

INTRODUCTION TO DOS

The Disk-based operating system used on almost all PCs is generically known as DOS. There are four types that may be encountered, in several versions. The commonest versions are as follows:

MS-DOS, as used on about 90% of all non-IBM PCs:
Versions 3.2, 3.3, 4.0, 4.01, 5.0, 6.0 and 6.2 are common

PC-DOS, as used on genuine IBM machines:
Versions 3.2, 3.3, 4.01, 5.0, 6.1 and 7 are common

DR-DOS, a rival to MS-DOS, used on many non-IBM machines over the last few years:
Versions 5.0 and 6.0 are common.
Version 7.0 of DR-DOS is now called Novell DOS Version 7.0

Windows95 DOS, the DOS kernel of the Windows95 operating system. Microsoft have chosen to call this MS-DOS 7 for compatibility.

There have been other types in various versions (e.g. Compaq DOS, Tandon DOS, MDos).

If you are able to choose a DOS for your own computer, choose either DR-DOS version 6.0, MS-DOS versions 6.2, or PC-DOS version 7. Avoid MS and PC DOS versions 4, 4.01 and 6.0 like the plague!

In the time available, I cannot teach you to become operating system gurus. The best way to learn DOS is to make yourself a system disk and to play.

Most of what we are going to do on this course works with all the common versions listed above. The version we will actually be using varies from centre to centre..

The RSA CLAIT File Management (DOS) Certificate.

If you are taking this course and using this book in conjunction with an assessment for the above certificate, your tutor will cover additional topics necessary for you to achieve all elements. In addition, you will be pleased to know that the course and this book go beyond what is required by the RSA!

CONVENTIONS USED IN THIS BOOK.

Computer output is displayed in `Courier Font`

DOS commands (i.e. what you type in) are displayed in `Bookman Old Style Font`.

Syntax lines are displayed in `Century Gothic font`.

FILES

Work is stored on disk in FILES. Each file will have a filename and (usually) an extension. The filename can be up to 8 characters and the filename up to three. The name and extension are separated by a full stop. Windows 95 will allow longer filenames under certain circumstances.

Examples of filenames are:

CONFIG.SYS
COMMAND.COM
CHESS.EXE
MYWORK.TXT
READ.ME

There can be no spaces in a filename, although numbers and certain other characters are allowed. Filenames are not case sensitive, either upper or lower case being allowed.

Certain extensions are reserved by DOS to mean certain things:

SYS - a SYStem file or device driver
COM - a COMmandable file
EXE - an EXEcutable file
BAT - a BATch file.

SYS files are normally only accessed on system start-up or "boot" COM EXE and BAT files are the files which can be run from the DOS command line. For the average user, there is no difference between COM and EXE programs. However, BAT files are text files of lists of instructions that are well within the capabilities of average DOS users to create.

Other extensions are conventionally used for specific purposes although not demanded by DOS. Some of the more common are:

DOC - a document file created by a word processing package
TXT - an ASCII text file
INI - a program initiation file.

Whilst talking about files, there are three files which must be present on any disk if it is to be bootable, one of which is COMMAND.COM, the DOS command processor. The other two are often "hidden" so that they cannot be erased. In addition, you will also find CONFIG.SYS and AUTOEXEC.BAT configuration files which can be changed by the user to suit their own specific requirements.

DRIVES

The different drives on your computer are labelled by letters of the alphabet followed by a colon, viz.:

A: B: etc.

All PCs have an A: This is the floppy drive from which the computer can be booted. A system disk in A: will always be used to boot from even if there is a bootable hard disk.

A second floppy if fitted will be B:. If there is no second floppy, the first can also be accessed as B:.

DOS has the ability to divide hard disks into partitions, each with its own drive letter. You may therefore have one hard disk which is divided into two or more drives (these are called logical drives), or you can have two or more hard disks, any of which may contain logical drives.

Whatever the arrangement, the first partition on the first hard disk is C:, and is normally bootable, so the computer will boot without the need to insert a system disk. Large systems have many drives; the author's system currently goes to Z:.

Other devices, notably CD-ROM drives, also have a drive letter. These letters are allocated on boot after all the hard disk partitions have been accounted for. For DOS purposes, CD-ROMs can be regarded as very large read-only floppy disks!

In order to make organising disks more easy, drives can be divided into subdirectories. Hard disks are always so divided, floppy disks less often.

The root, or main directory, (sometimes called top level directory) is the one that always exists, and from which other directories, often called subdirectories, branch.

The rules for directory names are exactly the same as those for files, although extensions are not normally used.

The layout of directories and files on a disk is called a TREE. Here is a simple example:

```
C:---|
|
|-COMMAND.COM           File in root directory
|-AUTOEXEC.BAT          "
|-CONFIG.SYS            "
|-DOS                   Directory
|  |-CHKDSK.EXE         File in DOS subdirectory
|  |-KEYB.COM           "
|  |-etc. etc.
|-WORKS                 Directory
|  |-WORKS.EXE          File in WORKS subdirectory
|  |-WORDS              Subdirectory of WORKS
|    |-LETTER1.DOC      File in WORKS\WORDS subdirectory
|    |-LETTER2.DOC      "
|  |-SPREAD             Subdirectory of WORKS
|    |-SHEET1.WS        File in WORKS\SPREAD subdirectory
|    |-SHEET2.WS        "
```

Note the use of the \ (backslash) above. To give the full location (PATH) of a file, we use the form:

C:\WORKS\SPREAD\SHEET2.WS

This means "the file SHEET2.WS, found in the SPREAD subdirectory of the WORKS subdirectory on C:"

When housekeeping in DOS, it is not always necessary to quote the full path, but it is NEVER, EVER WRONG to do so, and may prevent errors.

The best way to introduce the main DOS commands is to try them. Put your practical disk in A:, and either turn the computer on, or if it is already on reboot by pressing the CTRL, ALT and DEL keys simultaneously (the "three fingered salute").

COMMANDS

When the computer has finished going through its tests, and loaded DOS from your disk, it gives the PROMPT, which looks like:

```
A:\>
```

This means "I am in the root directory of A:, what do you want me to do now?"

Prompts can be user-configured, so if your usual prompt is more fancy, don't worry - we are only using the basics here.

