

HANDBOOK FOR DISK USERS

New Brain

ADDITIONAL INFORMATION


Any information on the system which could not be included in the *Handbook for Disk Users* will be found in a file on the system disk called 'INFO'.

To read the information, enter CP/M, type:

TYPE INFO

and press NEWLINE

As the text is shown on the screen, hold down **CONTROL** and press **S** to stop and restart the display.



New Brain

HANDBOOK FOR DISK USERS

How to use this Manual

If you just want
to use prepared
programs on your
NewBrain, and don't
want to know about
how computers work . . .

READ
SECTIONS 1, 2, 3

LOOK THROUGH
SECTIONS 10-12

If you know some
BASIC already and
want to develop your
ability by using
disk techniques . . .

WORK THROUGH THE
WHOLE OF THIS
MANUAL AS A BASIS
FOR FURTHER STUDY
OF CP/M METHODS

If you know little
of computers or
BASIC, but wish
to become
proficient . . .

READ SECTIONS 1, 2, 3

LOOK THROUGH
SECTIONS 10-12

WORK THROUGH THE
BEGINNER'S GUIDE

WORK THROUGH THE
REST OF THIS MANUAL

CONTENTS

1	Introduction	3
2	Connecting up	9
3	Getting started	13
4	NewBrain BASIC with disk—Programs	25
5	NewBrain BASIC with disk—Data Files	31
6	NewBrain BASIC with disk—Random Access Files	41
7	Cataloguing files through BASIC	51
8	Using cassettes	53
9	What is CP/M?	55
10	Working with disks	59
11	Using CP/M with NewBrain	78
12	Error codes	82
13	The Expansion Interface Module	84
	Index	90

1

Introduction

This manual describes how to use the NewBrain microcomputer model A or AD with disk memory.

Computers carry out the tasks we give them by following sets of instructions called programs. These may range from a simple program to calculate the wall area of a room to a very large and complex program for word-processing or keeping accounts.

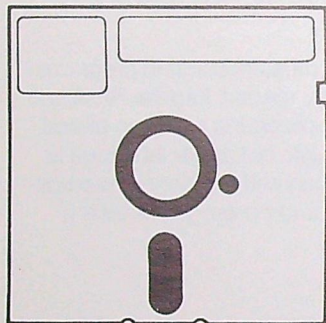
When using a program the computer stores it in a temporary memory called RAM (random access memory). The RAM can only hold a program while the computer is switched on. As soon as you switch off, the program disappears, which makes it possible to use many different programs in the same computer, and so to use the same computer for many different tasks.

It is necessary, therefore, to have some means of storing programs permanently, so that any program can be 'loaded' into the RAM and used at any time. With microcomputers, programs are often stored on cassette tape, which is effective enough, but rather laborious to use: tapes have continually to be re-wound and positioned; records have to be kept of where programs are; and the programs take a relatively long time to load.

1. Introduction

Disks on the other hand are easy to use; they can store many programs and load them into the computer quickly. The disks used by NewBrain are 5.25in. 'floppy' disks, made of mylar coated with a magnetic oxide, rather like recording tape. Each disk is permanently enclosed in a black plastic cover.

Disks are used in disk drives. The central hole is for the spindle of the drive, which spins the disk at high speed inside its plastic cover.

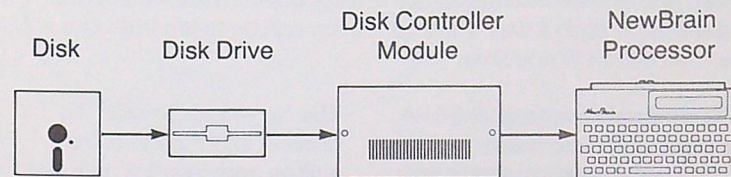


The small hole is an index mark which enables the disk drive to reference the disk correctly.

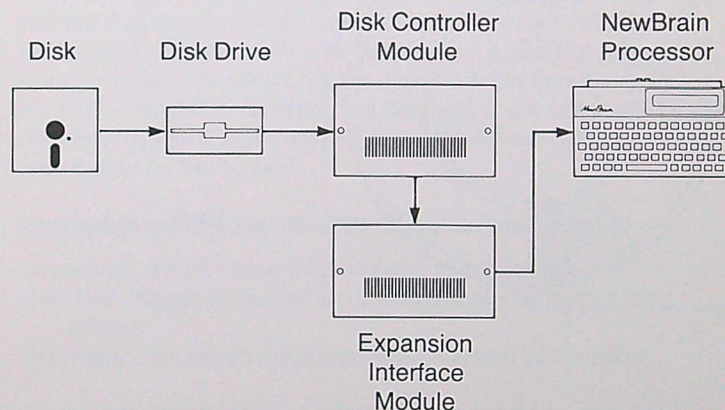
When the disk is used in the disk drive, its surface is scanned by the 'read-write head' through the long slot in the plastic cover. The whole operation is rather like that of a record player, except that the read-write head works by magnetism.

1. Introduction

The disk drive is linked to the computer by the Disk Controller, which interprets between the two and controls the operation of the disk drive.



You may also fit an Expansion Interface Module into the system to provide additional memory, which is needed for many business programs, and to enable you to link your NewBrain with other computers and equipment.



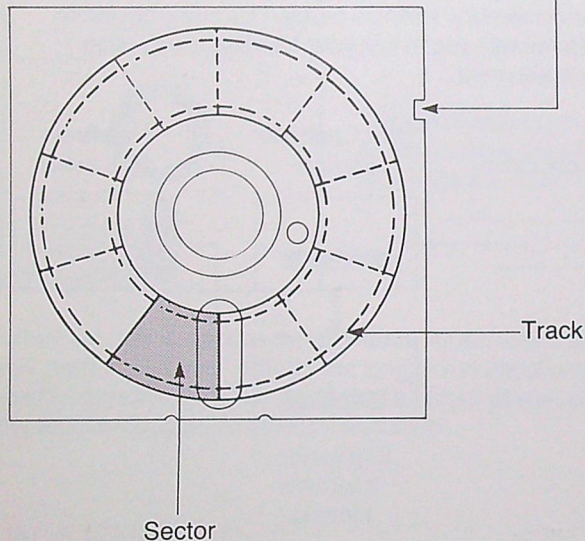
1. Introduction

How disk memory works

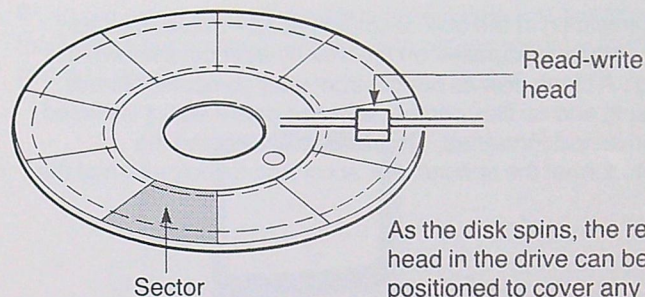
Earlier in this section we likened disk drives to record players, but there is an important difference in the way the information on disks is organised. With a sound recording, the information is in serial order, just like the information on a tape. But computer disks are organised in such a way that information can be taken from any part of the disk at any time.

When prepared for use, the disk is divided into tracks, each of which is divided into sectors.

The 'write-enable slot'. To prevent information being written onto the disk, this slot can be covered with an aluminised label.



1. Introduction



As the disk spins, the read-write head in the drive can be positioned to cover any sector of any track, and 'read' information from it or 'write' information onto it.

When a program is recorded onto a disk, it may be spread over several sectors on the same track, or even over more than one track. The computer needs to be able to find this information again and put it all together in the same order. To make this possible, each disk is provided with a 'directory' which, like the programs, is magnetically encoded onto the disk, and updated every time a program is added or deleted. The directory is not only used by the computer system; it also serves as a contents list which the user can display on the screen.

Information recorded on disks can be of several kinds:

- programs, which cause the computer to carry out tasks;
- text files containing text which will eventually be output to a printer;
- data files from which the computer can select information.

The general name for any block of data or program on a disk is a 'file'.

1. Introduction

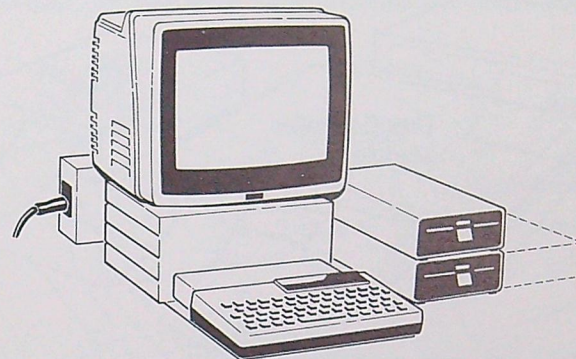
Formatting

All the organisation of the disk described so far—tracks, sectors and the directory—is imposed on the disk by a process known as ‘formatting’. A blank disk as bought from your computer store is literally blank, and no files can be recorded onto it until it is placed in a disk drive and formatted. This is done by causing the computer to format the second disk according to the pattern of the first.

2

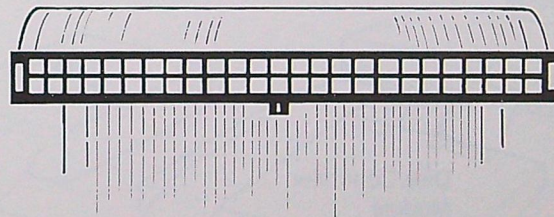
Connecting up

If you are using a NewBrain Module Housing, first install your modules in the housing, with the Expansion Interface Module (if any) lowest.



Before connecting the modules together, check the ribbon cables and connectors.

