted in drive 0 all the files beginning with "k" you must enter

```
        fk 0:k*  /CR/
```

and the M20 will respond

```
        Delete  0:kfile?
```

which gives you the option whether or not to delete this particular file. If you enter

```
         y
```

then the file is deleted and the M20 responds

```
        Delete  0:kfile1?
```

and so on through the complete list of files beginning with "k", after which the PCOS prompt appears. Note: The dialogue can be by-passed by specifying the no-interaction (%n) flag.

RECOVERING DELETED FILES

To recover a deleted file requires the RKILL command. For this command to be executed successfully the following conditions must apply:

- The volume containing the file to be recovered must be inserted in one of the drives
- The deleted file to be recovered must still be intact. That is, it must not have been fully or partially overwritten
- The volume must not have been alphabetised since the file was killed

To recover a file requires you to enter the RKILL command along with the file identifier. It is not necessary to enter the password. For example, to attempt to recover the deleted file "myfile" on the diskette inserted in drive 1 enter

```
        rk 1:myfile  /CR/
```

If the recovery is successful then the M20 will respond

```
        File Successfully Repaired
```

If the recovery was not successful then the following error message is displayed:

```
        ERROR 53 --- file not found in parameter 1
```

Note: It is not possible to use the wild card facility, neither is it possible to specify a list of files.

RENAMING FILES

Renaming a file requires the FRENAME command. Before this command can be executed the file to be renamed must reside on a volume that is currently inserted in one of the drives.

To rename a file requires you to enter the file identifier along with the new file name. The file must not be write-protected.

For example, to change to "oldfile" the name of a file named "newfile" with password "npass" that is resident on a disk named "datadisk" enter

```
        fr datadisk:newfile/npass,oldfile /CR/
```

Note: FRENAME has no effect on the password.

11. PCOS GRAPHIC AND CONSOLE - RELATED FACILITIES

ABOUT THIS CHAPTER

This chapter describes the graphics facilities that are available within the PCOS environment. For further details of the commands mentioned in this chapter, refer to Chapter 13.

CONTENTS

INTRODUCTION

PCOS allows you to set the display mode to either 256 x 512 pixels (64 columns x 16 lines) or 256 x 480 pixels, (80 columns x 25 lines) by means of the SSYS command, and to allocate space for a certain number of windows using the SBASIC command. Moreover, PCOS contains a number of graphic and console-related facilities that enable you to:

- display a label. That is, to display a string of characters of a specified magnification and orientation at a specified position, and within a specified window (LABEL command)
- print out the entire contents of the video display or a specified window (SPRINT command)
- print out just the textual content of a video display or a specified window (LSCREEN command)
- modify the ASCII code generated on striking a specific key or key combination (CKEY command)
- reconfigure the keyboard to simulate another standard national layout (SLANG command)
- assign a string to a key or key combination (PKEY command)
- create user-defined fonts (RFONT and WFONT commands)
- display control characters
- distinguish, in BASIC, among the three line termination keys /↵/, /S1/ and /S2/ (LTERM command)

Note: The LABEL, SPRINT, and LSCREEN commands are often called from BASIC by a CALL or EXEC statement, as windows cannot be openned within the PCOS environment.

DISPLAYING LABELS

Labels can be displayed by using the LABEL command. This command enables you to specify the following features of the label to be displayed.

FEATURE	MEANING
the string	the text to be displayed. This can be any string of printable characters, included in quotation marks
position	the x/y co-ordinates of the bottom left-hand corner of the first character of the label string with respect to the bottom left-hand corner of the screen or window. This is measured in pixels
magnification	the number of times that the font character is magnified. This can be up to 16 times
orientation	the number (0, 1 or 2) specifying the direction of the label string; that is, parallel to the x-axis left to right (0), parallel to the y-axis bottom to top (1), or top to bottom (2)
colour	the colour number in the range 0 to 7 for an eight colour display, 0 to 3 for a four colour display, or 0 or 1 for a black and white display. If omitted, the foreground colour is assumed

Using the LABEL command

If you enter

```
        la 'LABEL',100,150,4,2   /CR/
```

then the string "LABEL" is displayed in the foreground colour, at

(100,150) four times the normal size, with the text rotated as follows:

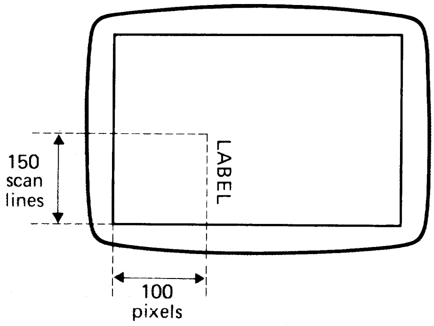


Fig. 11-1 Example of the LABEL command

PRINTING THE SCREEN IMAGE

There are two PCOS commands that perform this type of function. These are:

SPRINT which prints an image of all text and graphics displayed on the screen or within a specified window. With this command you are also able to specify the polarity of the print-out, (that is, n for negative - black on white on print-out for white on black on display, p for positive - white on black on print-out for white on black on display). You can also specify a title to appear at the top of the print-out along with the date and time.

The SRRINT command can, however, only be used with printers that have graphic capabilities.

LSCREEN which prints just the text displayed on the screen or specified window. Graphic elements are ignored.

The LSCREEN command can be used with any PCOS-compatible printer

USING THE SPRINT AND LSCREEN COMMANDS

Supposing, when working in the BASIC environment, you have segmented the screen into five windows as follows:

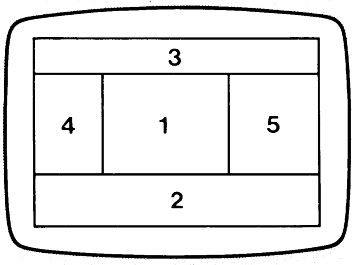


Fig. 11-2 Example of a Windowed display

IF you are in BASIC and you enter ...	THEN...
EXEC "sp 5,p,'SALES', dt" /CR/	the contents of window 5 will be printed with positive polarity beneath the heading "SALES" and the date and time (specified by dt)
EXEC "sp 0" /CR/	prints the contents of the entire screen
EXEC "ls 4" /CR/	prints the text contained in window 4. Any graphic elements are ignored

For details of how to define windows refer to the "BASIC Language Reference Guide".

ENTERING CHARACTERS AT THE KEYBOARD

Figure 11-3 illustrates what happens when you press a key.

