THE MICROCOMPUTER OS

This module will take a look at how an operating system works with hardware to manage resources. By using the IBM PC design and Microsoft OSs as a foundation for the study, the module includes exercises for students to complete, reinforce the knowledge proposed during the module.

File System. Storage devices are an important part of the continuing operation and recordings of an operating system. A key component of general purpose operating systems is the management of secondary storage devices, and the allocation of storage units (commonly referred to as files.)

Memory Management. When several software instructions are being processed, one of the tasks for the operating system is to keep track of the free space in the computer's memory and to make sure that no application task corrupts another application's memory area.

Process Management. A key attribute, function, of operating systems is to organise the use of CPU time by other programs. This function includes loading the program into memory, allocating it time and then executing the instructions in the program by passing it to the CPU.

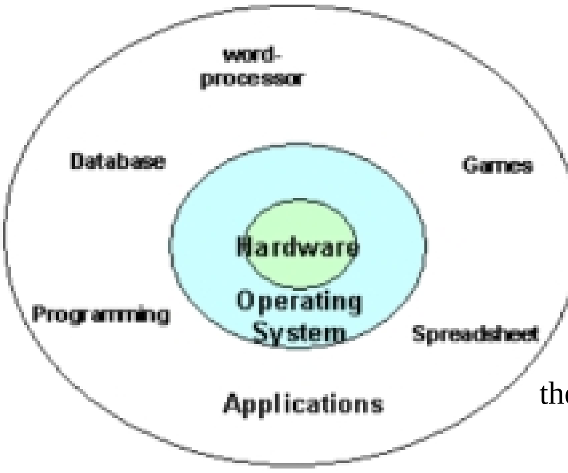
Communicating with devices / peripherals. Some of the most complex tasks performed by a computer involve communicating with computer monitors, printers, disk drives, and other peripheral devices. A computer's operating system generally includes programs to handle these tasks for appli-

This module begins by taking a look at what the computer user can immediately recognise, from turning the computer on, and using it. A slightly technical description towards the end covers operating system activities that are not immediately clear to the computer user.

1. Turning on the PC
2. The User Interface to the OS
3. File System
   - Organising Secondary Storage
   - Locating Files
   - Preparing a new storage medium
   - Managing Files

*. Memory Management
*. Process Management
*. Peripherals – Printers

The order in which this module has been organised does not indicate any priority in how an Operating System, nor Hardware will behave.



This Onion Skin diagram shows one relationship between the Operating System and other Computer System components. The operating system sits between the Hardware and applications, manages most communications between the Hardware and Applications.

Ancillary objectives of this module is for students skills to be developed in the concepts of OS implementation that allows for students to readily transfer skills to other OSs such as the Macintosh OS, Unix, X Windows Systems.

TURNING ON THE PC

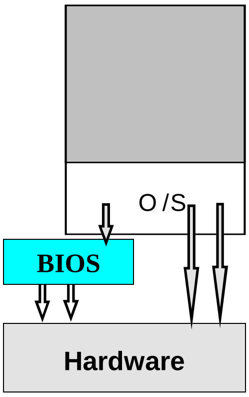
When the microcomputer is turned on, there is no active operating system to manage the hardware, so one of the initial tasks of the microcomputer is to find an operating system and let it take control of the microcomputer.

STARTING THE ENGINES

When you turn on an IBM PC Compatible microcomputer for the first time, the first thing that happens is that a piece of hardware wired to take control of the machine. For the IBM PC design, this hardware was called the BIOS. Microcomputer designs work in a similar manner, although the capabilities and things the other designs have for their 'BIOS' may differ.

BIOS - BASIC INPUT / OUTPUT SYSTEM

When IBM designed their first Personal Computer, the designers built a ROM chip with hardware control instructions to help the OS programmers and to define the minimal things the PC will perform before talking with an OS. The BIOS contains programs in a ROM chip written to provide a preset list of instructions to deal specifically with Input/Output. Because the BIOS is in a ROM chip, the program instructions are not lost when the computer is turned off, and it can be wired to automatically run its instructions when power is turned on.



Microcomputers such as Apple’s Macintosh have far more sophisticated software controls built into the hardware. The Macintosh “BIOS”, known as the Mac Toolbox, includes instructions for graphics and including the use of a mouse is so application programmers have more “tools” for writing their programs.

The BIOS on PCs are now programmed and built by more than just IBM. Companies who supply BIOS ROM include Phoenix, Award, and AMI.