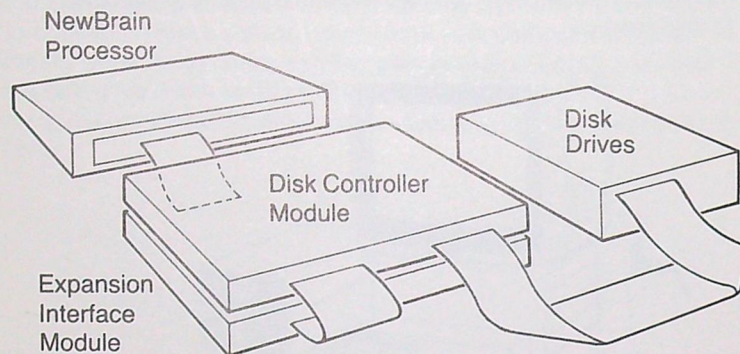
Note that the connectors have a small key notch at the bottom, matching a similar shape in the socket.



2. Connecting up

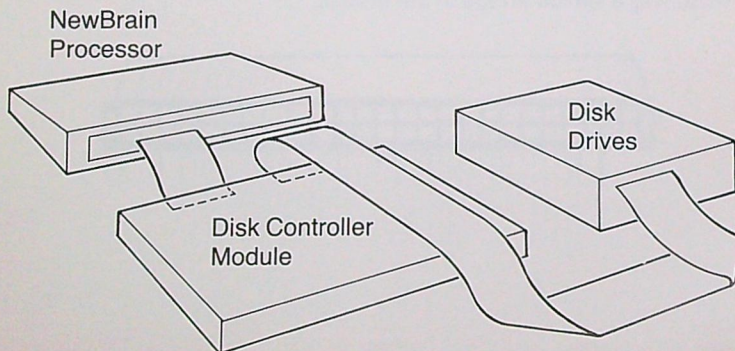
With Expansion Interface Module

Connect the ribbon cables as shown below.



Without Expansion Interface Module

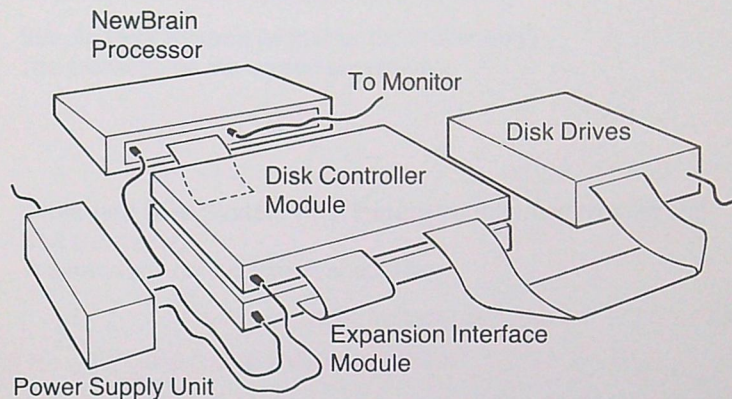
Connect the ribbon cables as shown below. Note that the Disk Controller Module is the other way up compared with the illustration above.



2. Connecting up

Power Supply Unit and Monitor

The Power Supply Unit should be connected as shown. Either a monitor or a television set may be used, connecting with the appropriate lead to the marked point at the back of the NewBrain processor. The disk drive has an independent power supply.



3

Getting Started

When you have fully connected up the system, plug into the mains and switch on. There is no separate switch for the NewBrain processor or modules, so these will start to operate immediately, but remember to switch on the disk drive(s) and monitor.

What happens next depends on your system.

Simple Disk System (with Disk Controller only)

The message on the screen should be:

NEWBRAIN BASIC
READY



Expanded Disk System (with Expansion Interface Module and Disk Controller)

The message on the screen should be:

NewBrain paged system main menu

Move cursor and press newline



BASIC
CP/M 2.2

If nothing happens or if you do not get the right message, check all the connections against the diagram in the last section and make sure all plugs are pushed securely home.

This starting display, incidentally, is called a **menu**, because it offers you a choice. Many computer applications are controlled by means of menus.

3. Getting Started

Disk Drives

You may be using a single disk drive, a twin drive holding two disks, or an even larger system. One of the drives is selected as Drive A, and this is the one that is always used unless you instruct the computer to use another one. When you have only one disk drive, that drive is Drive A.

Looking after your disks

Now to insert a disk, but before you do so, please read the following.

Disks should be handled with great care. Remember that successful operation depends on a piece of plastic rotating at high speed inside an envelope, so any physical damage is likely to increase friction and may render the disk useless.

Dust and grease on the magnetic surface can also be damaging. The read-write head of the disk drive actually touches the disk surface, and even a speck of dust can impair performance and cause reading errors. So always keep disks in their paper covers when not in use and never touch the magnetic surface.

Similarly magnetic fields must be avoided. Never lay a disk on a television set, a tape recorder or a telephone, for example.

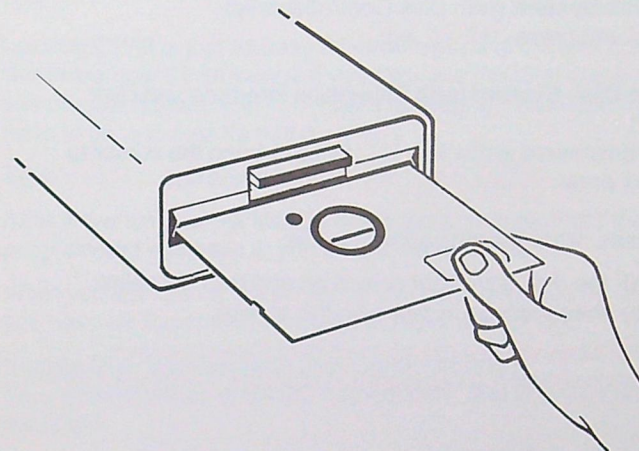
Ideally disks should be kept in a box designed for the purpose. It all adds up to common sense: if disks are handled sensibly they will give good service over a long period.

Of course, occasionally a disaster occurs: you leave a disk all night on a storage heater (after which it is slightly smaller and quite useless!) or you shut it in the latch of your briefcase. The only defence is to make back-up copies of your most valuable disks, and this will be dealt with in a later section.

3. Getting Started

Inserting disks

Lift the flap at the front of the disk drive. Hold the disk label upward, with the thumb of your right hand on the label. Push the disk gently into the drive and close the flap again.



If you have a single disk drive, insert your system disk into it. The system disk is the one provided with the disk drive and documentation, and which contains the information needed to make your disk system work correctly. If you have a dual disk drive, insert the system disk into the left-hand (or topmost) drive (Drive A).

3. Getting Started

CP/M

How you get into CP/M depends on whether you have an Expansion Interface Module or not.

Simple Disk System (with Disk Controller only)

Type **CPM** and press **NEWLINE**

Expanded Disk System (with Expansion Interface and Disk Controller)

Press the downward arrow key  twice to bring the cursor to CP/M; then press **NEWLINE**

In both cases, what should happen now is:

- a) the disk drive light comes on and the disk spins;
- b) the message appears on the screen:

NewBrain CP/M Version 2.2 Copyright © 1979 Digital Research
Implementation version 1.0 Copyright © 1983 Grundy Business Systems
A>

- c) the disk stops.

Now type **DIR** and press **NEWLINE** (You may type in upper or lower case (dir) as you wish; the system recognises either in commands.)

The disk will spin again and the next few lines of the screen will be covered with cryptic words. 'DIR' of course means 'directory' and the words on the screen are the names of all the files on the disk.

Many of these files are in fact programs which control the computer for you. Control is what CP/M is all about: the name means 'Control Program for Microprocessors' and CP/M's function is to manage all the detailed running of the computer so that you have only to give it instructions. We have already given it one instruction: DIR. We shall go into others later.

3. Getting Started

CP/M is a very popular system among the designers of the programs for microcomputers, and many packages for use in business, research and industry are available on CP/M-based disks.

Leaving CP/M is just as easy as entering it, and the process illustrates how CP/M works. If you examine the directory you will see one file named 'EXIT'. To call this program into action, all you have to do is to type its name.

Type **EXIT** and press **NEWLINE**

After a few turns of the disk, in which the computer finds the program and executes it, you will be out of CP/M.

What you will see on the screen at this point depends on whether you have an Expansion Interface Module fitted or not.


Simple Disk System (with Disk Controller only)

Your system will be in BASIC immediately, and you will see the message:

NEWBRAIN BASIC
READY



Expanded Disk System (with Expansion Interface and Disk Controller)

Your system will return to the starting menu. Press the downward arrow key  once and press **NEWLINE** You will see the message:

NEWBRAIN BASIC
READY



3. Getting Started

BASIC

This manual is not the place to go into what BASIC is and how it works. BASIC is the most popular language in which programs for microcomputers are written, and if you want to learn how to use it the best advice is to buy the NewBrain **Beginner's Guide**.

But many published packages for education and commerce are written in BASIC, and if you want to use these you will need to know how to load them from the disk into the computer and how to start them operating.

To load programs, type **LOAD** followed by the name of the program in inverted commas. To illustrate the next few commands, you should load a program called INDEX, which is on your system disk. To do this, insert the system disk. Then, with the system in BASIC, type

LOAD "INDEX" and press **NEWLINE**

The cursor will disappear, the disk light will come on and the disk will spin for a few seconds. Then the cursor will reappear on the screen and the disk will stop. The program is now in the computer's memory, but it will not work until you tell it to start. To start the program, type **RUN** and press **NEWLINE**

Take a few minutes off to try the program. It allows you to input ten index items with page numbers and then re-arranges them in alphabetical order. We shall look at how it all works later. Work through the whole program so that it comes to an end and returns the control of the computer to the keyboard.

If you do not want to work all the way through the program, you can stop it by pressing the **RESET** button on your **Disk Controller**. But remember that it is a total **RESET**, the equivalent of switching the system off and on again, so all programs and data in memory are lost.

3. Getting Started

Leaving BASIC

Before leaving BASIC you should decide whether you wish to keep the program or data in memory. Once you leave BASIC they are lost.