The first command we shall use will be the **DIR** command. This means "give me a directory listing". As with many DOS commands, DIR can take PARAMETERS, defining the file(s) which the command is to act on, and SWITCHES, which modify the way in which it acts. For example:

DIR *.EXE means "give me a listing of all files with the extension .EXE". As with all commands, you will have to look up the available parameters and switches in your DOS manual and practice.

Anyhow, type DIR followed by a carriage return and something like this should come up (note the details will vary from centre to centre)

```
A:\>DIR
```

```
  The volume label in drive A is NO NAME      FAT.
```

```
  Directory of A:\
```

```
IBMBIO   COM      24442   23-08-91   6:00a
IBMDOS   COM      39239   23-08-91   6:00a
COMMAND  COM      50456   23-08-91   6:00a
RUNNER   EXE      31004   15-07-92   3:18p
TEXTFILE TXT         115   15-07-92   3:14p
AUTOEXEC BAK         50   15-07-92   3:28p
CONFIG   SYS         54   15-07-92   3:27p
AUTOEXEC BAT         61   15-07-92   3:31p
DOS              <DIR>    12-09-92   4:58p
          9 file(s)    145421 bytes used
                          507904 bytes free
```

This has listed all the files in our root directory, in no apparent order. Reading across, you have the file name, extension, file size (in bytes), the date it was created, and the time (often used for the version number by large software houses). You should note that although DIR lists files with spaces between the filename and the extension, if we do this it will complain! We must always type in filenames with a full stop as the separator, i.e. IBMBIO.COM.

Note the entry for DOS. This is how a directory or subdirectory is shown.

How do we find out what's in the DOS directory? We need to learn a new DOS command - the one to Change Directories. Give the command:

```
CD A:\DOS
```

CD is the command to change directory, and we have instructed the computer to change to the DOS directory.

The prompt should read:

```
A:\DOS>
```

showing we are in the DOS directory.

Give the DIR command and see what's in this directory.

Don't worry about the . and .. entries - they are part of how DOS works, and of no consequence at this stage.

NOTE: you cannot use CD to change DRIVES. For example, if we wished to change from the F:\UTILS directory to the Q:\GAMES directory, we would have to give two commands:

```
Q:
```

```
CD Q:\GAMES
```

the first command giving the instruction to change drives, the second to change directories.

Sometimes, we want to clear the screen. The DOS command to do this is **CLS**. Try it now. All we are left with is the prompt at the top of the screen.

Screen clear?

Now we shall create a new directory. We will give it the name NEWDIR, and it will be a subdirectory of A:\.

First, we must go back to the root directory:

```
CD A:\
```

(the backslash on its own signifies the root directory)

Now give the **MD** command to make a directory:

```
MD A:\NEWDIR
```

Nothing seems to have happened - we have no message and the same prompt! This is because DOS often works on the "no news is good news" principle, i.e. if it can do something it will without making a song and dance about it. To prove your new directory has been created you will need to do a DIR.

Now change to your new directory - I'll leave you to work out how. Do a directory listing. Not surprisingly, there's nothing in it except the . and .. entries, since you've only just created it, but we're going to use it to demonstrate the next command: **COPY**

The general syntax for COPY is deceptively simple:

```
COPY [source file] [destination file]
```

or even more simply:

```
COPY [from] [to]
```

If you use full file paths, as you will have been told to do, you can't go far wrong.

As an example, we will copy the file COMMAND.COM from the root directory to the NEWDIR subdirectory:

```
COPY A:\COMMAND.COM A:\NEWDIR\COMMAND.COM
```

The file will show completion with the message:

```
1 file(s) copied
```

Your turn. Copy TEXTFILE.TXT from the root directory to the NEWDIR directory and prove it has been copied.

The commonest form of file after program files (COM and EXE files) is the text file, often called an ASCII text file.

A text file can be displayed on the screen with the **TYPE** command.

Make sure you are in the NEWDIR directory and give the command:

```
TYPE A:\NEWDIR\TEXTFILE.TXT
```

The contents of TEXTFILE.TXT are displayed.

Funny things happen when you type to type something other than a text file. Just for fun, try TYPEing the file A:\NEWDIR\COMMAND.COM
You just get junk and a lot of beeps!

COMMAND.COM is a program file (as it happens a very important one), as is RUNNER.EXE which is in the root directory. If you try to TYPE RUNNER.EXE nothing sensible appears, but wait and see what happens if you try to RUN or EXECUTE it.

There is no special DOS command for running a program file - just typing in the name of the file at the DOS prompt does the trick. If the file is not a program file, you will get an error message, otherwise DOS will run it.

So:

- (1) Change to the root directory
- (2) Try the following commands:
A:\RUNNER.EXE
\RUNNER.EXE
RUNNER.EXE
RUNNER

They all work and show that sometimes, if you know what you are doing, DOS commands can be shortened. But beware - try this!

change to the NEWDIR subdirectory (prompt A:\NEWDIR>)
now give the command:
NEWDIR\COMMAND

Command or filename not recognised.

Try to work out why!

Files (but not usually directories) can be renamed with the **REN** command.

The syntax is:

REN [old name] [new name]

Change to the root directory and do a directory listing.

Note how we have 10 entries - the NEWDIR directory we created is now listed.

What we are going to do is to rename TEXTFILE.TXT to NEWFILE.TXT

REN A:\TEXTFILE.TXT A:\NEWFILE.TXT

There is no message - just the prompt back - so do a DIR. TEXTFILE.TXT has disappeared and NEWFILE.TXT has appeared in its place. Try TYPEing NEWFILE.TXT - it is exactly the same as the old TEXTFILE.TXT. Change to the NEWDIR directory, and get a DIR. You will see that the TEXTFILE.TXT we copied earlier is still there - we only RENamed the copy in the root directory.

If COPY is the most used DOS command, **DEL** is one of the most dangerous. DEL means delete, and it is very easy to delete files you don't mean to. Therefore, when using DEL for real on your own computer, I urge you to use full file paths, and to double check the command line before pressing return. On our practice disk, of course, it doesn't really matter if we make a mistake, so don't worry!