The startup ROM, or firmware, for Macintosh Microcomputers is often called the Mac Toolbox. The Alpha Microcomputer calls its firmware MILO for '**MI**ni**LO**ader

The PC BIOS performs at least two important tasks; provide and initial check for computer hardware [Power on self test], and finds the OS to use on the computer.

Power on Self Test (POST)

One of the first things the BIOS performs is to take an inventory and check the list of hardware that is connected to the machine, if it recognises what it finds, then it ticks it off as something that the OS and other programs can use.

An example of this recognition system is that some video cassette players will control the TV, while others cannot.

If the BIOS does not recognise a piece of hardware connected to the computer in most cases it will ignore it, hoping the OS knows what to do with it.

Checking for the Operating System

A second thing the BIOS performs is to check the storage devices it recognises to see whether they contain an Operating System it can transfer control of the computer to. More specifically the BIOS has a list of storage devices it will look through. Most PCs can check the 1st floppy drive, the 1st hard disk and in some cases the CD-ROM drive.

The IBM PC is not the only PC design available on the market. Other microcomputers with their own distinct advantages include the Apple Macintosh design, the Compaq Alpha design.

Floppy Disks - Looking for the OS

Most BIOSs are configured to check the floppy diskette first for an operating system (a feature supporting the original designs of PCs having only floppy diskettes.)

PCs look for a storage device that contains a special section called a System Boot Sector which is supposed to contain a program that will take care of starting the operating system.

System Boot Sector (SBS)

The BIOS checks the floppy disk for a System Boot Sector (SBS) and if one exists it will load the program loader inside the SBS and transfer control to it. In general the program loader inside the SBS will load the remainder of the operating system files required to start a microcomputer.

If no System Boot Sector is found, or no program loader can be found, the BIOS will generally send a message to the screen to inform the user that an error has occurred. A common message may look like:-

```
Non system-disk or disk error.
Replace and strike any key when ready.
```

MS-DOS

An **MS-DOS start-up diskette** will contain as a minimum.

COMMAND.COM	- the transient / command processor
MSDOS.SYS	- the base system
IO.SYS	- input / output system program routines

Where msdos.sys, and io.sys are usually hidden files that is generally not seen when using the DIR command to list files.

MS-DOS. The bios finds the SBS, loads io.sys and msdos.sys, which then load command.com to give your MS-DOS.

Windows 95/98

A **Windows 95/98 start-up diskette** will contain as a minimum the following files.

COMMAND.COM	- the transient / command processor
MSDOS.SYS	- a text file with initialisation information
IO.SYS	- input / output system program routines
DRVSPACE.BIN	- system program routines to support special hard disk configurations.

Of course, the Windows 95/98 startup diskette does not start up the full Windows operating system but only a character based minimal system commonly used for maintenance or for older MS-DOS programs that cannot run inside the newer operating system.

The MS-DOS command.com *transient* program can dump parts of itself from memory if RAM is short for another program, and reload itself after the other program no longer needs that RAM

WIN9x. The bios finds the SBS, loads io.sys. io.sys loads drvspace.bin incase a strange hard disk exists, checks msdos.sys for settings and loads command.com

Other OSs for PCs ?

Operating systems for PCs such as Windows NT and Linux follow a similar procedure as shown above. The bios finds the loader program (such as msdos.sys or io.sys) which then completes the OS startup initialisation.

For a floppy disk start-up what we get on the screen is usually just a text screen for typing in keyboard commands.

HARD DISKS - LOOKING FOR AN OS

PC hard disks have evolved beyond the secondary storage designs in the original BIOS. To cater for continuing changes in Hard Disk design, the bios uses a new section on the hard disk for finding the System Boot Sector.

Master Boot Record

The Master Boot Sector (MBS) also known as the Master Boot Record (MBR) was created to contain at least the following information.

- Disk Configuration, Layout, Partition information
- Location of the System Boot Sector

When the BIOS looks for the operating system on the hard disk it loads the information in the Master Boot Record so it can understand how things are stored on the hard disk (as well as other things) then it locates the System Boot Sector where it loads the Operating System loader program into RAM.

The System Boot Sector, as originally designed for the floppy diskette, remains to simplify things.

Operating Systems

The same files listed for a floppy diskette are found on the Hard Disk. Fortunately there is more space on the Hard Disk so more files can be added.

Windows 95/98. The most obvious advantage of the Hard Disk installation is the ability to have Windows 95 start up with the complete Graphical User Interface (GUI) instead of just a text-screen.