If you are using a simple disk system (with Disk Controller only):

- place a CP/M disk in the drive you are using;
- type **CPM** and press **NEWLINE**

and the system will transfer immediately to CP/M.

If you are using an expanded disk system, type **EXIT** and press **NEWLINE**

The system will return to the starting menu.

Removing disks from the drive

Just lift the flap at the front of the drive and the disk will spring out. If it fails to do so, try again. (Never insert anything but disks into a disk drive!)

PROTECT YOUR SYSTEM!

Before you go any further with your NewBrain disk system, you should make a copy of your system disk.

If you damage it in any way, you could be effectively locked out of your system until you get another system disk.

There are two procedures, depending on whether you have two disk drives or more; or whether you have a single disk drive. These are set out below.

General procedures for formatting and copying disks are set out in Section 10.

Do it now—or you may regret it!

3. Getting Started

TO FORMAT DISKS USING TWO DRIVES OR MORE

1. Make sure the System Master disk is write-protected.
2. Place the System Master disk in drive A.
3. Type **FORMAT**
Press **NEWLINE**
REPLY: NewBrain disc format program Version 1.0
Copyright © 1983 Grundy Business Systems
Please enter letter of drive on which disc is
to be formatted or press newline to exit program
4. Type **B** (or whatever drive you are using)
REPLY: Place blank disk in drive B: and press newline
5. Place the blank disk in B and press **NEWLINE**
The disk spins and 'ticks' for about a minute.
DO NOTHING!
REPLY: Please enter letter of drive on which disc is
to be formatted or press newline to exit program
6. Press **NEWLINE**
and the prompt reappears: **A>**
7. Type **SYSGEN**
Press **NEWLINE**
REPLY: NewBrain CP/M System Generation program Version
1.0 Copyright © 1983 Grundy Business Systems
Source drive?
Type **A**
Press **NEWLINE**
REPLY: Destination drive?
8. Type **B** (or whatever drive you are using)
Press **NEWLINE**

3. Getting Started

9. The disk spins for a few seconds.

REPLY: Destination drive?

Press **NEWLINE**

and the prompt reappears: **A>**

You are ready to copy all programs from the system disk to the copy disk.

10. Type **PIP B:=*.***
Press **NEWLINE**

and all programs on the system disk will be transferred to the copy disk.

TO FORMAT DISKS USING ONE DRIVE ONLY

1. Make sure the System Master disk is write-protected.
2. Place the System Master disk in the drive.
3. Type **FORMAT**
Press **NEWLINE**
REPLY: NewBrain disc format program Version 1.0
Copyright © 1983 Grundy Business Systems
Please enter letter of drive on which disc is
to be formatted or press newline to exit program
4. Type **A**
REPLY: Place blank disk in drive A: and press newline
5. Place the blank disk in the drive and press **NEWLINE**
The disk spins and 'ticks' for about a minute.
DO NOTHING!
REPLY: Please enter letter of drive on which disk is
to be formatted or press newline to exit program

3. Getting Started

6. Press **NEWLINE**
and the prompt reappears: **A>**
Remove the disk you have formatted
from the drive.
7. Place the System Master disk in the
drive.
Type **SYSGEN**
Press **NEWLINE**
REPLY: **NewBrain CP/M System**
Generation program Version 1.0
Copyright © 1983 Grundy Business Systems
Source drive?
Type **A**
REPLY: **Destination drive?**
8. Remove the System Master and insert the disk on
which you wish to place the system information.
Type **A**
Press **NEWLINE**
9. The disk spins for a few seconds.
REPLY: **Destination drive?**
Press **NEWLINE**
and the prompt reappears: **A>**

You are now ready to copy all programs from the system
disk to the copy disk.

3. Getting Started

10. Place the System Master disk in the drive
Type **SCOPY**
Press **NEWLINE**
REPLY:
NewBrain disc copy (single drive) program Version 1.0
Copyright © 1983 Grundy Business Systems

Press newline to perform disk copy or something else
to return to system
11. Press **NEWLINE**
REPLY: **Place source disk in drive A: and press a key**

Place the disk you wish to copy in the drive.
Press any key

and the disk will be read.
12. REPLY: **Place destination disk in drive A: and press a key**

Place the disk which is to receive the copies in the drive.
Press any key

and information will be written onto the disk.

This process is repeated until
REPLY: **Type C to continue with another copy operation**
or newline to return to the system

3. Getting Started

SUMMARY

TO ENTER CP/M OR BASIC

Simple disk system (with Disk Controller only)

To enter CP/M from BASIC: type **CPM** and press **NEWLINE**

To enter CP/M at start-up:

- before switching on, place the disk in drive A;
- switch on; hold down **STOP** until CP/M appears

To enter BASIC from CP/M:

- place a disk containing the EXIT program in the current drive
- type **EXIT** and press **NEWLINE**

Expanded disk system (with Disk Controller and Expansion Interface)

To enter CP/M or BASIC:

- select CP/M or BASIC from the starting menu (which appears automatically when the system is switched on)

To return to the starting menu from BASIC:

- type **EXIT** and press **NEWLINE**

To return to the starting menu from CP/M:

- place a disk containing the EXIT program in the current drive;
- type **EXIT** and press **NEWLINE**

WARNING: on leaving BASIC any program or data in memory is lost.

For a list of files on a disk, enter CP/M. Then type **DIR** and press **NEWLINE**.

To use a program in CP/M, type its name and press **NEWLINE**

To use a program in BASIC, first load it by typing **LOAD** “program name” and press **NEWLINE** Then type **RUN** and press **NEWLINE**

4

NewBrain BASIC with Disk – Programs

In this and the following section, we shall load a program, merge it with another, save the combined program on a disk, and use that program to demonstrate disk techniques.

LOADING

Load the program **INDEX** from the system disk in the manner described in the last section, i.e. get into BASIC, type

LOAD “INDEX”

and press **NEWLINE**

MERGING

In order to work through the examples in this section, you will need to extend the **INDEX** program by adding to it the lines which make up another program, **INDEX2**. The function of these extra lines is to record the index which is sorted by the program onto disk.

First list the program by typing **LIST** and pressing **NEWLINE**. You will see that the last lines are:

```
1170 PUT22, 27, I+6: PRINT A(I)
1180 NEXT I
1190 END
```

If we were to type **LOAD “INDEX2”** at this point, we should get just that: **INDEX2** alone, since the command **LOAD** wipes out the program in memory before loading another.

The command to use is **MERGE**, which loads a new program without wiping the old one, except where a new line has the same number as one in the old program. In such a case, the new line replaces the old one.

4. NewBrain BASIC with Disk -- Programs

Type

MERGE "INDEX2" and press **NEWLINE**

(Make sure you do not leave a space between the 'INDEX' and the '2', or the computer will think you have finished the file name and load INDEX again.)

Now check line 1190 which previously contained only the word 'END'. This has now been overwritten by

1190 PUT 22,1,18

and the program extends to line 2150.

4. NewBrain BASIC with Disk -- Programs

SAVING PROGRAMS--FILE NAMES

To record a program onto the disk, use the command SAVE, followed by the program name in inverted commas. When you have done this and pressed NEWLINE, the program is recorded as a file on the disk.

Be careful when choosing file names. The rules are:

1. Names must be of 8 characters or fewer. If you try to give a file a longer name, the system will select the first 8 characters and forget the rest.
2. You may use upper or lower-case, as you wish. The system will use upper case only in the file directory.
3. You may not use the symbols:

<>.,,:=★?[]

in file names.

4. If you save a file with the same name as another file already on the disk, the file already there will be over-written by the new file. This is useful when you want to save an improved version of the same program, but can be a trap if you are careless in naming your files.
5. The system interprets a space as the end of a file name. This can also be a trap. For example, INDEX and INDEX2 are two different files, but if you tried to save INDEX2 onto a disk by entering

SAVE "INDEX 2"

the system would end the file name at the space and make a file called INDEX--and if a file of that name already existed on the disk it would be over-written.

Since we now have a combined program in the computer which both sorts index items and records them onto the disk, it would be convenient to save it under a name of its own. Type

SAVE "INDEX3" and press **NEWLINE**

4. NewBrain BASIC with Disk – Programs

THE DISK DIRECTORY

On the command SAVE the disk spins and the light comes on, but what indication do we have that the file is actually on the disk?

To get a list of files on a disk, first select CP/M. If you are following this explanation you will still be in BASIC.

If you have a simple disk system (Disk Controller only) type

CPM

and press **NEWLINE** This gets you back to CP/M directly.

If you have an expanded disk system (with the Expansion Interface as well), type

EXIT

and press **NEWLINE** This gets you back to the starting menu, from which you can select CP/M.

The disk spins and the prompt appears:

A>

This means that the system is directed to disk drive A, and that it is ready to receive a command. Type

DIR and press **NEWLINE**

Search among the file names (which are explained in detail in the sections on CP/M) for INDEX3. You should also find INDEX4, which we shall need later.

To return to the starting menu, type **EXIT** again and press

NEWLINE

4. NewBrain BASIC with Disk – Programs

PROTECTING FILES

We mentioned earlier the danger of over-writing files by accident. It is possible to protect files against this by converting them into 'read-only' files. The files INDEX, INDEX2 and INDEX4 are protected in this way, so that if you attempted to save a program with one of these names the system would leave the file unaffected and display an error message:

Bdos Err on A: File R/O

which means that the system has found an error on Drive A. R/O means 'read only'.

The method of protecting files involves a CP/M command, which we shall go into in more detail later. As an example, to protect INDEX3, the method is to get into CP/M and type

STAT INDEX3 \$R/O and press **NEWLINE**

To remove the protection, type

STAT INDEX3 \$R/W and press **NEWLINE**

4. NewBrain BASIC with Disk -- Programs

SUMMARY

Loading: `LOAD "FILENAME" NEWLINE`

Saving: `SAVE "FILENAME" NEWLINE`
(SAVE over-writes another file of the same name)
For rules relating to file names see page 24.