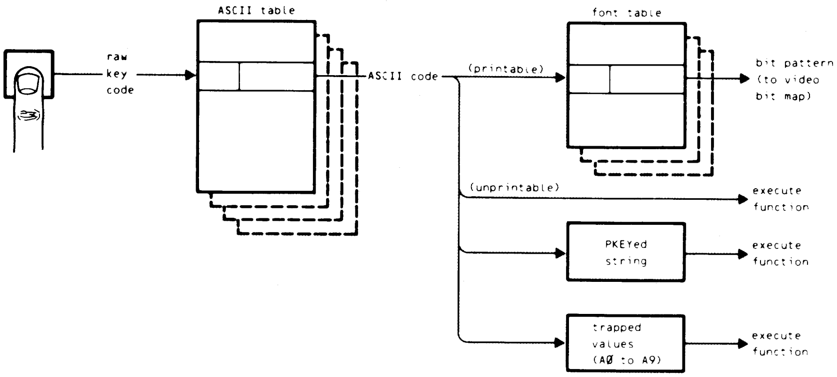


Fig. 11-3 Entering a Character at the keyboard

RAW KEY CODES

When a key is pressed it generates a raw key code (see Figure 11.4). This code depends on the physical position of the key on the keyboard, and on whether the key is pressed on its own, or in conjunction with the /SHIFT/ key, the /CTRL/ key, or the /COMMAND/ key. For example, if you press the 'A' key on the USA ASCII keyboard then raw key code (hexadecimal) 02 will be generated. The same raw key code will be generated by pressing 'Q' on the French keyboard since the code depends on the physical position of the key, not the key inscription. Moreover, pressing this key in conjunction with the /SHIFT/ key will generate raw key code (hexadecimal) 32, with the /CTRL/ key will generate (hexadecimal) 62, and with the /COMMAND/ key will generate (hexadecimal) 92.

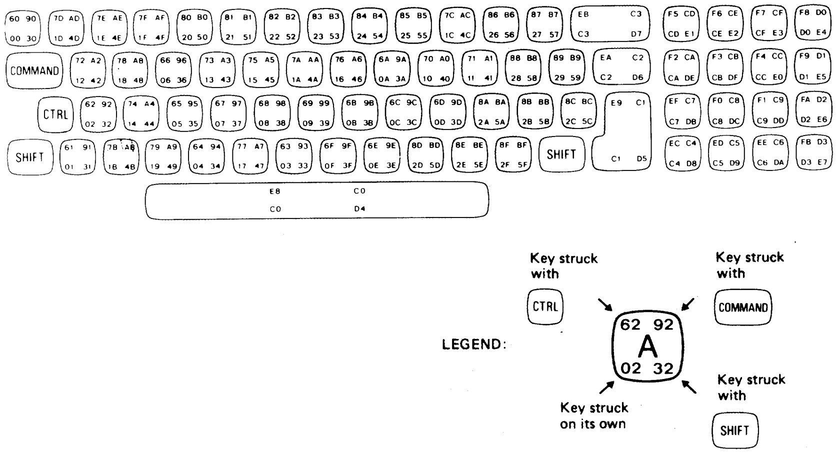


Fig. 11-4 Raw Key Codes

ASCII TABLES

The raw key code points directly into an ASCII table that was loaded at initialisation from the kb.all file. This table generates the appropriate ASCII code that corresponds to the inscription on the keytop. The ASCII table therefore varies from one national keyboard to another. For example, if you strike 'A' on the USA ASCII keyboard, raw key code 02 will be generated, which in turn will be converted by the USA ASCII table to (hexadecimal) 61 - the ASCII code for 'a'. Similarly, on the France keyboard, striking 'Q' will generate raw key code 02 which will be converted by the ASCII table corresponding to the France keyboard into the ASCII code for 'q' - (hexadecimal) 71. For a complete list see Appendix B.

The values of individual entries can be changed by use of the CKEY command. Moreover, the SLANG command can be used to replace the current ASCII table with one that corresponds to another national keyboard.

Using the CKEY command

Any entry in the current ASCII table can be re-defined using the CKEY command. To do this you need to include two parameters in the CKEY command; the first defining the key or key combination to be re-defined, and the second the ASCII code to be assigned to it. For example

```
        ck &C3,8  /CR/
```

assigns ASCII code 8 (backspace) to the raw key code (hexadecimal) C3, thereby enabling backspace to be performed by striking the S2 key. Note that values may be entered either in decimal (by entering the number on its own, or in hexadecimal) by preceeding the number with &. Furthermore, the ASCII code can be specified simply by striking the corresponding key enclosed within quotation marks. For example

```
        ck &72,'p'  /CR/
```

will cause the ASCII code for 'p' to be generated when /CTRL/ /Q/ are pressed simultaneously.

The ASCII code assigned to a particular raw key code can be examined, again by use of the CKEY command by specifying just one parameter - the raw key code. For example, if you wish to examine the code assigned to the key combination /CTRL/ /C/, enter

```
        ck &64  /CR/
```

then PCOS responds

```
        KEY  = 100
        CODE = 162
```

The values displayed are decimal. That is, KEY specifes the raw key code (100 decimal being the equivalent of 64 hexadecimal), and CODE indicates the assigned code.

Using the SLANG Command

The ASCII table can be replaced at will by the ASCII table corresponding to another national keyboard by means of the SLANG command. Doing so also destroys any values assigned using the CKEY command. If you enter

```
        sl  /CR/
```

then a menu is displayed enabling you to select a national configuration by entering the appropriate number (see Chapter 6 for details). Alternatively, the selection can be made by entering the appropriate number as a parameter to the SLANG command. For example

```
        sl 0  /CR/
```

selects the Italy keyboard. The result is that the ASCII table corresponding to the Italy keyboard is loaded from the kb.all file and overwrites the currently active table.

Making the Current ASCII Table Permanent

Any modifications made to the ASCII table by means of the CKEY command, or any ASCII table made active by means of the SLANG command remain active either until the current working session is terminated, or until further modified by a CKEY or SLANG command. However, such changes can be made a permanent feature of the operating system by means of the PSAVE command (see Chapter 6); that is, any CKEYed values will become permanent, as will any ASCII table loaded by means of the SLANG command.