Do a DIR for the root directory. You will notice two similarly named files: AUTOEXEC.BAT and AUTOEXEC.BAK. AUTOEXEC.BAT is a user-changeable configuration file, while AUTOEXEC.BAK is a backup - it contains the previous version and is no longer required. Delete it:

```
DEL A:\AUTOEXEC.BAK
```

No message! Check with a DIR. Hopefully, AUTOEXEC.BAT is still there, but AUTOEXEC.BAK has disappeared. I know that there are undelete utilities provided with the latest versions of DOS, but these cannot be relied on, so the following warning still carries a lot of weight:

```
**** ONCE YOU HAVE DELETED A FILE, YOU CANNOT NORMALLY  
GET IT BACK ****
```

If you wish to delete a directory, you can't do so with the DEL command.

Before we delete any directories, lets introduce the **wildcards * and ?**

The asterisk (*) means "anything", i.e. *.* means any file with any extension. Examples:

A:*.DOC - any file in A:\ with the extension .DOC

A:\AUTOEXEC.* - any file in A:\ named AUTOEXEC with any extension.

A:\DOS*.* - any file in the DOS directory

A:\AUTO*.* - any file in A:\, with any extension, the first four letters of the name being AUTO

Just as an aside, this won't work:

A:*EXEC.*

as once DOS has seen the asterisk, it ignores the rest and treats this the same as A:*.*.

To overcome this to some extent, we can use the ? wildcard.

? means any single character, which must exist, i.e. ? can't stand for nothing.

Examples:

A:\AUTOEXEC.BA? would cover AUTOEXEC.BAT and AUTOEXEC.BAK

A:\AUTO?????.BAT would cover AUTOEXEC.BAT and AUTOCART.BAT but not AUTOCAR.BAT

OK, now we know about wildcards, we can explain how to remove a directory. **YOU CAN ONLY REMOVE (DELETE) AN EMPTY DIRECTORY.**

First we must delete all the files in it, and then use the command **RD** [directory] to remove it - but you can't remove a directory if you're in it, you must remove it from one level back, i.e. from the PARENT directory.

Lets illustrate this by removing the NEWDIR directory.

First change to the NEWDIR directory and delete all the files in it (using wildcards)

Now change back to the root directory and give the command:

```
RD A:\NEWDIR
```

Check with a DIR - no NEWDIR!

VER - this command tells you what version of what OS you are using (if you can't remember).

```
VER
```

SWITCHES AND PARAMETERS

Many commands can take switches, which modify the way in which the command acts, as well as parameters, which define the files the command acts on. Switches usually (but not always) are defined after the parameters, and are shown by the forward slash (/) As an illustration, lets expand our use of DIR.

DIR takes a parameter, telling it exactly what to list, and several switches, including /S and /W. A parameterless DIR, which is what we have used until now, lists all files in the current directory, and is equivalent to DIR *.*.

Change to the DOS directory of your practice disk. You have three files, COUNTRY.SYS, EDITOR.EXE AND KEYB.COM Try:

```
DIR *.EXE
DIR EDITOR.*
DIR *.*
DIR EDITOR.EXE
DIR *.BAT
```

Note the "File not found" message for the last

There are two useful switches with DIR: /W and /P

If we are dealing with long directories with many entries, the listing can scroll off the screen.

DIR /W lists the files in a wide format, 5 across the screen, but omits much of the information.

DIR /P lists the files in the usual format, but pauses when the screen is full until you press a key.

Try them.

Switches with COPY.

COPY has many switches, but only two are relevant at this stage:

```
COPY [from] [to] /V
```

The V switch verifies (checks) the copy to make sure it is an exact copy. This takes time.

```
COPY [from] [to] /C
```

Useful when using wildcards, the C switch asks for confirmation before copying any file.

DEVICE NAMES

As well as files, COPY and certain other commands make use of the internal (built-in) device names. The most useful are as follows:

CON The console, i.e. input from the keyboard and output to the screen.

PRN The current printer

NUL Nothing, nowhere (more useful than you think!)

AUX Serial port no. 1

Let's use them:

Make sure you are in the root directory and try:

```
COPY A:\NEWFILE.TXT CON
```

This works in the same way as the TYPE command except that at the end DOS displays the message "1 file(s) copied".

```
COPY A:\NEWFILE.TXT NUL
```

All we get is the message "1 file(s) copied". The copy has disappeared, the original is intact.

```
COPY A:\NEWFILE.TXT PRN
```

A very simple, effective way of printing ASCII text files!

Copying from CON to a file is the simplest way of creating a text file. Enter the following:

```
COPY CON A:\MYFILE.TXT
```

All that appears to happen is that the cursor appears under the command. The computer is waiting to accept input from the CON (keyboard), so type a message. If your mind has gone blank, try the following:

This is my first textfile creation in DOS
written by (your name) on (today's date).

To close the file press the F6 function key, which shows on the screen as ^Z. Pressing Ctrl-Z also works.

Once again, the "1 file(s) copied" message appears and you get the prompt back. If you do a DIR, you will find your new file has appeared in the listing. TYPE it.

To end this section, we introduce two more commands - TIME and DATE. These do not work on files, so may safely be called from any directory. Try them. When prompted for a new date or time, just press enter. Please don't change the settings unless they are obviously wrong.

SYSTEM FILES

In order for DOS to boot, there must be three files present on any **SYSTEM DISK**. A system disk is simply a disk that will load DOS. The files are:

IBMBIO.COM (or IBMBIO.SYS or IO.SYS)
IBMDOS.COM (or IBMDOS.SYS or DOS.SYS)

These two files constitute what is known as the DOS KERNEL and deal with the input / output (to disk, screens, from keyboards, etc.), and communications with the CPU respectively. They are normally "hidden", which means that although they are present, they cannot be listed with a DIR command. In addition, they are read-only, so that you cannot accidentally delete or over-write them. On your practice disk, I have deliberately revealed them.

COMMAND.COM

This is the command shell or command processor, which carries out the internal commands (explained later). It is not hidden, or read-only (by default), and can be replaced by alternatives, although only one, 4DOS.COM, is in common use.

If any of these files are ever deleted, you will not be able to boot from the disk, which can be disastrous in the case of a hard disk!

INTERNAL AND EXTERNAL COMMANDS

All the commands we have used so far have been INTERNAL. This means they are contained in COMMAND.COM and are therefore always available to us. Other commands are EXTERNAL. They exist as a file that must be called from the command line in the same way as any other file. One such command is CHKDSK, which carries out a health check on disks. Try running CHKDSK. You won't succeed - yet!