Merging: `MERGE "FILENAME" NEWLINE`
MERGE is equivalent to LOAD, except that any program in memory is not wiped.

Disk Directory: put the system into CP/M and type

`DIR NEWLINE`

Protecting files: put the system into CP/M and type

`STAT FILENAME $R/O NEWLINE`

To remove protection, type

`STAT FILENAME $R/W NEWLINE`

5

NewBrain BASIC with Disk --Data Files

We are now in a position to use INDEX3 (i.e. INDEX merged with INDEX2) and see how it works.

First load it and list it as described in previous sections.

In principle what happens is that the data to be recorded is loaded into two arrays, and the arrays are recorded as a file on the disk using Device 12, which is the NewBrain facility for recording data in this way.

In order to use Device 12 (or indeed any device) a BASIC statement must be executed opening a stream to that device. A stream is a route through which data can flow, and streams must be given a number from 0 to 255. After a specific stream has been opened to a specific device, any data passed through that stream goes to the device.

Once the system has been set up in this way, data can be passed to the device quite simply by using the command PRINT. As described in the rest of this section, data can also be retrieved from disk files using Device 12.

For more details of streams and devices, see the **NewBrain Handbook**.

5. NewBrain BASIC with Disk – Data Files

```

10 REM INDEX WRITER
20 DIM A$(10),A(10)
30 PUT 31,23,66
40 PRINT: PRINT "This program accepts ten index words"
50 PRINT: PRINT "with their page numbers and arranges"
60 PRINT: PRINT "them in alphabetical order."
70 PUT 22,1,20: PRINT "Press 'NEWLINE'"
80 INPUT ("" ) S$
500 REM INPUT WORDS
510 PUT 31: PRINT
520 PRINT "Type words in any order,"
530 PRINT "pressing 'NEWLINE' after each."
540 PRINT "(Not more than 20 letters in each word)"
550 PUT 22,1,7
560 FOR I=1 TO 10
570 INPUT ("" ) A$(I)
580 A$(I)=LEFT$(A$(I),20)
590 IF LEFT$(A$(I),1)>="A" AND LEFT$(A$(I),1)<="Z" THEN 610
600 A$(I)=CHR$(ASC(LEFT$(A$(I),1))-32)+MID$(A$(I),2)

```

- Line 570 accepts input to the array A\$(1) to A\$(10).
- Line 580 limits the content of each array element to 20 characters.
- Lines 590 and 600 change the first character of each item to a capital letter if it is not already a capital letter.

5. NewBrain BASIC with Disk – Data Files

```

610 PUT 22,1,22.
620 PRINT "Type page number and press 'NEWLINE'"
630 PUT 22,27,I+6:PRINT " "
640 PUT 22,27,I+6
650 INPUT ("" ) S$: IF NOT NUM(S$) THEN 630
660 IF VAL(S$) <1 OR VAL(S$) >9999 THEN 630
670 A(I) = VAL(S$)
680 PUT 22,1,I+7
690 NEXT I
1000 REM ALPHABETISE
1010 PUT 31: PRINT
1020 PRINT "Here is your index in alphabetical order"
1030 PRINT: PRINT "-----"
1040 J = 10
1050 FOR S=1 TO J-1
1060 Q=0
1070 FOR I=1 TO J-S
1080 IF A$(I) <=A$(I+1) THEN 1120
1090 X$=A$(I): A$(I)=A$(I+1): A$(I+1)=X$
1100 X=A(I): A(I)=A(I+1): A(I+1)=X
1110 Q=1
1120 NEXT I
1130 IF Q=1 THEN 1050
1140 NEXT S

```

Lines 610 to 690 accept input to the array A(1) to A(10) for page numbers. The PUT codes position the cursor so that the display is easier to use.

Lines 1000 to 1140 are a 'ripple sort' routine in which pairs of items are successively compared for alphabetical order and swapped over if necessary, the process being repeated until the items are all in alphabetical order.

5. NewBrain BASIC with Disk – Data Files

```
1150 FOR I=1 TO 10
1160 PUT 22,1,I+6: PRINT A$(I)
1170 PUT 22,27,I+6: PRINT A(I)
1180 NEXT I
1190 PUT 22,1,18
1200 PRINT "-----"
1210 PRINT "For another index, press 'I' and NEWLINE"
1220 PRINT "To record on disk,press 'R' and NEWLINE"
1230 INPUT ("" ) S$
1240 IF S$="I" OR S$="i" THEN RUN
1250 IF S$="R" OR S$="r" THEN 2000
1260 GOTO 1190
2000 REM RECORD FILE ON DISK
2010 PUT 31,22,1,5
2020 PRINT "Choose a name for your index file"
2030 PRINT: PRINT "(Not more than 8 letters; no spaces)"
2040 PRINT: PRINT "Type the name and press NEWLINE"
2050 PRINT: INPUT ("" ) F$
2060 PUT 31,22,1,5
```

Lines 1150 to 1260 display the index on the screen in alphabetical order, and offer the user a choice.

Lines 2000 to 2060 ask the user to choose a name for the index file to be recorded on the disk.

5. NewBrain BASIC with Disk – Data Files

```
2070 OPEN OUT#12,12,F$
2080 FOR I=1 TO 10
2090 PRINT#12,A$(I)
2100 PRINT#12,A(I)
2110 NEXT I
2120 CLOSE#12
2130 PRINT "FILE: ";F$: PRINT
2140 PRINT "Recorded on disk"
2150 END
```

Line 2070 opens stream 12 to Device 12. The word 'OUT' tells the computer the direction of data flow.

Device 12 is the facility to record data in the form of variables or arrays onto a disk. From line 2070 onwards, until stream 12 is closed, any variable sent along the stream will go to the disk and be recorded. The choice of 12 for the stream number is arbitrary. It could be any number from 0 to 255, and does not have to be the same number as the device.

The actual recording onto the disk is done in lines 2090 and 2100 where the array elements are 'printed' one by one onto the disk.

Line 2120 closes the stream. It is important to close the stream since data does not go directly to the disk, but is made up into batches in a buffer store. The command CLOSE empties the buffer and sends the last batch out to the disk.

5. NewBrain BASIC with Disk – Data Files

Now run the program, enter some words and numbers, and record the alphabetised index on disk. Make a note of the file name since you will need it later.

Having got data recorded in a disk file, we need a procedure to get it back again. In a full-scale program for commercial use, data is continually written onto files, retrieved from them, added to and re-processed, and recorded back onto the files again.

The program INDEX4 is an illustration of the method for retrieving data from files using Device 12. Load it and run it, and if you enter the right file name you will see all your data displayed on the screen again.

5. NewBrain BASIC with Disk – Data Files

```
3000 REM RETRIEVE INDEX FILE
3010 PUT 31,23,66,22,1,5
3020 PRINT "Type the name of your file"
3030 PRINT: PRINT "and press NEWLINE":PRINT
3040 INPUT("")F$
3050 PUT 31,22,1,2
3060 PRINT "INDEX FROM FILE ";F$
3070 PRINT "-----": PRINT
3080 OPEN IN#12,12,F$
3090 FOR I=1 TO 10
3100 INPUT#12,A$(I)
3110 INPUT#12,A(I)
3120 PRINT TAB(1);A$(I);TAB(27);A(I)
3130 NEXT I
3140 CLOSE#12
3150 END
```

Line 3080 opens Device 12 (using stream 12 again, although any stream number could be used) for data to flow into the computer. The INPUT commands in lines 3100 and 3110 are just like the keyboard INPUT, except that the data comes in through stream 12 instead. The effect is the same, however: the arrays are filled and their contents are displayed by line 3120. Again the stream should be closed (line 3140).

DEVICE 13

The previous example uses device 12, which is the device you would normally expect to use for data files on disk.

Device 13 does much the same job as device 12 and can be used in the same way. The difference is that it is geared to text processing. Not only will it display the contents of files on the screen through the same procedures as device 12. It will also adapt the files for printing out by inserting a line feed after every NEWLINE.

You can test the effect of this even if you do not have a printer available. Change INDEX3 (i.e. INDEX2 merged with INDEX) so that it writes to device 13 instead of device 12. Only one line need change:

```
2070 OPEN#12,13,F$
```

Then make another index and save it under a different file name from the one you chose for the stream 12 version.

When you have done this, change the system to CP/M by typing

```
EXIT [NEWLINE]
```

Now try this:

```
TYPE filename used with device 12 [NEWLINE]
```

```
TYPE filename used with device 13 [NEWLINE]
```

You should be able to see very little of the file you wrote with device 12 at all, since it contains no line feeds and will all be crowded onto a single line, each successive item overwriting the previous one.

The file you wrote with device 13, however, should be spread down the screen, item by item, although of course there will be no tabbing.

SUMMARY**To save data in a disk file**

1. Load the data into variables or arrays.
2. Open a stream to Device 12. The stream number can be any you choose, unless you are already using it for something else. For example

```
OPEN OUT#50,12,"DEBTORS"
```

opens stream 50 to create a data file called "DEBTORS".

You may also use a variable for the file name, allowing the user to name the file, as in the INDEX program.

3. Record the variables or arrays on the disk. For example

```
PRINT#50,A$      PRINT#50,Z
PRINT#50,X$(2)   PRINT#50,P(3)
```

Always use a separate PRINT statement for each variable or array.

4. Close the stream. For example

```
CLOSE#50
```

This empties the buffer and so completes the writing of data onto the disk.

5. NewBrain BASIC with Disk – Data Files

To retrieve data from a disk file

1. Open a stream to bring data from Device 12.
For example

```
OPEN IN#25,12,"DEBTORS"
```

2. Input the data into variables or arrays.
For example

```
INPUT#25,A$      INPUT#25,Z  
LINPUT#25,X$(2)  INPUT#25,P(3)
```

LINPUT may be used as an alternative to INPUT with string variables. It avoids the problems which occur with INPUT when strings including commas or quotes are entered. For more details see the **NewBrain Handbook**.