INTERPRETATION OF ASCII TABLE OUTPUT

The output from the ASCII table is treated as follows:

- values A0 to A9 (hexadecimal) are special cases. They are never placed in the keyboard buffer but are always trapped by the keyboard handler to perform the following functions:

  . A0 - Logical Reset

  . A1 - (reserved)

  . A2 - Break facility

  . A3 - Halt Display

  . A4 - Cursor lock

  . A5 - Shift lock

  . A6 - Two zeros

  . A7 - End of line (CR in keyboard buffer, zero in LTERM buffer)

  . A8 - End of line (CR in keyboard buffer, '1' in LTERM buffer)

  . A9 - End of line (CR in keyboard buffer, '2' in LTERM buffer)

  . AA - Special function for DATEV keyboard

  . AB - Special function for DATEV keyboard

  . AC - Special function for DATEV keyboard

  . AD - (reserved)

  . AE - (reserved)

  . AF - No operation

- for values that have a PKEYed string assigned to them the value is placed in the buffer and the corresponding function is subsequently executed

- for unprintable ASCII values (codes (hexadecimal) 00 to 1F and 7F) the code is placed in the keyboard buffer and the corresponding function is subsequently performed

- for printable ASCII characters other than those that have strings assigned by means of the PKEY command (codes (hexadecimal) 20 to 7E plus any additional CKEYed values) the code is placed in the keyboard buffer and the corresponding entry in the currently active font table is accessed and the bit pattern is written to the video bit map - see the section entitled "Creating User-Defined Fonts"

USING THE PKEY COMMAND

Any code output from the ASCII table (with the exception of the special cases A0 to AF) can have a string assigned to it by means of the PKEY command, thereby ensuring that the original function is not destroyed.

Assignment is made by passing parameters to the PKEY command, the first of which specifies the code as output from the ASCII table, and the second and subsequent parameters define the string to be assigned.

The ASCII code can be specified either as a decimal value on its own, a hexadecimal value preceeded by an ampersand (&), or in the case of an ASCII code that already has a corresponding font defined in the currently active font table, the character can be entered directly from the keyboard, but enclosed within quotation marks. For example

```
'B'

66

&47
```

all refer to the same key.

Similarly, the strings to be assigned to the ASCII code can be specified as either the actual characters enclosed within quotation marks, or the ASCII code for each character, or a combination of the two. For example

```
'ba',13
```

is a valid string representing 'ba' followed by a carriage return.

Suppose that you wish to assign the string 'FILES "1:"',13 to the key combination /COMMAND/ /!1/ (ASCII code hexadecimal ED, decimal 237). Do this by entering

```
pk 237,'FILES "1:"',13  /CR/
```

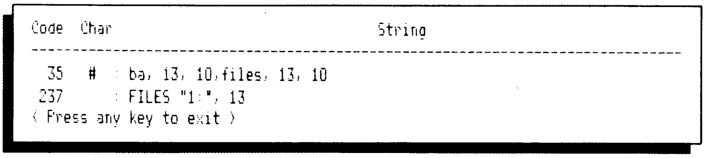
Thus when in the BASIC environment, the key combination /COMMAND/ /!1/ can be used to list the directory on the diskette inserted in drive 1.

String assignments made in this manner are valid up to the end of the current working session. However, such assignments can be made a permanent feature of the operating system by means of the PSAVE command as described in Chapter 6.

The template that fits into the slot above the top row of the keyboard can be used as a memory aid to the keys that you have programmed. Alternatively you can display a list of programmed keys along with the string assignments by entering

```
        pk  /CR/
```

PCOS will typically respond:



```
Code  Char                                String
----------------------------------------------------------------------------------
  35   #   : ba, 13, 10,files, 13, 10
 237       : FILES "1:", 13
( Press any key to exit )
```

Fig. 11-5 Sample Display of Programmed Keys

Note that the code is given in decimal.

An individual key assignment may be cancelled by entering the ASCII code as a single parameter to the PKEY command. For example

```
        pk 237 /CR/
```

Moreover, all assigned strings may be cancelled by entering

```
        pk %c /CR/
```

For details of the effect sting assignments have on user memory refer to Chapter 6.

CREATING USER-DEFINED FONTS

The RFONT command creates a graphic representation of the active font set and stores it in a font matrix file. This file can be edited using the Video File Editor (see Chapter 12) to redefine the shape of existing characters and add new characters. Once the editing session is complete you can invoke the WFONT command to cause the system to display character fonts as they appear in the edited file. Thus you can create customised font sets, each of which is stored on diskette (or hard disk) in a separate file, and can be selected to become the active font set.

CREATING A FONT MATRIX FILE USING THE RFONT COMMAND

The RFONT command is invoked by entering, for example

```
      rf 1:myfont /CR/
```

The specified file will be created if it does not exist. If the file already exists, the following prompt will appear:

```
      File Already Exists. Do You Wish to Overwrite? (y/n)
```

A "y /CR/" response causes the existing file to be overwritten.

THE FONT MATRIX FILE

A font matrix file must be structured as defined below. A file created by the RFONT command is of this structure.

At the beginning of a font matrix file is a four line header. All four lines must be present, although only the fourth line is actually read by the WFONT command. The header is defined as follows:

- line 1: Any text that describes the file (for example, the national keyboard that the file corresponds to). It is for your reference and is ignored by the WFONT command

- line 2: The country number that was active when the file was created by the RFONT command. This number corresponds to the particular national keyboard. For details refer to the SLANG command. The content is ignored by the WFONT command

- line 3: The height (in lines) of a valid font matrix. This must always be 10

- line 4: The character count followed by at least one other word (for example, "characters"). The count must match the number of font matrices that follow. The minimum is 95 characters and the maximum is 190

Example:

```
USA
country 4
matrix height = 10
95 characters
```

Each matrix is defined as follows:

- line 1:	A decimal code representing the character defined in the matrix. (For standard fonts this will be the ASCII code.) Its value is for your reference and is not read by the WFONT command, but it must be present
- lines 2 to 11:	A matrix, ten lines down and eight characters wide, made up of ' - ' and (upper case) 'X' characters

Example:

```
50
--------
----XXX-
---X---X
-------X
------X-
-----X--
----X---
---XXXXX
--------
--------
```

The correspondence between a font matrix and the ASCII code generated by a particular key on the keyboard depends on the position of the font matrix within the file. That is, the first font matrix will correspond to /SPACE/ - the first printable ASCII character (ASCII code 32) - the second to /!/ (ASCII code 33) and so on up to the 95th entry which will correspond to /±/ - the last printable ASCII character (ASCII code 126).

Redrawing Existing Characters

To redraw a character, invoke the Video File Editor and place 'X's so that they show the intended appearance of the character.

In 64 x 16 display mode, the left most three columns are generally reserved for spacing between characters and should not be used except in special cases where regular spacing between characters is not desired. In 80 x 25 display mode, the left most two columns are not displayed at all, and the third column should be left blank unless joined characters are desired.

Once the edited file is saved and the Video File Editor exited, the new font can be made active by means of the WFONT command.