CONFIG.SYS and AUTOEXEC.BAT

These two files contain the user's own system configuration. CONFIG.SYS usually specifies device drivers to work such devices as a CD-ROM or mouse, while AUTOEXEC.BAT contains other instructions (such as to load a menu). We already have these files in the root directory (which is where they must always be), but they need to be amended to enable us to access the external DOS programs on the hard disk. They are ASCII text files, and I have included on your disks a text editor. Run the program EDITOR.EXE from the DOS directory.

(Of course, I am assuming that by this stage you know how to do this. If necessary I will do a recap on running executable programs).

When prompted for a file by editor, specify A:\CONFIG.SYS. If you specify the wrong file, you will be prompted to specify whether you wish to create the file you specified - say no, and try again. Note in this case, it is *essential* to give a full file path.

You may need to add two lines to the existing CONFIG.SYS. This will vary depending on the setup of the particular computers you are doing your course on. These will be at the beginning of the file. You can access this by pressing a carriage return with the cursor by the very first character of the file. The two lines are:

```
device=c:\sstordrv.sys  
device=c:\devswap.com
```

Although these lines are not case sensitive, there should be no spaces in each line. You might like to check with your tutor before saving your amendments, which is done by holding down the control key whilst tapping K, letting go of the control key and tapping X - normally specified in print as CTRL-KX or ^K-X

Now we must edit A:\AUTOEXEC.BAT in the same way, only this time all we need to do is to amend one line. At the end of the line starting path, add the following after the word newdir with no spaces:

```
;c:\dos
```

note the leading semicolon.

If you've got it right, save the file (^K-X).

Reboot the computer. Do this by holding down the CTRL, ALT and DELETE keys simultaneously. This is known as the three fingered salute.

Let us consider what we have done. We may have added two device drivers to our CONFIG.SYS file. which enable us to access the hard disk. If we haven't, then the hard disk does not need special drivers to access it. We have also added an entry to the path. What is the path and why do we need it?

All the time we were using DOS's internal commands, we didn't have to worry about whether they were available to us in our position in the tree - they were always there. Other DOS commands are external - they are executable programs that we run in the same way as any other program. As we have already found, they are not always immediately available - unless we configure our setup to make them so.

If you give DOS a program to run, it looks first in the current directory (CSD). If it can't find the program there it checks in the directories specified by the Path statement in AUTOEXEC.BAT. Only if it still has no luck finding the specified program does it give up. It therefore makes sense to specify the directories of our most commonly needed programs - the DOS commands - in the path. It is normal to store your DOS commands in a directory called DOS (or MSDOS or DRDOS, or something similar) on your hard disk. In Windows95, the DOS commands are normally stored in the \WINDOWS\COMMAND directory. By adding the c:\dos directory to our path, all the programs in there are always available to us. To prove this, try the following:

We should be in the root directory of A:. Check with a directory listing that there is no program called CHKDSK, with any extension, there. Type CHKDSK.COM and press return. It works!! What DOS has found is the file CHKDSK.COM in the directory C:\DOS, which is in the path, and has executed this. Most of the programs which we will be using from now on will actually be found in the C:\DOS directory.

CHKDSK (spoken as checkdisk) is a very useful diagnostic tool. It tells you how big your disk is and how much of it is used, and also gives the current status of your base memory. If there is anything wrong with the disk, it will find it and report. Those of you who have a home PC with a hard disk are strongly advised to run CHKDSK once a week. The syntax is:

```
CHKDSK [drive:] [/F] [/V]
```

If no drive is specified, the current drive is checked. The /F switch tells CHKDSK to fix any faults it finds (otherwise they are reported but not fixed). The /V switch makes CHKDSK list the files as it checks them. As with many other commands, I have not given the full list of more advanced switches. If you are interested in these, check in any DOS reference book. Note also my syntax: optional parameters and switches are put in square brackets. This is a fairly common convention. Starting with MS-DOS v.6, a more thorough disk checker has been included named SCANDISK. If you are taking this course at a centre where this is available, we will have a look at SCANDISK as well as CHKDSK.

Before we go on, let's consider the implications of having access to more than one drive; in particular the behaviour of CD.

Each drive has a current selected directory (CSD) held in memory, being the directory last accessed on that drive. If we are in A:\ and give the command CD C:\DOS, nothing seems to happen; you certainly don't change to the directory you have specified. However, the CSD of C: HAS been changed. Change drives by giving the command C:, and you will find you are now in C:\DOS.

To recap: you can't change drives with CD, you need to access the drive as a separate command, but you can change the CSD of one drive from another.

THE TREE

We keep talking about the directory tree; now let's have a look at it. Although we will stick with A:\ as our CSD, the tree of A: is very short, and that of C: more interesting, so we'll look at that. The full syntax for the **TREE** command varies with the version, but for DR-DOS 6 or MS-DOS 7 it is:

```
TREE [drive:] [/B] [/F] [/G] [/P]
```

so let's try

```
TREE C:
```

and see what we get. The switches help to make things clearer and are very useful.

/B gives a brief display, omitting the number of files in each directory
/F displays the files in each subdirectory as well as the subdirectory
/G displays the tree in graphical form
/P pauses after each screen in a long TREE display.

TREE is a good utility to practice your use of switches, so take a few minutes trying them out. Don't worry if some of them don't work if you're using MS-DOS 5 or 6.22. You can combine them, such as

```
TREE C: /F /P
```

Shortening paths and filespecs.

So far, you have been encouraged to specify a file by its full path name. This is NEVER wrong. However, it is not always necessary. The rules to go by are as follows; we are assuming the CSD to be A:\

- (1) If full path specified, no problems (i.e. A:\DOS\EDITOR.EXE)
- (2) If no drive specified, current drive assumed (i.e. \DOS\EDITOR.EXE)
note the leading backslash
- (3) If no path specified, CSD assumed (i.e. EDITOR.EXE), which would not necessarily work since in our present example EDITOR.EXE is not in the CSD. However, the file will be found because it is in one of the directories in the path.
- (4) If the command is an executable file, the extension can normally be omitted.

. and ..