- NOTES. a) Always retrieve variables and arrays in the order in which you recorded them.
- b) Always use a separate INPUT for each variable or array.
3. Close the stream.
For example

```
CLOSE#25
```

6

NewBrain BASIC with Disk –Random Access Files

Once you are used to them, data files as described in the last Section seem a very natural and helpful technique. BASIC programs commonly store information in arrays and variables, so what could be more natural than to store the arrays and variables on a disk?

There are some situations, however, where the sheer volume of data to be stored and accessed is far greater than your computer's RAM will hold. Suppose for example we wish to write a program which will enable medical staff to look up the characteristics of drugs prescribed for patients. Of course such data is already available in books, but a computer could select and classify it in many different ways, and could make it instantly available.

The problem would be the amount of information to be held and accessed. Random access files on disk are ideal for such a purpose. The idea is that instead of holding all the data in RAM, the computer 'looks it up' on the disk itself, positioning the read-write head so as to pick out the relevant data on request.

Before going into the how random access files work, load the program called RAF2 from the disk and run it. RAF2 is a small example of the kind of look-up facility for drugs mentioned above. It offers information on just 6 drugs, but the program could easily be extended to cover hundreds.

6. NewBrain BASIC with Disk – Random Access Files

Notice what happens when you run the program. On giving the command RUN, the disk spins. When you select a drug it usually spins again and the text appears line by line on the screen.

As the disk spins the computer literally looks up the data. This data is coded in bytes, and since a character is expressed as a single byte the program can look up the file literally character by character if necessary. In the present example, the information on each drug is coded into a block of twelve records. So the information on Sectral, for example, is found at a certain number of bytes from the beginning of a file called DRUGS, and the program directs the read-write head of the disk drive to go straight to that point and begin reading it. No matter where the information is stored on the disk, the computer can find it in a matter of seconds.

So the number of drugs in the index and the amount of information on each are no longer limited by the size of the computer's RAM, but only by the capacity of the disk, which can hold many hundreds of items of this size.

WRITING ONTO THE DISK

Random access files use device 14, which is specifically designed to enable a BASIC program to write records onto a disk and read them back again. When using device 14 for random access files, you should note the following.

1. Records on the disk always relate to a named file (e.g. DRUGS in RAF2) and are numbered in bytes from the start of that file.
2. A record can be as small as one byte or as large as 8 megabytes, if the memory capacity of your computer allows this.

6. NewBrain BASIC with Disk – Random Access Files

3. In order to write or read a record, the BASIC program must specify

- which file it is in;
- where it starts in that file;
- how long it is.

4. The length of the record must be observed in all statements. Thus if you tell the computer you are using records of 39 bytes, you must send records of precisely 39 bytes down the appropriate stream to device 14, so that they are recorded on the disk. If your records are not the right length you must adjust them by abbreviating them or by adding spaces.

The method is illustrated in the following example. This is RAF1, the program actually used to create the file DRUGS and place in it the data used by RAF2.

6. NewBrain BASIC with Disk -- Random Access Files

RAF 1

```
10 REM Make random access file
20 REM open file
30 PUT 31
40 PRINT "Type file name and press NEWLINE"
50 INPUT("")(F$
60 OPEN#1,14,F$
70 REM Enter items for file
80 FOR R=0 TO 960 STEP 12
90 PUT 31
100 PRINT "Enter information as shown, pressing"
110 PRINT "NEWLINE after each entry. If no more"
120 PRINT "items, type *** and press NEWLINE": PRINT
130 PRINT "Item ";INT(R/12)+1
140 GOSUB 500
150 NEXT R
160 CLOSE#1
170 PRINT "File: ";F$;" recorded on disk"
180 END
```

Line 60 opens stream 1 to device 14 for file F\$. In the example on the disk F\$ was 'DRUGS'.

6. NewBrain BASIC with Disk -- Random Access Files

```
500 REM Input items
510 LINPUT("Brand name: ")A$
520 IF A$="***" THEN CLOSE#1: PUT 31: END
530 N=R+1: GOSUB 1000
540 LINPUT("Manufacturer: ")A$
550 N=R+2: GOSUB 1000
560 LINPUT("Generic name: ")A$
570 N=R+3: GOSUB 1000
580 LINPUT("Type: ")A$
590 N=R+4: GOSUB 1000
600 PRINT: PRINT "Special precautions: enter four lines,"
610 PRINT "pressing NEWLINE after each, "
620 PRINT "even if blank.":PRINT
630 FOR I=5 TO 8
640 LINPUT A$
650 N=R+I: GOSUB 1000
660 NEXT I
670 PRINT: PRINT "Contraindications: enter four lines,"
680 PRINT "pressing NEWLINE after each,"
690 PRINT "even if blank."
700 FOR I=9 TO 12
710 LINPUT A$
720 N=R+I: GOSUB 1000
730 NEXT I
740 RETURN
```

The subroutine at line 500 makes up a batch of 12 records which are tailored to produce a standard format for information on a drug. Each of these records is one line in length (in fact 39 characters rather than 40, to avoid any problems with continuation lines).

R is the number of the item (e.g. Betaloc is item 1), and N is the number of the record (so Betaloc uses records 0 to 11, for example, and Blocadren uses 12 to 23).

6. NewBrain BASIC with Disk – Random Access Files

```
1000 REM Write A$ at record N
1010 REM Set starting point
1020 X=39*N-39
1030 REM Convert X to binary
1040 N1=INT(X/256):N2=X-N1*256
1050 REM Send header to disk
1060 PUT#1,0,N1,N2,0,0,39
1070 REM Set A$ to 39 characters
1080 A$=LEFT$(A$+"",39)
1090 REM Send A$ to disk
1100 PRINT#1,A$;
1110 RETURN
```

A\$ contains the information for each record in turn. The starting address (X) in the file must be zero for the first set of records. Line 1020 sets the value of X.

The most complex part of the operation is in line 1060. PUT is used to send single bytes through stream 1. Figures 2 and 3 in the PUT list (N1 and N2) represent the starting address of the record, and figures 5 and 6 represent its length.

Each of these pairs of numbers is a two-byte binary number, i.e. figure 3 is the low byte and figure 2 the high byte, and each may contain a number from 0 to 255—so the two together can express any number from 0 to 65535 (e.g. LO+256*HI).

6. NewBrain BASIC with Disk – Random Access Files

If you are using the program as a model for your own programs, you should take particular care with line 1100. The semi-colon at the end may look insignificant, but it is very important. If you miss it out you will send NEWLINE characters to the disk and your records will become unreadable.

In line 1060, the part of the PUT list which expresses the length of the record is quite straightforward: 0,39 means 39 bytes.

The starting address of each record is more difficult to specify since it changes with each successive record. The simplest way is to convert each value of X in turn to a two-byte binary number. This is done in line 1040.

Having established where the record is and how long it is, the data in the variable A\$ has to be adjusted to precisely 39 bytes. This is done in line 1080, and the data is eventually sent to the disk with the PRINT statement in line 1100.

As with the data files described in the last Section, it is important to close the stream after you have sent your data. Data is sent in batches and the command CLOSE (lines 160 and 520) is a signal to send the last batch.

6. NewBrain BASIC with Disk – Random Access Files

READING FROM THE DISK

```
2000 REM Read A$ from record N
2010 REM Set starting point
2020 X=39*N-39
2030 REM Convert X to binary
2040 N1=INT(X/256):N2=X-N1*256
2050 REM Send header to disk
2060 PUT#1,0,N1,N2,0,0,39
2070 REM Build A$ byte by byte
2080 A$=""
2090 FOR I=1 TO 39
2100 GET#1,B$: A$=A$+B$
2110 IF LEFT$(A$,1)=" " THEN A$=""
2120 NEXT I
2130 RETURN
2140
3000 REM Adjust length of A$ and print
3010 A$=LEFT$(A$,39)
3020 PRINT A$
3030 RETURN
```

Getting data back from the disk is very similar to reading onto it, as the program RAF2 illustrates.

The subroutine at line 2000 starts by setting the value of X, converting it to a two-byte binary number and sending this number to the disk, as in RAF1.

The task from here onward is to get the record back from the disk. Unfortunately there is no equivalent of the PRINT command for reading records. Lines 2080 to 2120 look after this process by getting the record back from the disk byte by byte.

6. NewBrain BASIC with Disk – Random Access Files

SUMMARY

TO CREATE A RANDOM ACCESS FILE

1. Open the file: **OPEN#stream,14,"filename"**
Example: **OPEN#1,14,"DRUGS"**
2. Set the starting point of the first record and its length using a PUT statement:

PUT#stream,0,H1,L1,0,H2,L2

where H1, L1 are the starting address of the record, and H2, L2 are the length of the record.

Example: **PUT#1,0,0,200,0,0,50**

This starts a record of 50 bytes at byte 200.

3. Make sure the record you send to device 14 is actually of the length you have specified.
4. Send the record: **PRINT#stream,record**
Example: **PRINT#1,14,A\$**
5. When you have sent all the records, close the stream:
CLOSE#stream
Example: **CLOSE#1**

6. NewBrain BASIC with Disk – Random Access Files

TO READ A RANDOM ACCESS FILE

1. Open the file:

`OPEN#stream,14,"filename"`

Example: `OPEN#1,14,"DRUGS"`

2. Set the starting point of the first record and its length using a PUT statement:

`PUT#stream,0,H1,L1,0,H2,L2`

where H1, L1 are the starting address of the record, and H2, L2 are the length of the record.

Example: `PUT#1,0,1,44,0,0,50`

This starts a record of 50 bytes at byte 300.