Defining Additional Characters

Characters can be added to a font matrix file by incrementing the character count (line 4 of the header) and adding matrices to the end of the file. However, matrices beyond the 95th, (that is, those corresponding to codes 127 to 222) do not necessarily have a corresponding key-generated ASCII code. You must therefore define a key combination for each additional matrix using the CKEY command.

Example:

To add a font matrix, defining the character 'Ø', to the standard font matrix file and assign key combination /COMMAND/ /Q/ to it:

1. Using the Video File Editor update the character count (line 4 of the header) to 96

2. Using the Video File Editor add a font matrix to the end of the file, thus:

```
127
--------
----XXX-
---X---X
---X--XX
---X-X-X
---XX--X
---X---X
----XXX-
--------
--------
```

3. Assign ASCII code 127 to the key combination /COMMAND/ /Q/ using the CKEY command:

```
ck &A2, 127 /CR/
```

   This assigns code 127 to raw key code (hexadecimal) A2, which is the raw key code generated when /COMMAND/ /Q/ are pressed simultaneously

Once the edited font has been made active by means of the WFONT command, the key combination /COMMAND/ /Q/ will display 'Ø'.

Note: It is possible to remove or insert font matrices within the first 95 characters, but this would offset the matrix/key correspondence following the first matrix to be removed/inserted.

USING THE WFONT COMMAND

The WFONT command takes one parameter which is the name of a font matrix file. After execution, the font defined by that file becomes active.

Invoke the WFONT command by entering, for example

```
    wf 1:myfont /CR/
```

Assuming the specified font matrix file is on an active volume, and that enough memory is available, the font matrices will be read, converted to binary, and written into memory. Once execution is completed, the new fonts will replace those currently known to the system. The system will return to the fonts known at initialisation when re-initialisation occurs, or when the WFONT command is invoked with no parameter.

The WFONT command allocates user memory each time it is invoked with a valid font file. This memory is released either by re-initialisating PCOS, or by invoking the WFONT command with no parameter. In order to save memory, it is advisable to release space allocated by the WFONT command before activating another user-defined font. The effect on memory can be determined using the DCONFIG command.

User-defined fonts can be made permanent by means of the PSAVE command.

Example

The following example assumes the existance of two user-defined font matrix files named "myfont" and "yourfont." Both files are located on the diskette inserted in drive 1.

If you enter...	THEN...
wf 1:myfont /CR/ . . .	the font defined by the font matrix file named "myfont" is made active
wf /CR/ . . .	"myfont" is removed from memory and the font that was active at initialisation is again made active

wf 1:yourfont /CR/ . . .	the font defined by the font matrix file named "yourfont" is made active
wf 1:myfont /CR/ . . .	"myfont" is again made active, but data for "yourfont" is not removed from memory

The effect on user memory is illustrated in Figure 11-6

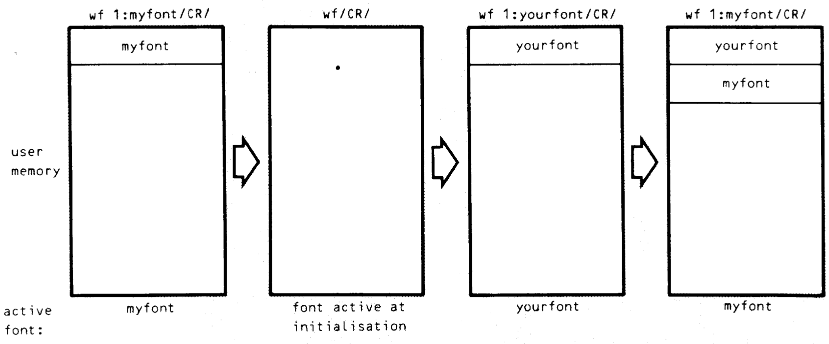


Fig. 11-6 Effect of the WFONT command on user memory

DISPLAYING CONTROL CHARACTERS

When displaying files or displaying data received from a communications line, you may wish to have a visual representation of the ASCII control characters (hexadecimal 00 to OF). PCOS contains a facility whereby such characters can be represented by the character font definitions illustrated in Figure 11-7. Above each matrix in the figure is the corresponding hexadecimal ASCII code in parentheses, and the control function.

```
(00)        (01)        (02)        (03)        (04)        (05)
NUL         SOH         STX         ETX         EOT         ENQ
--------    --------    --------    --------    --------    --------
---XXXXX    ---XXXXX    -----X--    -------X    ----X---    ---XXXXX
---XXXXX    ---X----    -----X--    -------X    -----X--    ---X---X
---XXXXX    ---X----    -----X--    -------X    ------X-    ---XX-XX
---XXXXX    ---X----    -----X--    -------X    -----X--    ---X-X-X
---XXXXX    ---X----    -----X--    -------X    ----X---    ---XX-XX
---XXXXX    ---X----    -----X--    -------X    -----X--    ---X---X
---XXXXX    ---X----    ---XXXXX    ---XXXXX    ------X-    ---XXXXX
--------    --------    --------    --------    --------    --------
--------    --------    --------    --------    --------    --------

(06)        (07)        (08)        (09)        (0A)        (0B)
ACK         BEL         BS          HT          LF          VT
--------    --------    --------    --------    --------    --------
-------X    --------    ---XXX--    ----X---    ---XXXXX    -----X--
--------    --------    ---XX---    -----X--    --------    -----X--
------X-    --------    ---X-X--    ------X-    --------    -----X--
--------    ----XXX-    ------X-    ---XXXXX    ---XXXXX    -----X--
---X-X--    ---X---X    -------X    ------X-    --------    ---X-X-X
--------    ---X---X    -------X    -----X--    --------    ----XXX-
----X---    ---XXXXX    -------X    ----X---    ---XXXXX    -----X--
--------    --------    --------    --------    --------    --------
--------    --------    --------    --------    --------    --------

(0C)        (0D)        (0E)        (0F)        (10)        (11)
FF          CR          S0          S1          DLE         DC1
--------    --------    --------    --------    --------    --------
-----X--    ------X-    ----XXX-    ----XXX-    ---XXXXX    ----XXX-
---X-X-X    -----X--    ---X---X    ---X---X    ---X---X    ---X-X-X
----XXX-    ----X---    ---XX-XX    ---X---X    ---X---X    ---X-X-X
-----X--    ---XXXXX    ---X-X-X    ---X-X-X    ---XXXXX    ---X-XXX
---X-X-X    ----X---    ---XX-XX    ---X---X    ---X---X    ---X---X
----XXX-    -----X--    ---X---X    ---X---X    ---X---X    ---X---X
-----X--    ------X-    ----XXX-    ----XXX-    ---XXXXX    ----XXX-
--------    --------    --------    --------    --------    --------
--------    --------    --------    --------    --------    --------

(12)        (13)        (14)        (15)        (16)        (17)
DC2         DC3         DC4         NAK         SYN         ETB
--------    --------    --------    --------    --------    --------
----XXX-    ----XXX-    ----XXX-    -------X    ----XXX-    --------
---X---X    ---X---X    ---X-X-X    --------    ----X-X-    -------X
---X---X    ---X---X    ---X-X-X    -----XXX    ----X-X-    -------X
---X-XXX    ---XXX-X    ---XXX-X    --------    ----X-X-    ---XXXX-
---X-X-X    ---X-X-X    ---X---X    ---X-X--    ----X-X-    -------X
---X-X-X    ---X-X-X    ---X---X    --------    ----X-X-    -------X
----XXX-    ----XXX-    ----XXX-    ----X---    ---XX-XX    --------
--------    --------    --------    --------    --------    --------
--------    --------    --------    --------    --------    --------
```

```
(18)       (19)       (1A)       (1B)       (1C)       (1D)
CAN        EM         SUB        ESC        FS         GS
--------   --------   --------   --------   --------   --------
---XXXXX   -----X--   ----XXX-   ----XXX-   ---XXXXX   ---XXXXX
---X---X   -----X--   ---X---X   ---X---X   ---X-X-X   ---X---X
----X-X-   -----X--   ---X----   ---X---X   ---X-X-X   ---X---X
-----X--   ----XXX-   ----X---   ---XXXXX   ---XXX-X   ---XXX-X
----X-X-   -----X--   -----X--   ---X---X   ---X---X   ---X-X-X
---X---X   -----X--   --------   ---X---X   ---X---X   ---X---X
---XXXXX   -----X--   -----X--   ----XXX-   ---XXXXX   ---XXXXX
--------   --------   --------   --------   --------   --------
--------   --------   --------   --------   --------   --------

                      (1E)       (1F)
                      RS         US
                      --------   --------
                      ---XXXXX   ---XXXXX
                      ---X---X   ---X-X-X
                      ---X---X   ---X-X-X
                      ---X-XXX   ---X-XXX
                      ---X-X-X   ---X---X
                      ---X-X-X   ---X---X
                      ---XXXXX   ---XXXXX
                      --------   --------
                      --------   --------
```