If you know what you are doing the . and .. facility can be a quick way of referring to files or navigating your disk. Examples:

CD .. means change to the parent directory

COPY . .. means copy everything from the current directory to the parent directory.

Beware! DEL . means delete everything in the current directory.

Try moving about your disk by means of the .. facility.

MORE EXTERNAL COMMANDS.

Remember that to use these commands, you must either specify the full path name of the file, or make sure the directory is in the path specified in AUTOEXEC.BAT

There are over 80 command files in a modern DOS system, many of which you will never need. This is just a selection of the more useful ones, in alphabetical order.

ATTRIB

This changes a file's attributes. The attributes of a file include R (read only), S (system), H (hidden) and A (archive). There are others. We talk of an attribute being set or unset, i.e. a file with the R attribute set becomes read only. The syntax of this very useful command is:

```
ATTRIB [+|-R] [+|-S] [+|-H] [+|-A] filename(s)
```

Examples:

```
ATTRIB +R C:\COMMAND.COM
```

makes the file COMMAND.COM in the root directory of C: read only (not a bad thing to do!)

```
ATTRIB A:\*.*
```

will display the attributes of all the files in A:\. This is a good way to find out what the hidden files are, and to unhide them if you wish.

The S attribute is only used on certain system files, to make them even more "undeleteable", so don't mess!

The A attribute, the so-called "archive bit" is used by many backup programs to decide whether or not a certain file should be backed up. When a file is created, its A attribute is always set. After backing up, the backup software unsets all A attributes. Thus the next time a backup is done, only the files with A set (i.e. the new or changed ones) will need backing up.

By all means check the attributes of the files on the hard disk, but please don't change them.

COMMAND and EXIT

We have already come across the command processor, COMMAND.COM. This has the very necessary ability to be multilayered, i.e. to load a second copy of itself. This ability is used by certain programs (such as MS Works) to offer a facility known as "Shelling to DOS" whereby a secondary copy of COMMAND.COM is loaded on top of Works itself. When the DOS work has been completed, you can type EXIT and return to Works. The limitation is the amount of available memory.

Try this by continuously loading COMMAND until you get an "out of memory" error. You may well load over 30 shells before this happens. You can, if you wish, then EXIT these shells one by one until you get to your original, but rebooting will cause no harm and may be quicker.

DISKCOPY

This will make an EXACT copy of a floppy disk to another of the same sort. It works in a very different way from COPY, copying the disk structure as well as the files, including any system settings.

Syntax: DISKCOPY drive1: drive2:

where drive1: is the source and drive2: the destination.

Since very few computers have two drives of the same type, DOS very sensibly allows for the command:

DISKCOPY A: A:

in which case it will prompt you to insert source and destination disks as required - it is up to you not to get them mixed up.

If you have a spare floppy, try making a copy of your practical disk by means of DISKCOPY. Your new disk should boot in the same way as the original.

This is how I prepared the disks you are using on this course.

FORMAT

FORMAT has two uses, one of which you will probably use, and the other you won't. Remember that potentially it is the most dangerous of all DOS commands, so use with care.

If you use FORMAT on a hard disk, it will destroy any information already on your disk. DR-DOS has a switch that you must use when specifying a hard disk format, which gives some protection, but most versions of MS-DOS lack this. However, we are only interested in using FORMAT with floppy disks.

All disks must be formatted before they can be used. This involves marking down the tracks and sectors where the data will be stored, and creating the boot sector and FAT. FORMAT has a complex syntax in order to cope with a great variety of non-standard disks, and the available switches vary from version to version. If you need to format floppies at home, I suggest that you find out what switches you need and write a batch file to incorporate them. The general syntax for FORMAT is:

```
FORMAT drive: [/T:tracks] [/N: sectors] [/F:size] [/1|4|8] [/S] [/V:label]
```

Usually, all you need to specify are the drive and the size. If the size of disk is the default size for the drive (1.44Mb for 3.5" drives), even the size can be omitted. In our case, we use either 720K or 1.44Mb disks in A:, and thus the format is either:

```
FORMAT A: /F:720 or
```

```
FORMAT A: /F:144
```

in the second case, simply FORMAT A: will work.

Instructions will be given by DOS.

If you wish the disk to be a SYSTEM DISK, i.e. one which will boot the computer, you will need to add the /S switch.

KEYBUK (v.3.2 and below) and **KEYB** (v.3.3 and above)

These two commands load the keyboard program that makes the characters displayed on the screen the same as those marked on the keyboard. They are normally loaded in AUTOEXEC.BAT (discussed later), the line for the UK using KEYB usually being:

```
C:\DOS\KEYB UK,,C:\DOS\KEYBOARD.SYS
```

and yes, there are two commas in the middle.

If you find that your " comes out as a @ and vice versa, you probably haven't got KEYB loaded correctly.

LABEL

Syntax: LABEL drive: label

All DOS disks, be they hard or floppy, have the ability to take what is called a Volume Label to identify them. Unlike many other operating systems, DOS does not make use of volume labels, so this is a totally optional feature. If you wish to try it, remember that a volume label can consist of up to 11 characters, alpha-numeric, with no spaces. Try altering the label on your practice disk.

MEM

Only available under versions 5 onwards

The MEM command gives information on the available memory on your system. It is one of those commands which have several switches of limited use to the average user, so try the command (DR-DOS only)

MEM /A /P

which shows All information and Pauses every screenful.

PATH

was discussed when we changed our AUTOEXEC.BAT, where it is usually to be found. It tells DOS which directories to look in to find a command file (.COM, .EXE or .BAT) which can't be found in the current directory. Typing PATH with no parameters will display the current search path.

PRINT

is used to print text files, and is an alternative to the COPY . . .PRN technique. Whereas COPY works in the foreground, and does not return control to you until the file is printed, PRINT works in the background leaving your computer free to carrying on working as normal. This can be useful when printing a large ASCII file. However, there is a drawback. PRINT is what is known as a TSR program (Terminate and Stay Resident). In other words, when it has finished, it doesn't go away. Instead, it remains in memory, reducing the amount free for other programs. The syntax is simple:

PRINT filename

This works provided you have a printer attached to the standard port, LPT1. There are switches to accommodate non-standard set-ups. Try the following one after the other

```
MEM  
PRINT A:\CONFIG.SYS  
MEM
```

Make sure you are attached to the printer before printing. Now note that there is less memory available after you have printed than before. For this reason, I do not recommend the use of PRINT, preferring COPYing a file to the printer or using a third party utility.