3. Get the record back from the disk byte by byte
Example:

`500 Z$=""`

`510 For I=1 to 50`

`520 GET#1,X$`

`530 Z$=Z$+X$`

`540 NEXT I`

This loads the whole of the record into Z\$.

4. Close the stream: `CLOSE#stream`
Example: `CLOSE#1`

7

Cataloguing Files through BASIC

The easiest way of getting a list of files on a disk is the way described in Section 3: transfer into CP/M and type DIR.

If you are working in BASIC, this method has the disadvantage that the moment you leave BASIC you lose whatever program is in memory at the time.

An alternative way of getting a catalogue of programs on a disk is by using device 15, but this is by no means easy. The program below is recorded on your system disk under the name "CAT". Load it and try it out.

Lines 65501 to 65511 open the stream and select the drive to access all file names on the disk. Lines 65512 to 65517 input the file names and display them.

The reason for starting the program at such a high line number is to keep it out of the way of the line numbers where programs are usually written.

Suppose you already have a program in memory which you do not want to lose, but at the same time you want to check the names of the files on the disk, perhaps to make sure you are saving the program on to the right disk. All you need to do is to type

`MERGE "CAT" NEWLINE`

to bring the program into memory, followed by

`GOTO 65500 NEWLINE`

and the list of files will be displayed. When you no longer want the CAT program, you have only to type

`DELETE 65500- NEWLINE`

7. Cataloging Files through BASIC

```
65500 REM CATALOGUE OF FILES
65501 PUT 31: PRINT: ON ERROR GOTO 65519
65502 PRINT "For which drive do you want"
65503 PRINT "a list of files?": PRINT
65504 PRINT: PRINT "Type drive letter and press NEWLINE"
65505 PRINT: LINPUT FS
65506 FS=CHR$(ASC(FS) AND 95)
65507 IF FS<"A" OR FS>"D" THEN PRINT"Drive must be A to D": GOTO 65501
65508 OPEN#15,15
65509 PUT 31: PRINT "Files on drive ";FS
65510 PRINT "-----": PRINT
65511 PRINT#15,3,FS+":*.*)"
65512 FOR I=1 TO 27 STEP 13
65513 LINPUT#15,AS
65514 IF AS="" THEN 65520
65515 PRINT TAB(1);AS;
65516 PRINT#15,4
65517 NEXT I
65518 GOTO 65512
65519 RESUME 65520
65520 ON ERROR GOTO 0
65521 CLOSE#15
65522 END
```

8

Using Cassettes

Once you have fitted a disk controller module and disk drive to your NewBrain Processor, the system will relate to disk only.

However, you may have been using the NewBrain for some time and have many programs on tape that you wish to transfer to disk, or you may even wish to continue using tape from time to time.

Using tape with the NewBrain disk system involves much the same procedures as those described in the **Handbook** for using the TAPE 2 socket with the unexpanded NewBrain.

TO LOAD FROM CASSETTE

1. Open a stream to the cassette port.

For example:

OPEN#2,2 **NEWLINE**

(assuming you are using the TAPE 2 socket)

2. Rewind the tape and press PLAY.
The tape moves and the computer reads the filename and displays it on the screen.
3. Type **LOAD#2 NEWLINE**
and the file will be loaded.
4. Remember to close the stream after loading:

CLOSE#2

VERIFY and **MERGE** follow the same procedure.

8. Using Cassettes

TO SAVE A PROGRAM ONTO A CASSETTE

1. Set the cassette recorder to RECORD.
2. Open a stream to the cassette recorder, specifying the file name.
For example:

OPEN OUT #2,2,"filename" NEWLINE

The tape moves and the file header (the name of the program) is sent to the tape. The tape stops.

3. Type **SAVE#2**
and the program is saved onto tape.
4. Close the stream:

CLOSE#2

9 What is CP/M?

CP/M means 'control program for microprocessors' and its function is to manage the working of your computer and enable it to run a wide range of packages, including text processors, accounts programs and financial forecasting systems.

CP/M itself is a set of control programs recorded on your system disk. Those programs most intimately concerned with operating the computer are not visible to you at all, even when you type DIR and call up the directory. Others such as EXIT are included in the directory and are placed on the disk to enable you to carry out certain functions—in the case of EXIT the function of leaving CP/M.

Other programs still are recorded on the disk to perform some task for you. An example of these is ED, the CP/M text processor and assembler editor.

This manual is not intended to tell you all about CP/M. Such is the popularity of the system that there are many books describing what it is and how to use it. (For example: Rodney Zaks' **The CP/M Handbook**.) The following notes are to help you use CP/M-based packages and to describe briefly how CP/M works.

9. What is CP/M?

COMMANDS

CP/M has two kinds of commands, although you will not notice any difference between them when you use them. In both cases the disk spins as the computer searches for the program which matches the command word you have typed in, and when it has found it, executes it. The only time when you will have to take notice of the different kinds of command is when you come to make copies of disks.

Built-in commands

These are facilities which are part of the CP/M system. DIR is one of these, which puts a list of the files on the disk onto the screen. Others you will find useful are ERA, which erases files, and REN, which allows you to rename files. Built-in commands are automatically transferred as part of the system when you format a new disk from your System Master. (Formatting is dealt with in Section 10.)

Transient commands

These are commands which will only work if the appropriate program is recorded on the disk. You can check whether a transient command is available on any disk by looking it up in the directory. EXIT is an example of a transient command. Others you may find useful are STAT, which tells you how many bytes you have left on the disk; PIP, which allows you to copy programs between disks; ED, the text processor; and SYSGEN, which transfers CP/M to other disks.

Transient commands are not automatically transferred when you format a new disk. If you want them you must copy them onto the new disk using the commands PIP, SCOPY or DCOPY.

9. What is CP/M?

FILE NAMES

File names are usually in two parts: the name and the extension. For example

EXIT COM

The COM is the extension and means 'command'. Other extensions are BAS (files in BASIC), BAK (back-up files created by a text processor), HEX (hexadecimal files).

When program names are typed in full, as they have to be for some commands, you should separate the name from the extension with a full stop:

For example:

EXIT.COM
PIP.COM

BASIC AND CP/M

In the NewBrain system, BASIC and CP/M are used in conjunction with each other. Disks are formatted to be compatible with NewBrain, and then subjected to a further process which equips them with the CP/M system. When you SAVE or LOAD through BASIC, therefore, you are using both systems simultaneously. In the few seconds which follow those very simple commands, a good deal of activity is going on in your NewBrain system.

Earlier in this manual we laid some stress on the need to take care of disks. However, accidents will happen, and the only real insurance against a damaged or lost disk is a complete copy of that disk. This applies particularly to your System Master.

There are three ways in which you can protect vital disks and files, and you should use all of them regularly.

1. Make a duplicate of your System Master

This involves:

- a) formatting a blank disk and copying CP/M onto it. See **FORMAT** in this Section;
- b) copying all the files on the System Master onto the duplicate. See **PIP** and **SCOPY** in this Section.

PIP can also be used to copy any or all of the files on any disk onto another disk, or to output a file to a printer.

2. Write-protect your System Master

—and any other disks which are particularly valuable to you and which you do not wish to write onto.

To write-protect a disk, stick one of the small aluminised strips (provided with new disks) over the notch in the side of the disk.

With this in place, no file can be written onto the disk and an error message will be given if you try to do so.

3. Write-protect valuable files

Files on unprotected disks may also be protected. This is especially needed when files have similar names (e.g. when part of a series) and so may be over-written in error.

To write-protect a file, see **STAT** in this Section.

FACILITIES

The following are a selection of facilities available to users of the NewBrain disk system. CP/M commands are included, but for a complete account of the facilities offered by CP/M you should consult the CP/M user's guide and other literature on CP/M such as Rodney Zaks' **CP/M Handbook**.

Facilities

Autostarting programs
 Configure CP/M
 Copy files
 Directory of files
 Display text files
 Erase files
 Format disks
 Load BASIC programs
 Print out files
 Protect files (read-only)
 Rename files
 Save BASIC programs
 Statistics of files
 Switch between BASIC and CP/M
 Text editing

Commands

SETINIT
 CONFIGUR
 PIP, SCOPY, DCOPY
 DIR
 TYPE
 ERA
 FORMAT, SYSGEN
 LOAD
 TYPE
 STAT
 REN
 SAVE, SAVF
 STAT
 EXIT, CPM
 ED

DISK DRIVES

When the system starts up in CP/M or changes to CP/M, it is related to drive A and the screen shows the prompt:

A>

To change to drive B, type

B: NEWLINE

The prompt will change to

B>

and the system will relate to B (or any other disk drive you specify).

The commands on the following pages can be used with any drive which has a correctly formatted disk in it containing the correct programs.

PAUSE COMMAND

It is often convenient to limit the speed with which some commands operate: for example, TYPE which displays text files on the screen in CP/M.

The action of such programs can be made to pause by holding down **CONTROL** and pressing **S**. To set the process going again, hold down **CONTROL** and press **S** again.

CONFIGUR**CP/M TRANSIENT COMMAND**

Action: configures various parameters of the CP/M operating system

Use: Not recommended for the non-technical user.

The program asks a series of questions about disk format and I/O device parameters, which are answered in sequence. Configuration information stored onto a disk is activated only after CP/M is re-entered with that disk in drive A:.

CPM**BASIC COMMAND**

Action: switches the system from BASIC to CP/M.

This command is available only on simple disk systems, i.e. with a Disk Controller only. If an Expansion Interface Module is fitted, the command EXIT is used.

Type: CPM NEWLINE

CP/M BUILT-IN COMMAND**DIR**

Action 1: lists all files on a disk.

Type: DIR NEWLINE

Action 2: lists a specified file.

Type: DIR FILE.EXT NEWLINE

Action 3: lists files on a specified disk drive

Type: DIR B: NEWLINE
lists the files on drive B.