Fig. 11-7   Control Character Font Matrices

Control characters that produce a visible/audible effect require special explanation. These are described in the following table:

CONTROL CHARACTER	COMMENTS
TAB (09) LF (0A) FF (0C) CR (0D)	Any of these characters will simply appear as a single character on the screen without performing the corresponding visible effect. That is, characters will 'wrap around' the screen as a continuous line
BEL (07)	The bell will sound and the character will appear on the screen

Table 10-1   Control Charactrers that Produce a Visible/Audible Effect

Control character display can be specified:

- locally	as a parameter to a PCOS command for the duration of that command
- globally	as a directive that will remain active until cancelled, or until the current working session is terminated

In either case control character display is activated by specifying "+cc", or cancelled by specifying "-cc".

By default control characters are not displayed, but control character display can be made a permanent feature of the operating system if it is active when the PSAVE command is executed.

Examples

IF you enter...	THEN...
flist +cc 1:myfile /CR/	a text listing of "myfile" is displayed incl-uding any control characters. On completion, control character display is cancelled
+cc /CR/	control character display is activated glob-ally; that is, until either re-specified, or until the end of the current working session
vq -cc /CR/	1. If control character display is set glob-ally, it will be cancelled for the dura-tion of the VQUICK command. On completion of the command, control character display will be re-activated 2. If control character display is not set globally, "-cc" will have no effect
-cc /CR/	control character display is cancelled until respecified

THE LINE TERMINATION KEYS

For most operations the three line termination keys perform the identical function of placing the ASCII code for carriage return (08) in the keyboard buffer, irrespective as to whether the key is struck on its own or in conjunction with one of the keys /SHIFT/, /CTRL/ or /COMMAND/. Alternative functions may be assigned by means of the CKEY command by assigning different ASCII codes to the raw key codes generated; but note that the PKEY command can only assign a string to the ASCII code placed in the keyboard buffer and therefore cannot assign unique strings to each of the keys (unless unique ASCII codes have first been assigned using the CKEY command).

Regardless of the CKEY command, PCOS contains a facility whereby the three line termination keys can be distinguished among the three from within a BASIC program. In addition to placing the ASCII code in the keyboard buffer, striking one of these keys also places a unique code in a special (LTERM) buffer (0, 1 or 2 for /↵/, /S1/ or /S2/, respectively). This buffer can then be interrogated from BASIC using the LTERM command to distinguish which of the three keys was last pressed. This can be useful in a situation where a BASIC program prompts the user for an entry of some sort. The LTERM command can subsequently be CALLed to return the current value of the LTERM buffer, in order to process the entry in one of three ways depending on which of the three line termination keys was used to terminate the entry.

12. VIDEO FILE EDITOR

ABOUT THIS CHAPTER

This chapter describes how files containing text can be edited using the Video File Editor.

CONTENTS

INTRODUCTION

The Video File Editor enables you to create and edit files of text. A text file is a file of records containing printable ASCII characters, and each record is separated from the next either by a carriage-return/line-feed pair or by the record separator character RS.

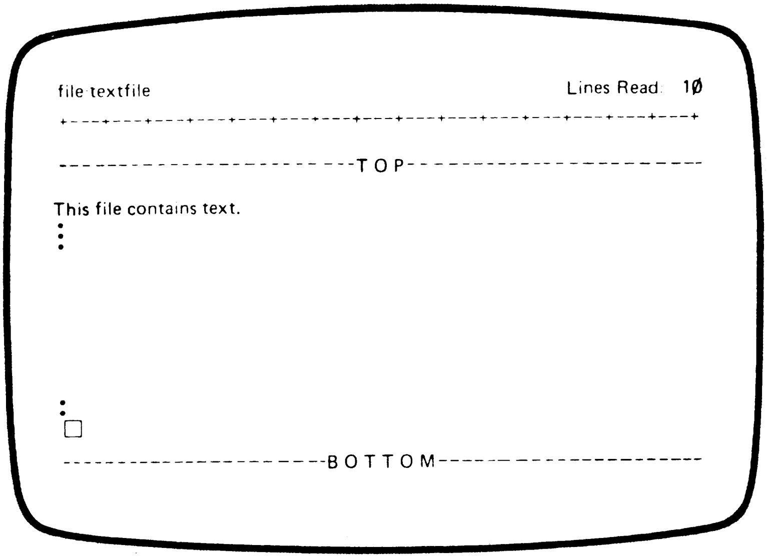
The Video File Editor displays a 21-line "window" within which you can perform editing functions via the keyboard. A subset of these functions enables you to move the window to access any part of the file.