SYS

This command places a system on a disk to make it bootable. Unfortunately, attempts to SYS a floppy under older versions of MS-DOS will result in the message "No room" even if the disk appears almost empty. This is due to the way these versions work. DR-DOS and later versions of MS-DOS have no such drawbacks, but I would still recommend transferring the Operating System when formatting with the /S switch.

XCOPY

This is an external command which expands on the capabilities of the internal COPY command. With XCOPY you can copy complete directory structures, including subdirectories, subject to their being enough room on the destination. The usual syntax is:

```
XCOPY source_files destination_files /S /E /H
```

where /S means including subdirectories, and /E forcing the creation of a subdirectory even if it ends up being empty. The /H forces hidden and system files to be included in the XCOPY.

REDIRECTION, INDIRECTION AND PIPING

DOS has the ability to divert input and output from the standard devices (the keyboard and the screen), and also to "pipe" or filter output through another file or device.

Piping (|)

The most important use of piping is in conjunction with the utility file MORE.COM. Output piped through MORE is displayed one screenful at a time. In some cases, the use of MORE has been superseded by the /P switch, but it still remains useful. By way of test, try the command :

```
MEM /A |MORE
```

| is known as the pipe, and |MORE instructs DOS to pipe, or process, output through MORE before displaying it.

Two other piping utilities are FIND and SORT. Try:

```
TYPE A:\CONFIG.SYS | FIND "Device"
```

which will only display the lines with the word Device in them, and

```
TYPE A:\CONFIG.SYS | SORT
```

which will sort the lines into alphabetic order before displaying

Redirection (>)

Redirection tells DOS to send the output elsewhere than the screen. Consider, again, the case of the MEM command. We may wish to make a permanent record of its output. All we need to do is to tell DOS the filename we want the redirection to go to:

```
MEM /A > A:\MEMORY.TXT
```

Once you've worked out what will happen, try:

```
TYPE A:\MEMORY.TXT  
TYPE A:\MEMORY.TXT |MORE  
COPY A:\MEMORY.TXT PRN
```

The last of which will give the same printout as:

```
MEM /A > PRN
```

Redirection is a good way to obtain a printout of a directory:

```
DIR A:\ > PRN
```

Instead of being displayed on screen, the directory will be printed.

Indirection (<)

This is a very seldom-used facility. It is used when you wish DOS to take its instructions from somewhere other than the keyboard. Sometimes, it is used to auto-answer "Are you sure" type requests, but this is dangerous:

```
DEL A:\*. * < ECHO "Y^M"
```

Note this is NOT in Bookman Old Style - Don't try it!!

Just to recap: > redirection, < indirection, | pipe.

SIMPLE BATCH FILES.

A batch file (i.e. a file with the extension .BAT) is a list of instructions in text form (i.e. they can be TYPed), which DOS will carry out one by one. Complex batch files can be written by those who are experienced, but as a start, let's write a batch file to carry out an operation which is not catered for in many versions of DOS - to move a file from one directory to another. We shall call it MOVEIT.BAT

There are many ways to create a batch file. For very short files, the COPY CON filename facility suffices. Normally, however, a text editor is used. A text editor is a simple word processing facility which produces output in text, or ASCII format. Most word processing programs have the facility to save in ASCII format and can thus be used to produce batch files if this is your preference. Prior to Version 5, the only text editor included with MS-DOS was the powerful but hard to use EDLIN.COM. New versions of MS-DOS have a program called simply EDIT.COM, which is much better. DR-DOS has a facility called EDITOR.EXE and it is this which we will use to create our batch file.

What is it we want this batch file to do?

Firstly, we wish it to copy a file from its current directory to that to which we wish it to be moved. Secondly, we wish to delete the original. Thus the file has been moved from one place to another.

The copy part can be covered by the generic line

```
COPY file1 file2 /V
```

we verify the copy as the original is about to be deleted by the line:

```
DEL file 1
```

How do we show file1 and file2 in a batch file? After all, we don't know which files the file is to be used on, and we want it to cover all possibilities! The answer lies in REPLACEMENT VARIABLES. These take the form %1 to %9, where %1 is the first parameter on the command, %2 the second and so forth. To give an example if this, suppose we wrote a one line batch file called TYPEIT.BAT:

```
TYPE %1
```

If we gave the command TYPEIT C:\AUTOEXEC.BAT, C:

AUTOEXEC.BAT would be interpreted as the %1 in the batch file, and so would thus be typed.

Coming back to our current batch file, we can now work out the core of the file:

```
COPY %1 %2 /V  
DEL %1
```

This requires two parameters on the command line, and would work as it stands, but it isn't very user-friendly. Time to introduce two batch file commands - REM and ECHO, as well as our old friend CLS.

REM means REMark, and instructs DOS to ignore the rest of the line. It is used for putting in comments meant for us mere humans to help us understand what is going on. Normally, ECHO means type whatever follows on this line to the screen, but has two special extras: ECHO OFF means for DOS not to echo the commands as they happen ECHO ON means the opposite. Let's look at a full version of MOVEIT.BAT

```
REM MOVEIT.BAT  
REM Written by Bill Hayles  
REM On 24th September 1992  
ECHO OFF  
CLS  
COPY %1 %2 /V  
DEL %1  
ECHO One file moved successfully.
```

The example above is far from perfect: you might like to tidy it up a bit yourself.

You will recall that EDITOR.EXE is in the DOS subdirectory of A: I hope you will agree that this is a good place to put MOVEIT.BAT. When EDITOR asks you for the name of the file, enter A:\DOS\MOVEIT.BAT. When it says that the file can't be found and asks you if you wish to create a new file, answer yes! Enter the text of your version of MOVEIT.BAT, and when it is correct, save it by typing CTRL-K X.

Try out your new file by using it to move the file NEWFILE.TXT from the root directory to the DOS subdirectory. Check it has worked with a DIR of the appropriate directories. Everything OK? Congratulations, programmer!