Action 4: lists all files with a particular name

Type: DIR .* NEWLINE

Action 5: lists all files containing certain characters

Type: DIR CHRS?.EXT NEWLINE
(where CHRS represents the known characters)

CP/M TRANSIENT COMMAND**ED**

Action: edits a named file or creates a new text file.

Type: ED FILE.EXT NEWLINE

ED is the CP/M text processor and assembler editor.

ERA**CP/M BUILT-IN COMMAND**

Action 1: erases one file from the current disk.

Type: **ERA FILE.EXT** **NEWLINE**

Action 2: erases all files on a disk, after confirming that you want to do this.

Type: **ERA *.*** **NEWLINE**

Action 3: erases all files of a certain kind (e.g. HEX) on the current disk.

Type: **ERA *.HEX** **NEWLINE**
erases all HEX files on the disk.

Action 4: erases all files of a certain name (e.g. all INDEX files) on the current disk

Type: **ERA INDEX.*** **NEWLINE**

Action 5: erases files on a disk other than the current disk.

Type: **ERAB: FILE.EXT** **NEWLINE**
erases the file from drive B.

The form *.* can also be used to erase all files from another drive.

**CP/M TRANSIENT COMMAND
BASIC COMMAND****EXIT**

The command EXIT can be used in BASIC, if you have an Expansion Interface Module fitted. It can also be used in CP/M if the program EXIT.COM is on the disk concerned.

Action: if a Disk Controller only is fitted, exits from CP/M into BASIC;

if an Expansion Interface Module is fitted as well, exits from CP/M or BASIC back to the starting menu.

Type: **EXIT** **NEWLINE**

NOTE: the disk is now formatted so that it is compatible with the NewBrain system, but it cannot yet be used to get into CP/M. Go on to the next procedure: SYSGEN (item 7)

TO FORMAT DISKS USING TWO DRIVES OR MORE

1. Make sure the System Master disk is write-protected.
2. Place the System Master disk in drive A.
3. Type **FORMAT** **NEWLINE**
 REPLY: NewBrain disc format program Version 1.0
 Copyright © 1983 Grundy Business Systems
 Please enter letter of drive on which disc is to be formatted or press newline to exit program
4. Type **B** (or whatever drive you are using)
 REPLY: Place blank disc in drive B: and press newline
5. Place the blank disk in B and press **NEWLINE**
 The disk spins and 'ticks' for about a minute.
 DO NOTHING!
 REPLY: Please enter letter of drive on which disc is to be formatted or press newline to exit program
6. If you have another disk to format, repeat the procedure. If not, press **NEWLINE**
 The prompt reappears:
A>
 and the system is ready for another command.

NOTE: you must format every blank disk using both these commands before it is possible to use it in your system. If you format a disk which already has programs stored on it you may lose the programs.

TO FORMAT DISKS USING ONE DRIVE ONLY

1. Make sure the System Master disk is write-protected.
2. Place the System Master disk in the drive.
3. Type **FORMAT** **NEWLINE**
 REPLY: NewBrain disc format program Version 1.0
 Copyright © 1983 Grundy Business Systems
 Please enter letter of drive on which disc is to be formatted or press newline to exit program
4. Type **A**
 REPLY: Place blank disc in drive A: and press newline
5. Remove the System Master, place the blank disk in the drive and press **NEWLINE**
 The disk spins and 'ticks' for about a minute.
 DO NOTHING!
 REPLY: Please enter letter of drive on which disc is to be formatted or press newline to exit program
6. If you have another disk to format, repeat the procedure. If not, remove the disk. Place the system master in the drive and press **NEWLINE**
 The prompt appears: **A>**
 and the system is ready for another command.

NOTE: the disk is now formatted so that it is compatible with the NewBrain system, but it cannot yet be used to get into CP/M. Go on to the next procedure: SYSGEN (item 7)

7. Type **SYSGEN** **NEWLINE**
 REPLY: **NewBrain CP/M System Generation program Version 1.0**
Copyright © 1983 Grundy Business Systems
Source drive?

This means that you should type the letter of the disk drive where the System Master is (i.e. A) and press **NEWLINE**

Type **A** **NEWLINE**
 REPLY: **Destination drive?**

8. This means that you should press the letter of the drive where the disk is onto which you wish to transfer CP/M, and press **NEWLINE**.

If you wish to stop the procedure, just press **NEWLINE** alone, and the system will reboot, i.e. return to the prompt and be ready for other commands.

If you are using one disk drive only

- remove the System Master and insert the disk on which you wish to place the CP/M system
- type **A** **NEWLINE**

If you are using more than one drive

- type **B** (or whatever drive you are using)
- press **NEWLINE**

9. The disk spins for a few seconds.

REPLY: **Destination drive?**

If you have other disks you wish to process, repeat the procedure. If not just press **NEWLINE** and the system will return to the prompt:

A>

and is ready for other commands.

NOTE: the disk you have formatted is now ready to have programs recorded onto it, but at the moment it cannot operate with any CP/M transient commands at all. You can check this by typing

DIR **NEWLINE** (if it is in drive A)
DIR B: **NEWLINE** (if it is in drive B)

and the system will reply: **NO FILE**

TRANSFERRING PROGRAMS TO THE NEW DISK

With one drive only: use the SCOPY command.
 With two drives or more: use the PIP command.

LOAD

BASIC COMMAND

NOTE: you must be in BASIC to use this command.

Type: **LOAD "filename" NEWLINE**

Action: LOAD erases whatever program is currently in the computer's memory and replaces it with the program requested. To load from drive B, use the form **LOAD "B:filename"**

If there is no such program on the disk, the computer will give an error message.

PIP

CP/M TRANSIENT COMMAND

Action: copies files from one disk to another, provided that the program PIP.COM is present on one of the disks.

GENERAL METHOD (assuming the system is currently in drive A)

Place a disk with PIP.COM on it in drive A

Type: **PIP NEWLINE**

The prompt changes from **A>** to *****

This means that you are now 'in PIP'.

1. To copy file EXIT.COM from drive A to drive B, keeping the name the same:

Type: **B: =EXIT.COM[V] NEWLINE**

(The [V] does a verification check against the file in the user area.)

PIP

2. To copy file RECORD from drive A to drive B, changing the name to RESULT:

Type: **B: RESULT=RECORD NEWLINE**

or: **B: RESULT=A: RECORD NEWLINE**

(although if A is the current drive it is not necessary to specify it)

3. To copy all files from drive A to drive B:

Type: **B: =*. * NEWLINE**

To exit from PIP, press **NEWLINE.**

METHOD FOR SINGLE FILES

If you wish to copy only one file, you need not go into PIP as a separate action. Instead you can use the PIP command along with details as to what it should do.

To copy file RAF2.BAS from drive A to drive B, changing the name to FILE2.BAS type:

PIP B: FILE2.BAS=RAF2.BAS [V] NEWLINE

REN**CP/M BUILT-IN COMMAND**

Action: to rename a file

Type: **REN NEW.EXT=OLD.EXT** **NEWLINE**

The new file name with its extension comes first, followed by the old file name.

For example: **REN COMP.BAS=MUSIC.BAS** **NEWLINE**

This changes the file name from MUSIC.BAS to COMP.BAS

SAVE**BASIC COMMAND**

NOTE: you must be in BASIC to use this command.

Action: records a BASIC program onto the disk.

Type: **SAVE "filename"** **NEWLINE**

SAVE records the file named onto the current disk. If a file of that name is already on the disk, it is over-written, unless it has been converted to a read-only file (see STAT).

To save to drive B, use the form **SAVE "B:filename"**

SAVF**BASIC COMMAND**

Action: records a BASIC program onto the disk or cassette in such a way that it will LOAD or MERGE faster than it would with SAVE. Programs recorded with SAVF will not VERIFY.

SAVF is used in the same way as SAVE

CP/M TRANSIENT COMMANDS**SCOPY**

Action: copies a disk, track by track, when only a single disk drive is available.

Type: **SCOPY** **NEWLINE**
The program tells the user alternately to
'Place the source disc in drive A.'
and
'Place the destination disc in drive A:.'

CP/M TRANSIENT COMMAND**DCOPY**

Action: copies a disk, track by track, from one drive to another.

Type: **DCOPY** **NEWLINE**
The program tells the user to place the source disk in drive A and the destination disk in drive B. After pressing **NEWLINE** the disk is copied track by track.
This program will only work on two similarly configured drives. PIP should be used otherwise.

SETINIT

CP/M TRANSIENT COMMAND

Action: causes a command to be executed on booting. SETINIT can help non-expert users by causing the computer to start running a specific program immediately after switching on.

Type: SETINIT NEWLINE

Reply: NewBrain CP/M initial command setting program Version 1.0
Copyright © 1983 Grundy Business Systems

Enter desired initial command or just return to clear it

Type: the command, and press NEWLINE You should type the command itself (not the file name and extension) just as you would if you were entering the command to be executed.

If an initial command has already been set and you wish to remove it, just press NEWLINE alone.

Example

Type: SETINIT NEWLINE

Reply: (as above)

Type: PIP NEWLINE

To test the effect, get out of CP/M by typing EXIT. Then return to CP/M again. When the system boots, you will see that the CP/M prompt (A>) has been replaced by the PIP prompt (*). (To get out of PIP, just press NEWLINE)

CP/M TRANSIENT COMMAND

STAT

Action 1: gives the number of bytes still available for use on the current disk.

Type: STAT NEWLINE

Reply: R/W, Space: 120k

This would mean that the disk was in read-write condition (i.e. not write-protected) and that there were 120k bytes still available for use.

Action 2: gives statistics for a specific file

Type: STAT FILE.EXT \$\$

Reply:

Size	Recs	Bytes	Ext	Acc	
1	1	4k	1	R/W	A: FILE.COM

The information in this example would mean:

Size: how many 128 byte records are allocated to the file, whether or not the file is using them.