In addition to the functions mentioned above the Video File Editor can also perform an extensive set of line editing and cursor moving functions and can operate in overstrike, insert text or command mode. The latter enables a subset of high level commands.

Each text line can contain up to 80 characters.

THE DISPLAY

Once the Video File Editor has been invoked the M20 produces a display such as the one shown in Figure 12-1.



```
file textfile                                          Lines Read  1Ø
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
-------------------------TOP-------------------------
This file contains text.
•
•
•



•
•
□
-----------------------BOTTOM-----------------------
```

Fig. 12-1 Video File Editor Screen Layout

Line 1 indicates the file name and the current message.

Line 2 is used for high level commands and search strings and is therefore only used when in command mode. Refer to the section entitled "Commands and Searching" for details.

Line 3 shows the tab stop settings (4 character positions per tab).

Lines 4 to 24 contain the text window.

Line 25 is not used.

On entering the Video File Editor the beginning and end of the file are marked by two display lines containing the words TOP and BOTTOM, respectively. The former, known as the TOP bar, always appears immediately before the first line of text in the file. And the BOTTOM bar always appears immediately after the last line of text. They are not actual lines of text and are there merely as markers. The cursor is positioned on the TOP bar.

The cursor changes shape when switching between certain modes of editing. It is represented here as underline.

THE KEYBOARD

The keyboard functions in a different manner once the Video File Editor has been invoked. This provides the means by which the required editing functions are entered. The name of each function key is shown below.

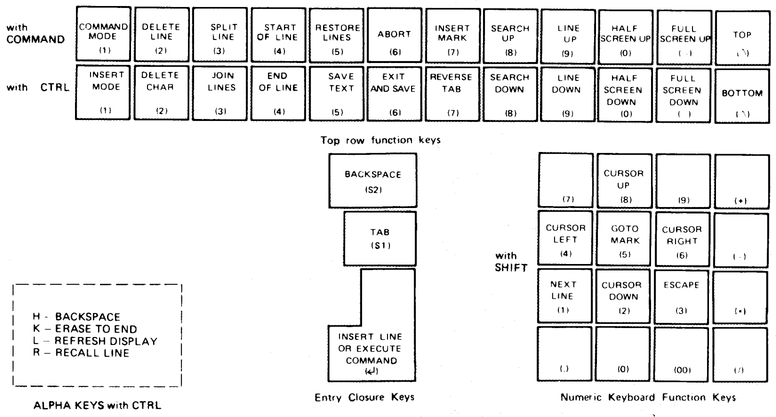


Fig. 12-2 Location of Video File Editor Function Keys

Note: The values in parentheses shown in Figure 12-2 represent the appropriate key on the USA ASCII keyboard. Refer to Appendix B for other keyboard layouts.

The function keys are divided into four areas:

The Numeric Keypad

Pressing the SHIFT key in conjunction with keys on the numeric keypad provides a set of functions primarily for cursor motion, as inscribed on the upper half of the keys themselves. Keys 1, 3 and 5 have other functions and are described later. Note that the numeric keypad can be locked into shift mode by means of the /CTRL/ /!// keys. Subsequently, the numeric values can be entered by means of the /SHIFT/ key. Return to unshifted mode can be made by entering /CTRL//!//.

The Top Row

Twelve of the top row of keys are used in conjunction with the /COMMAND/ and /CTRL/ keys to provide 24 functions. These perform or enable most of the major editing operations such as moving the text window, saving text, inserting text and switching between different modes of editing.

Alphabetic Keys

Some of the alphabetic keys when used with the /CTRL/ key provide additional functions.

Entry Closure Keys

These three keys are used for the most frequently used editing functions. They require no shift key.

Programmed Function Keys

You are free to re-arrange function keys at will (while in the PCOS environment) using either the CKEY or PKEY command. For example, if you prefer to have the INSERT MARK function assigned to the /CTRL/ /|/ key (bottom left on the USA ASCII keyboard) enter

```
     pk &7F,&F3 /CR/
```

Moreover, frequently entered text strings can be assigned to a single key value by means of the PKEY command. However, YOU MUST TAKE CARE NOT TO DISABLE FUNCTIONS THAT YOU WILL REQUIRE AFTER ENTERING THE VIDEO FILE EDITOR. For example, if you assign some value to the key combination /CTRL/ /6/, then the corresponding edit function (EXIT AND SAVE) will be disabled.

HOW TO INVOKE THE VIDEO FILE EDITOR

The EDIT command is used to enter the Video File Editor.

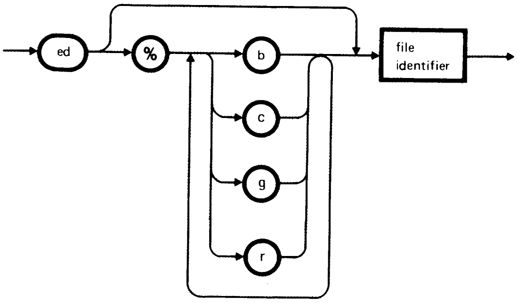


Fig. 12-3 EDIT

Where

SYNTAX ELEMENT	MEANING
b	a backup of the file is to be made when the Video File Editor is entered. This back-up is named filename.bak where filename is the same as that specified in the command line

c	the option which allows you to edit BASIC program files. This is necessary because BASIC files have a different format from standard text files. That is, they contain no tab characters (even if you enter TAB or REVERSE TAB) and have carriage-return/line--feed pairs between lines. Conversely, standard PCOS text files do have tab characters, moreover they delimit lines by means of record separator characters (RS = hexadecimal 1E). Note: if the BASIC program file already exists it must be an ASCII file. Moreover, if a line of such a file contains more than 80 characters, characters beyond the 80th will be lost
g	a special flag that must be included when invoking the Video File Editor from the Greece keyboard
r	the read-only option and is used when you only wish to examine the contents of the file. This protects the file from accidental damage while examining it
file identifier	the name of the file to be edited plus any necessary password or volume identifier

Characteristics

If the file does not already exist the prompt "OK to Create?" appears on the screen to which you must type "y" to create the file.