CONFIG.SYS and AUTOEXEC.BAT

These two very important files configure your computer to your own particular requirements. These will be found in the root directory of your system drive. CONFIG.SYS is where device drivers are loaded; indeed it is the ONLY file which will do this on most systems. AUTOEXEC.BAT is a standard batch file, capable of accepting all batch file commands, which is AUTOMATICALLY EXECUTED on boot. Its use is normally limited to loading a few TSR (terminate and stay resident) programs, setting system any environmental variables (such as prompt), and maybe loading a menuing system. There is no such thing as a standard CONFIG.SYS or AUTOEXEC.BAT, and it is this very flexibility which makes the PC unique. However, you can expect to see lines similar to many of the following in **CONFIG.SYS**

DEVICE=ANSI.SYS	Loads a screen colour handler
DEVICE=MOUSE.SYS	Loads a mouse driver
DEVICE=RAMDRIVE.SYS	Loads a virtual (RAM) drive
DEVICE=SMARTDRV.SYS	Loads a disk cache
FILES=20	Defines how many files can be open
BUFFERS=15	Defines how many disk buffers
COUNTRY=044,,C:\DOS\COUNTRY.SYS	Defines the country defaults
LASTDRIVE=Z	Defines how many available drives
SHELL=C:\COMMAND.COM	Defines the command interpreter

and in **AUTOEXEC.BAT**

KEYB UK+	Defines the keyboard layout
SET TEMP=C:\] Set environmental variables
SET CPTMP=D:\	
SET COMSPEC=C:\COMMAND.COM	
PROMPT \$P\$G	Define the command prompt
PATH C:\;C:\DOS;C:\PCTOOLS	Define the search path
CD \MENU	Change to the menu subdirectory
MENU	And load the menu program.

Some of these settings need explaining further.

FILES defines how many files may be open at one time. Some programs require access to a considerable number of files at the same time, so this setting is often set at 20 or 30 to accommodate them. However, the greater the setting for FILES, the less the memory available for applications.

BUFFERS defines the disk buffers, a primitive way of speeding up disk access times. Nowadays, a disk cache is often used, which can enable BUFFERS to be set low. Once again, the higher the setting, the more memory is used.

SHELL defines the command shell. If this line is missing, the default is COMMAND.COM in the root directory. Alter this line at your peril!

The **COUNTRY** setting defines how DOS displays such things as dates and times (i.e. to an American 09/12/92 is 12th September, whereas to us it is 9th December).

The **ENVIRONMENT** is a small area of memory where variables or labels are stored. These variables are usually defined by SET statements, and have many uses. Consider the CPTEMP setting. When the PCTools program is running, it creates temporary files on disk. It looks in the environment for the CPTEMP variable to decide where to store them. The environment is also where the command prompt is defined. Most people are happy with the \$P\$G setting which is what we have used on this course. However, long and flash prompts can be defined with a knowledge of ANSI. The PATH, which defines where DOS will search for an executable file if it can't find it in the current directory, is also stored in the environment.

LOCKOUTS.

It is very easy, when changing CONFIG.SYS and AUTOEXEC.BAT, to find yourself locked out of your computer, and in a vicious circle. The configuration is wrong, so the computer won't boot. You can't reconfigure the system without booting the computer. The answer is to create a system floppy with enough drivers to boot your computer, and also containing an editor, and to keep it handy when changing your configuration

Many software packages come with an installation routine, designed to make your life easier. While these are very worthy, some change your setup in peculiar and unwanted ways. A common example is the package assuming it is the only one running on the computer, and changing AUTOEXEC.BAT to load itself automatically. If possible, do not let the installation routine change your setup. Instead, make a note of the changes proposed, and make the necessary ones manually.

FINALE

You will be allowed to examine the setup on the computers at your centre. This will be explained to you by your tutor.

This brings us to the end of our short course on DOS. You should now be able to perform your own everyday DOS housekeeping. It will be worth your while to study your own DOS manuals to become fully conversant with the commands, parameters and switches available to you. A working knowledge of batch files can prove rewarding. Those of you with 386 or newer computers should look into the memory management tools available for your system, the different types of memory, disk caches and RAMdrives. It is always a good idea to try out these ideas on a practice system floppy before using them on your main hard disk configuration!

An occasional follow-up to this course, on batch files and memory management, is held at Crayford Manor House. Please let your tutor know if you wish to be considered for the next one. If you are taking the longer (14 week) course for the RSA File Management Certificate, much of the batch files course will be covered.

APPENDIX

Listing of the files in a typical DOS directory, captured by the command
DIR /W > PRN

Several of these files are discussed in the text

```
Volume in drive C is CEEDRIVE          Serial number is 1C83:48D2
Directory of  c:\dos\*. *

[.]                [..]                [TIPS]            1050.cpi           4201.cpi
4208.cpi           5202.cpi           ansi.sys          append.exe         assign.com
atfloppy.exe      attrib.exe         backup.com        bnw.com            check.bat
chkdsk.com        command.com       comp.com          config.sys         cursor.exe
danger.com        debug.com         delpurge.exe     delwatch.exe      devswap.com
dir.txt           diskcomp.com      diskcopy.com     diskmap.exe       diskopt.exe
display.sys       dosbook.exe       drcommand.com    drdos.img          drdos.ini
driver.sys        editor.exe        ega.cpi          emm386.sys         emmxma.sys
exe2bin.exe       fastopen.exe     fc.exe           fdisk.com          filelink.exe
find.exe          format.com        graftabl.com     graphics.com       help.bat
hidos.sys         join.exe          keyb.com         label.com          lock.err
lock.exe          lock.txt          login.err         login.exe          login.txt
mem.exe           memmax.exe        mode.com         move.exe           nlsfunc.exe
old.img           password.exe      pckwik.sys       pckwin.sys        printer.sys
ramdrive.sys      rendir.exe        restore.com       script.exe         setup.exe
share.exe         sid.exe           sort.exe          ssem386.sys        sshidos.sys
sstor.exe         sstordrv.sys     subst.exe        superpck.exe       superpck.sb1
superpck.sb6     superpck.sb7     superpck.sb8    sys.com            taskmax.exe
taskmax.ini       touch.exe         tree.com          udcat9.ega        udcca9.cga
udhrc9.ega        uduni9.vga       undelete.exe     unformat.com       uninstal.exe
viewhelp.rsc      viewhi.icn       viewlo.icn       viewmax            viewmax.acc
viewmax.cfg       viewmax.drv      viewmax.exe      viewmax.ini       viewmax.ovl
viewmax.rsc       viewrun.rsc      xcopy.exe        xdel.exe          xdir.exe

    2,546,085 bytes in 115 file(s)          2,670,592 bytes allocated
    20,840,448 bytes free
```

ADDITIONAL NOTES FOR THOSE TAKING THE RSA CLAIT FILE
MANAGEMENT: DOS ASSESSMENT.