Recs: the number of records in each 'extent' (i.e. each 16k block)

Bytes: the actual number of bytes allocated to the file—note that 4k bytes is the minimum.

Ext: how many 'extents' are allocated to the file.

Acc: what type of access is allowed, i.e. R/W (read-write) or R/O (read-only)

A: the file is on the disk in drive A.

STAT

Action 3: gives statistics of all files.

Type: **STAT *.*\$S** **NEWLINE**

This can be varied to show files without extensions only, for example:

STAT *.*\$S **NEWLINE**

Action 4: gives statistics of files of disks on other drives.

Type: **STAT B: *.*\$S** **NEWLINE**

which gives statistics of all files on drive B.

Action 5: converts a file to read-only.

Type: **STAT FILE.EXT \$R/O** **NEWLINE**

To reverse the process, type:

STAT FILE.EXT \$R/W **NEWLINE**

Action 6: gives the characteristics of the current disk drive.

Type: **STAT DSK:** **NEWLINE**

SYSGEN

see **FORMAT**

CP/M BUILT-IN COMMAND

TYPE

Action 1: displays the contents of any file on the screen, but it is only possible to read the contents of text files, such as are created in the course of text processing.

Type: **TYPE FILE.EXT** **NEWLINE**

Action 2: prints out the contents of a file on a printer.

Press: **CONTROL P**

Type: **TYPE file** **NEWLINE**

Press: **CONTROL P**
to close the printer stream again.

BOOTING

When you start off the system in CP/M, or transfer it to CP/M, the disk spins for a few seconds as the system goes through a process known as 'booting'. In booting the computer reads CP/M from the disk and starts to use it.

If at any time CP/M appears to be 'stuck' and will not obey the commands you know to be on the disk, you can 're-boot' by holding down **CONTROL** and pressing **C**.

Remember, however, that if you re-boot, any program you are using will cease to function and any text or data you have entered will be lost.

CONTROL CODES

Almost all control codes retain their normal meaning as listed and explained in the **Handbook**. The only difference is:

5 CONTROL E Cursor ceases to flash

The effect of control codes in CP/M, however is not immediate. For example, clearing the screen can still be done with SHIFT HOME, but SHIFT HOME by itself simply places a vertical arrow and a bar on the screen. When NEWLINE is pressed, the screen clears, but CP/M has received an instruction it cannot use, so it responds with a query. The effect, however, has been achieved.

11. Using CP/M with NewBrain

PRINTER

Connecting up Simple disk system: connect to printer socket on NewBrain processor

Expanded disk system: connect to printer socket on expansion interface module

Use with BASIC The printer is accessed through device 8, unless connected to the Expansion Interface Module. Access is then through device 16.

Use with CP/M When working with CP/M packages, follow the instructions with the package and the printer.

To print the contents of files

—press **CONTROL P**

—type **TYPE FILENAME** **NEWLINE**

WARNING. When using a printer with CP/M packages, always connect up and switch the printer on **before** starting to use the package.

RETURN

The word RETURN is used in instructions with many CP/M packages. On the NewBrain keyboard, this is

NEWLINE

11. Using CP/M with NewBrain

DELETE

An instruction to delete is sometimes specified in CP/M packages. NewBrain has no DELETE key, but the same effect can be achieved by holding down **CONTROL** and pressing the **COMMA**

KEYBOARD

Teletype caps: press **CONTROL 8**

The effect is to produce shifted characters for A–Z, and unshifted characters for other keys.

To reverse the effect, press **CONTROL 8** again.

11. Using CP/M with NewBrain

SCREEN BACKGROUND

The system defaults to white letters on a black background. This can be changed as follows:

BLACK ON WHITE—press **CONTROL W** followed by **CONTROL A** and **NEWLINE**

WHITE ON BLACK—press **CONTROL W** followed by **CONTROL B** and **NEWLINE**

Although the effect is achieved by this method, CP/M will query the input because two control characters do not constitute a valid CP/M command.

12 Error Codes

When using BASIC with disk the following error codes may appear on the screen. Other error codes are listed and described in the **Handbook** and in Section 13.

- 150 – Bad filename. The filename provided as parameter when opening device 12, 13 or 14 or a device 15 argument is syntactically incorrect.
- 151 – Input error. Either a physical disk error has occurred or the end-of-file has been reached on device 12.
- 152 – Output error. Either a physical disk error has occurred or the disk has become full.
- 153 – Directory error. A file requested does not exist in the directory, the directory is full, or there was a physical disk error when reading the directory.
- 154 – Initialisation error. The CP/M filing system could not be loaded, either due to lack of memory or a disk fault.
- 155 – Transaction error. With devices 14 and 15, inputting when an output was expected, or vice versa.
- 156 – Not port zero. Opening device 15 with a non-zero port.
- 157 – Filename too long. A filename presented to device 15 is too long.
- 158 – Transaction type error. A transaction number presented to device 15 is illegal.

12. Error Codes

159 – Random transput error. Attempt to access a file beyond 8 Mbytes, or a length of zero has been specified.

With some physical disk errors, a message may be output to the console in addition to the error code.

13

Notes on the Expansion Interface Module

The NewBrain Expansion Interface Module contains the NewBrain paged memory system and 64K RAM. Up to 1 megabyte of RAM may be added for fast access disk replacement, and up to 1 megabyte of ROM for additional software, in blocks of 64K, 128K, or 256K. The NewBrain Expansion Interface Module also includes a full RS232 MODEM port for telephone communications, a parallel printer port, and digital, parallel and analogue ports for industrial and laboratory use, and for use with special printers.

The Expansion Interface Module enables the NewBrain to access paged memory and distinguishes between ROM and RAM pages. The memory is divided into 8K blocks or pages, and 64K can be accessed at any one time.

13. Expansion Interface Module

Device drivers available on expanded systems

CP/M

Device 12	BDISCIO Binary file serial I/O facility
Device 13	TDISCIO Text file serial I/O facility
Device 14	RDISCIO Random file access I/O
Device 15	SDISCIO Direct read and update

Expansion Interface Module

Device 16	ACLPIO ACIA line printer device
Device 17	ACVIO ACVIO driver
Device 18	SSEIO Simple screen editor
Device 19	(not yet available) Keyboard driver with buffers
Device 20	SMDD Serial memory device driver
Device 21	CENTR Centronics printer driver
Device 22	APIO Analogue I/O driver

13. Expansion Interface Module

Paged Operating System Error Numbers

PIOS

- 104 – String too long. The parameter string supplied to OPEN is longer than 256 bytes
- 105 – Stream not open. Stream referred to by INPUT, OUTPUT, BLKIN, or BLKOUT is not open
- 106 – Attempt to OPEN a non-existent device
- 107 – Attempt to OPEN a device/port that is already open
- 108 – Attempt to open a stream that is already open
- 109 – Attempt to use a device function that does not exist

Page Allocator

- 199 – Insufficient pages in system for REQUESTPAGE request

Serial Device

- 192 – Read past end of file
- 193 – Insufficient memory

13. Expansion Interface Module

Centronics Printer

- 120 – Error in parameter string
- 195 – Printer number out of range
- 206 – Buffer does not exist
- 207 – Buffer already in use
- 208 – Buffer full or data could not be output to ACIA
- 209 – Illegal use of multiplexed ACIA attempted
- 210 – Buffer empty or no data to input from ACIA
- 211 – Attempt to run two ACIA's on the same peripheral interface
- 212 – Attempt to run multiplexed ACIA at high speed
- 213 – (not used at present)
- 214 – Illegal buffer number
- 215 – Illegal combination of data and stop bits and parity requested
- 216 – ACIA framing error
- 217 – ACIA overrun error
- 218 – ACIA parity error
- 219 – ACIA loss of carrier detected

13. Expansion Interface Module

Error numbers for BASIC extensions with peripherals

- 58 – Use of an EXTERNAL subroutine as a function
- 59 – Use of an EXTERNAL function as a subroutine
- 60 – The name of a requested EXTERNAL routine cannot be found in the attached library
- 62 – Attempt to use a function with a string result parameter within a DEF statement
- 64 – Wrong type of parameter

Index

Background	82
BASIC	18, 25, 57
Booting	79
Built-in commands	56
Cassettes	53
Catalogue of files (BASIC)	51
Commands	56
CONFIGUR	62
Connecting up	9
Control C	79
Control P	77, 80
Control S	61
Control Codes	79
Copying files	21, 23, 70, 73
CP/M	16, 55, 60
CPM	62
Data files	31
DCOPY	73
Delete	81
Deleting files	64
DIR	16, 28, 55, 63
Devices	31, 38, 42, 86
Directory	7, 16, 25, 55, 63
Disc controller module	5
Disk drives	14, 61
Disks, care of	14
ED	55, 63
ERA	64
Error codes	83, 87-89
EXIT	28, 55, 65
Expanded disk sytem	13
Expansion interface module	5, 85
Files	7
data	31
names	27, 57
protection	29, 59
random access	41

Index

FORMAT	20, 59, 66
Formatting	8, 20
Housing	9
INDEX (programs)	25, 32
Keyboard	81
LOAD	25, 53, 70
Menu	13
MERGE	25
Pause Command	61
PIP	21, 59, 70
Printing	77, 80
Programs	3
BASIC	25
CP/M	55
RAM	3
Random access files	41
Read only/read write	29
REN (renaming files)	72
RESET	18
Return	80
SAVE	27, 72
SAVF	72
SCOPY	23, 59, 73
Screen background	82
Sectors	6
SETINIT	74
Simple disk system	13
STAT	29, 59, 75
Streams	31
SYSGEN	22, 66
Teletype caps	81
TYPE	77
Tracks	6
Transient commands	56
Write-protection	59

Grundy Business Systems Ltd.
Somerset Road
Teddington
Middlesex TW11 8TD

This handbook was written by
Information Transfer Ltd., Cambridge.