The Video File Editor is initially in "overstrike" mode. That is, you can enter text and overwrite whatever is already written on the file. The methods of entry into other modes of operation are described later.

The four optional flags (b, c, g and r) can be specified in any order.

Note: For the special project Delta keyboard, %h must be entered in the EDIT command line.

GENERAL EDITING FUNCTION KEYS

The keys whose functions are described below perform general editing functions such as moving the cursor and inserting and deleting text.

CLASS	FUNCTION KEY	MEANING
to move the cursor	/SHIFT/ /8/ (CURSOR UP)	moves the cursor one line up the screen but keeps the same position within the line. If the cursor was on the second line of the window then the window is moved one line up the file and the cursor remains on the second line
	/SHIFT/ /2/ (CURSOR DOWN)	moves the cursor one line down the screen but keeps the same position within the line. If the cursor was on the penultimate line of the window, it stays there and the window is moved down one line
	/SHIFT/ /4/ (CURSOR LEFT)	moves the cursor one character position to the left within the same line
	/SHIFT/ /6/ (CURSOR RIGHT)	moves the cursor one character position to the right within the same line
	/S1/ (TAB)	moves the cursor one tab position (four characters) to the right
	/CTRL/ /7/ (REVERSE TAB)	moves the cursor one tab position (four characters) to the left
	/COMMAND/ /4/ (START OF LINE)	moves the cursor to the start of the current line
	/CTRL/ /4/ (END OF LINE)	moves the cursor to the character position immediately following the last non-space character in the current line

to insert text	/CTRL/ /1/ (INSERT MODE)	is entered from overstrike mode. The cursor changes its shape to show that a new mode has been entered. Any character which is subsequently entered is inserted immediately before the cursor position, and the remainder of the text in the line and the cursor are moved one character position to the right. Any character that was in the last character position in the line is discarded. Striking the INSERT MODE key a second time returns the Video File Editor to overstrike mode and the original cursor is restored
	/↲/ (INSERT LINE)	inserts a blank line immediately after the current line and places the cursor at the beginning of that line. Subsequent text is pushed one line down the screen. If the cursor was already on the bottom line of the screen then the window is moved one line down the file and the blank line is inserted on the last line of the window
to delete text	/S2/ or /CTRL/ /H/ (BACKSPACE)	moves the cursor one character position to the left and deletes the character under the cursor. Subsequent characters in the line do not move. The deleted characters are replaced with spaces. This function is usually used for correcting typing errors when entering new text
	/CTRL/ /2/ (DELETE CHAR)	deletes the character under the cursor and shifts the subsequent characters in the line one position to the left

	/CTRL/ /K/ (ERASE TO END)	deletes the contents of the current line from the current cursor position to the end of the line
	/COMMAND/ /2/ (DELETE LINE)	deletes the current line and moves subsequent text one line up the screen. The position of the cursor is not changed, it remains in the same column position. The deleted line of text is placed in a holding area called the restore buffer. This action overwrites the previous contents of the restore buffer except where DELETE LINE functions immediately follow each other, in which case subsequent deleted lines are appended to the buffer. This enables you to move a block of text from the file into the buffer, from where it can be re-inserted into the same or another file using the RESTORE LINES function
to restore text	/CTRL/ /R/ (RECALL LINE)	restores the contents of the current line to its original state. The contents restored are those that existed before the cursor was moved to this line. Once the cursor is moved off a particular line the old contents of that line cannot be recalled using this function
	/COMMAND/ /5/ (RESTORE LINES)	inserts the contents of the restore buffer into the file starting at the the line below the current cursor position. The cursor is moved to the start of the inserted line(s). The restore buffer itself is not changed. This function is used in conjunction with the DELETE LINE function to move and/or copy blocks of text

to split and join lines of text	/COMMAND/ /3/ (SPLIT LINE)	divides the current line into two by moving all text under and to the right of the cursor onto the next line. The cursor does not move. Text on subsequent lines is shifted one line down the screen
	/CTRL/ /3/ (JOIN LINES)	combines two lines into one. The text on the subsequent line is placed immediately after the last non-space character on the current line. The cursor does not move. If the current line cannot accommodate the entire text of the next line then only that amount which fits is moved and the remaining text stays on the same line but is moved to the left hand edge of the screen
to insert a marker	/COMMAND/ /7/	causes a marker to be inserted in the text immediately following the current line. The marker is a line of reverse video spaces containing the text "MARK". If the MARK line was previously located somewhere else in the text it is moved from where it was to the new position. Note that this is not an actual line of text and will never be written to the file. Its placement is therefore only significant during the current editing session. It is used in conjunction with the GOTO MARK function as a place marker (for details see the section entitled "Window Moving Function Keys"), and in conjunction with the high-level command DELETE (see the section entitled "Commands and Searching")

to enter control characters	/SHIFT/ /3/ (ESCAPE)	inserts the Escape ASCII character (ESC, hexadecimal 1B). The Video File Editor allows you to enter only the printable ASCII character set (hexadecimal codes 20 to 7E). To force the generation of "control" codes (hexadecimal 00 to 1F and 7F) the ESCAPE character must be used. When you type the ESCAPE key a special character (a reverse video pound Sterling symbol) is placed on the screen. This is treated like any any other character except that the following character becomes a control character. This means that only the lower five bits of code are written to the file thereby generating a code in the range 00 to 1F. To generate a code of 7F you must enter /ESCAPE/ /?/

Examples

The following table shows some examples of how text can be modified using the functions discussed above.

STEP	IF you enter...	M20 displays...
		The purpose of this text is to act as an example of how to use the editing functions of the Video File Editor
1	DELETE LINE	as an example of how to use the editing functions of the Video File Editor

2	CURSOR UP	as an example of how to use the editing functions of the Video File Editor
3	INSERT LINE INSERT MODE /T/ /h/ /i/ /s/ /SPACE/ /i/ /s/ /SPACE/	This is _ as an example of how to use the editing functions of the Video File Editor
4	JOIN LINES	This is as an example of how to use the editing functions of the Video File Editor
5	DELETE CHAR DELETE CHAR DELETE CHAR	This is an example of how to use the editing functions of the Video File Editor
6	NEXT LINE	This is an example of how to use the editing functions of the Video File Editor
7	DELETE LINE	This is an example of how to use the Video File Editor
8	RESTORE LINES NEXT LINE	This is an example of how to use the Video File Editor the editing functions of
9	INSERT MODE /T/	This is an example of how to use the Video File Editor The editing function of

10	END OF LINE	This is an example of how to use the Video File Editor The editing functions of_
11	BACKSPACE BACKSPACE	This is an example of how to use the Video File Editor The editing functions _
12	RECALL LINE	This is an example of how to use the Video File Editor the editing functions of
13	SPLIT LINE	This is an example of how to use the Video File Editor the editing functions _ of
14	CURSOR UP	This is an example of how to use the Video File Editor _ the editing functions of
15	INSERT LINE	This an example of how to use the Video File Editor the editing functions of

Note: To delete a character in the 80th column you should move the cursor to that position in overstrike mode and enter /SPACE/.