The RSA requires additional knowledge about data backup and protection, disk storage and protection from the elements and other hazards. This will take the form of theory questions. It is hoped that these notes will assist you in answering these questions. The references are to RSA Assessment Objectives: In this section, what you need to know is in Arial font, and notes as to the answers in **Letter Gothic font**.

1.1 You will need to know about the different sizes and densities of floppy disks, CD-ROMs, digital tapes and how they can be used.

Floppy disks come in two sizes, 5.25" and 3.5", and in double density and high density forms. The capacity of 5.25"HD disk is 360k, 5.25"DD is 1.2Mb, 3.5"DD is 720k and 3.5"HD is 1.44Mb. CD-ROMs are read-only devices taking a standard 5" Data Compact disk. Digital Tape is used in special drives, not normally allocated drive letters, mainly for backup purposes.

1.3 You will need to know about the potential hazards to disks and the data contained on them.

Disks need to be protected from extremes of temperature and humidity, and should never be allowed to get wet. The data on them can be affected by magnetic fields such as those generated by loudspeakers, telephones, computer monitors or even the motors of an electric train. They should be protected from exposure to dust by being kept in a sleeve or jacket. The surface of the disk itself should NEVER be touched.

Therefore, ideally, disks should be kept covered in a temperature controlled environment.

1.4 You should be familiar with the need to back up data, and how to store backups to ensure their safety. You should know what can cause loss of data on a PC.

Data can be lost by system failure. In other words if a hard disk fails and you haven't backed up the data on it, that data is irrecoverable. Although a less likely cause, if good housekeeping is not practised, computer viruses can corrupt disks.

Therefore, before an unknown floppy is introduced to a computer, it should be checked for viruses, preferably on a computer kept specially for the purpose.

In order to minimise the risk to data from theft, backups should be stored in a secure, locked container. Important data being carried on the person should be protected in the same way as a valuable document. Furthermore, at least one backup set should be stored at a remote site to guard against loss from natural disasters such as flood or fire. Ideally, three backup sets should be maintained.

1.5. You will need to know how to write protect a floppy disk.

On a 5.25" disk, a paper tab is used to cover a notch on the right hand side of the disk (looking from the front). On a 3.5" disk, again looking from the front, there is a square hole at the top on the right which can be exposed or covered by moving a sliding tab sighted on the back of the disk. When the hole is exposed, the disk is write protected and cannot be written to or altered in any way. When the hole is covered the disk can be both read from and written to. The second, permanently exposed hole on LHS of a 1.44Mb disk is merely to convey its size to the computer.

1.8 You will need to know why, and in general terms how, disks are checked for viruses.

A virus is a piece of computer code designed to be transmitted from computer to computer, with a malevolent intent. A virus can change data on a disk, affect the computer's operation, or even destroy the data completely. A virus can be transmitted to a computer in one of three ways: by means of a removable disk (floppy or CD-ROM), from another computer by way of a network, or from the Internet or other on-line communication system. The RSA are only concerned with floppies. In an environment where it is feasible, a dedicated, virus checking "footbath" computer is used to check all floppies before they are allowed onto other machines. Where this is not available, reliable virus checking software should be carried and activated on all machines.

1.9 You will need to understand and act on various error messages.

This will have been covered on the course, but just to recap, the commonest causes or error messages are:

Non-existent files or directories

Incorrect typing or attempts to use illegal names or characters

Directories cannot be removed if they contain files

Floppy not in drive

Disk write protected

2.1 You will need to know what is, and what is not, a legal filename.

This is mentioned because the RSA have got it wrong! You are expected to respond to any question about filenames to the effect that the filename can have up to 8 characters (correct), but that certain characters are illegal (correct) They also state there must be a 3 character extension preceded by a dot (wrong!) In fact, a filename extension can be anything from 1 to 3 characters or even not exist at all. At the time of writing (December 1996), your author is taking this up with the RSA.

GLOSSARY

The following is a list of the main terms that you are expected to know. If you are uncertain as to the meaning of any term, then ask!

Applications Software	Option
Backslash	PC Compatible
Batch File	Pathname
Boot	Prompt
CD-ROM	Root Directory
CPU	Stand alone computer
Computing System	Sub-directory
Cursor	Tree
Data	Virus check
Directory	Volume label
Disk	Wildcard characters
Disk Drive	Write Protect
Diskette	
Display	
DOS	
Drive Reference	
Error Message	
File	
File Formats	
Filename	
File Protection	
Floppy Disk	
Format	
Hard Disk	
High Density	
Low Density	
MB	
MS-DOS	
Megabyte	
Operating System	

ASSESSMENT OBJECTIVES.

This is what you will be expected to do or know about:

- 1.1 Select appropriate disk
- 1.2 Format floppy disk and label disk volume
- 1.3 Handle / store disks in a safe manner
- 1.4 Store backup data in a safe manner
- 1.5 Write protect disks
- 1.6 Create a system floppy disk
- 1.7 Select and change disk drives
- 1.8 Check integrity of data
- 1.9 Recognise error and system messages
- 2.1 Naming files
- 2.2 Using wildcards
- 2.3 Setting file attributes
- 2.4 Checking and setting the system date
- 2.5 Checking and setting the system time
- 2.6 Copying a file
- 2.7 Copying a group of files
- 2.8 Deleting a file
- 2.9 Deleting a group of files
- 2.10 Renaming a file
- 3.1 Create directories / subdirectories
- 3.2 Delete directories / subdirectories
- 3.3 Change directories
- 3.4 Change to root directory
- 3.5 Display system tree structure
- 3.6 List all the files in a directory
- 3.7 List all the files one page at a time
- 3.8 List all the files across the screen
- 3.9 Clear the screen display.