WINDOW MOVING FUNCTION KEYS

The function keys described in the following table enable you to move the window up and down the file.

FUNCTION KEY	MEANING
/COMMAND/ /^/ (TOP)	moves the window to the top of the text file. The cursor is placed on the top bar of the file
/CTRL/ /^/ (BOTTOM)	moves the window to the end of the file. The cursor is placed on the last line of text
/COMMAND/ /-/ (FULL SCREEN UP)	causes the window to be moved up the file by 20 lines. This allows one line of overlap between the old and new displays. The cursor remains on the same screen line
/CTRL/ /-/ (FULL SCREEN DOWN)	causes the window to be moved 20 lines down the file. This allows one line of overlap between the old and new displays. The cursor remains on the same screen line
/COMMAND/ /0/ (HALF SCREEN UP)	causes the window to be moved half a screen (10 lines) up the file. The cursor remains on the same screen line
/CTRL/ /0/ (HALF SCREEN DOWN)	causes the window to be moved half a screen (10 lines) down the file. The cursor remains on the same screen line
/COMMAND/ /9/ (LINE UP)	causes the window to be moved one line up the file. The cursor remains on the same screen line
/CTRL/ /9/ (LINE DOWN)	causes the window to be moved one line down the file. The cursor remains on the same screen line

/SHIFT/ /1/ (NEXT LINE)	moves the window one line down the file and places the cursor at the start of the next text line
/SHIFT/ /5/ (GO TO MARK)	moves the window up or down the file such that the cursor lies on the MARK line. The cursor remains on the same screen line

EXITING AND SAVING FUNCTION KEYS

The function keys described in the following table enable you to exit from the Video File Editor and/or save the file you have been working on.

FUNCTION KEY	MEANING
EXIT AND SAVE	causes the revised text to be written back to the file and the Video File Editor to be terminated. The screen is erased and control is returned to PCOS
SAVE TEXT	causes the revised text to be written to the file. The Video File Editor does not terminate
ABORT	causes the Video File Editor to terminate without writing the revised text to the file. If text has been altered or added since starting the editor you are asked to "Abort?". Strike the ABORT key again to confirm. Any other action causes the Video File Editor to ignore the ABORT. Control is returned to PCOS

COMMANDS AND SEARCHING

The second line of the screen (above the scale line) is called the editor command line and is used for entering high level commands and search strings.

To enter text on the editor command line you must first press the COMMAND MODE function key. This moves the cursor to the second line. You can now enter text there. All line editing operations - such as INSERT MODE, BACKSPACE and DELETE CHAR - now apply to the editor command line. The RECALL LINE function when used in command mode restores the editor command line to its previous contents. The /↵/ key functions instead as EXECUTE COMMAND when used in this mode.

Repeating the COMMAND MODE key returns the cursor to the text window without performing any command operation. The RECALL LINE function, when used in command mode, restores the command line to its previous contents.

STRING SEARCHES

This feature enables you to search the file for a particular combination of characters. Before searching for a text string you must enter command mode by striking the COMMAND MODE function key. Then type in the text to be searched for followed by the appropriate function key, as described in the following table:

FUNCTION KEY	MEANING
/CTRL/ /8/ (SEARCH DOWN)	searches for the text string starting from the the current cursor position and moving down the file until the first occurence of the string is encountered. If found, the window and cursor are moved to it
/COMMAND/ /8/ (SEARCH UP)	searches for the text string starting from the cursor position and moving up the file. If the string is found then the window and cursor are moved to it

Examples

The following table exemplifies the use of the searching functions.

If you enter on the editor command line	Then strike function key...	M20 displays...
		This is an example of how to use the search function keys of the Video File Editor to find a particular combination of characters
/f/ /u/ /n/ /c/	SEARCH DOWN	This is an example of how to use the search function keys of the Video File Editor to find a particular combination of characters
/e/ /SPACE/ /o/ /f/	SEARCH UP	This is an example of how to use the search function keys of the Video File Editor to find a particular combination of characters

COMMANDS

The Video File Editor commands are a set of special commands that enable you to perform a number of high level functions. Before entering a command you must strike the COMMAND MODE function key. You can then type in the command which is subsequently displayed on the editor command line. To execute the command you must then strike the EXECUTE COMMAND key.

This command enables you to move the window to a specific line number in the file.

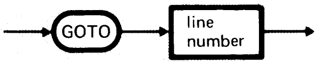


Fig. 12-4 GOTO

Where

SYNTAX ELEMENT	MEANING
line number	a decimal integer and is the desired line number in the file. If this number is greater than the number of lines in the file then the window is moved to the end of the file

Characteristics

Each line of the text file is automatically numbered. That is, the first line of the file is line 1, the TOP bar is line 0 and the MARK bar does not count.

This command removes all text between the current line and the MARK line and places the removed text in the restore buffer from where it can be re-inserted at will. If the MARK line does not exist an error message is given.

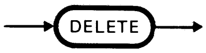


Fig. 12-5 DELETE

The FILE command allows you to suspend processing of the current file and invoke the editor on another file. When editing of the new file is terminated by a SAVE AND EXIT or ABORT function, the old file is recalled at the point at which it was exited.

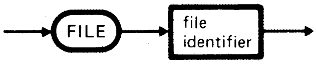


Fig. 12-6 FILE

Where

SYNTAX ELEMENT	MEANING
file identifier	the name of the new file to be edited including any necessary password or volume identifier

Characteristics

The command line option flags (b, c, g or r) used by the old file remain the same for the new file.

Editing of each file is kept entirely independent except for the restore buffer, which enables the transfer of lines of text from one file to another.

Further files can be entered and edited from the new file using the FILE command. There is no limit to the number of levels that can be created in this way except that the text of all the files invoked must fit into user memory.

Files cannot be called recursively.

PART II

13. PCOS COMMANDS

ABOUT THIS CHAPTER

This chapter provides a reference of all PCOS commands. Each command is described in terms of its purpose, a syntax diagram, a description of each syntax element, characteristics of the command, and examples.

CONTENTS