For Windows 95/98, after IO.SYS has loaded, interpreted MSDOS.SYS, and loaded command.com it usually starts the Windows 95/98 GUI. From the command-prompt the GUI can be manually started using the `win.com` command.

Voila!

Once the OS loader is in RAM the BIOS transfers control to the loader which completes the loading, executing of the operating system.

If the BIOS does not find a valid Master Boot Record or System Boot Sector to start the operating system then it will give an error message.

```
Non system-disk or disk error.
Replace and strike any key when ready.
```

Otherwise it will give a message such as:

```
Starting Windows 95.
```

Other general purpose computers using an operating system loaded from secondary storage initialise in the same manner.

File System - Organising Secondary Storage

Turning on a computer and starting up the OS we now recognise that organising the secondary storage is an important part of an operating systems responsibility, if not for the user, for its own initiation.

Things related to organising secondary storage include defining and maintaining what has generally been termed a “File System” for storing or ‘filing’ data.

Many operating systems on microcomputers use a hierarchical filing system sometimes analogous to the Filing Cabinet, its folders and files within folders.

Now we have a Filing System – What are Files ?

Technically, a file can contain anything that can be generated for storage, but for computer users files are often categorised into two types of files.

(i) a file with user or otherwise created information (data) or
(ii) a file with computer instructions (programs)

Within these two categories can be very different types of files.

The Operating System is itself made up of different data and program files, so it is very important for the OS to create, manage and control the file system.

Identifying Files – filenames, path

Files are uniquely identified by applying a location and label to it. The label we give files is usually called the *“filename”*. The location of the file within the storage system is often called the *“path”*. The *path*, *filename* combined together specifies a location on the storage device and the name of the file.

File Naming Limitations – MS-DOS

MS-DOS’s File System restricts filenames to contain 8 letters/characters with an “extension” which itself can be made of three letters/characters. This MS-DOS limitation on naming files is also referred to as “8.3”.

— — — — — — — — . — — —

Windows 95 supports Long Filenames (LFN) which allows filenames to be up to 255 characters long, including spaces.

The **Macintosh** Hierarchical File System (**HFS**) supports up to 32 characters including spaces.

Windows **NT**’s NTFS, **OS/2**’s HPFS, and **Linux**’s ext2 support 255 character long filenames.

These operating systems also provide tools for graphically viewing the contents of the hard disk.

More specific rules governing the naming of files in MS-DOS are:

- Can use the alphabet, 0 to 9 and the following special characters:
  No other special characters are acceptable.

A – Z 0 – 9 _ ^ $ ~ ! # % & - { } @ ‘ `	Allowed characters in a filename
“ , \ / : * < > ?	Not Allowed characters

- Spaces are not allowed in MS-DOS, although allowed in Windows 95
- Cannot use filenames already being used by the operating system such as: CLOCK$, CON, AUX, COMn (where n=1-4), LPTn (where n=1-3), NUL, and PRN.

An MS-DOS File – Example of 8.3

MS-DOS Prompt

An example for the 8.3 is the name Samiuela Taufa. This name requires 14 characters, including the space. To fit 8.3 you can manipulate, change the name to something like:

```
STAUFA          SAMTAUFA        SAMT
SAMIUELA.TAU    SAM_TAU.FA      S_TAUFA
```

Of course, in Windows 95 or Macintosh you can just use "Samiuela Taufa's File" which fits in just fine for their naming restrictions.

Determining File Use, Purpose – The File Extension

PC operating systems, MS-DOS and Windows use the file's extension to indicate the purpose or group type of a file. Some conventions used in file names for MS-DOS are:

Windows Explorer

`.BAT, .COM, .EXE` are names for files that contain instructions, programs which MS-DOS can execute/run to perform a task.

MS-DOS Prompt

Extension	Purpose	MS-DOS Program	Windows Program
BAT	**BAT**ch Instructions	command.com	command.com
COM	**COM**mand instructions	command.com	command.com
EXE	**EXE**cutable instructions	command.com	GUI
DOC	**DOC**ument	Microsoft Word,	Microsoft Word
TXT	**T**e**XT**	type	Notepad
TMP	**TeMP**orary file		

Other file names usually have a special purpose, or the user screwed up in writing their filename. Most programs use a set of extension to specify which files they have written and to simplify looking at the files at a later time.

What the operating system does when the user tries to open, use a file is dependent on the 'Shell.'

Can you determine the purpose grouping for the following file extensions?

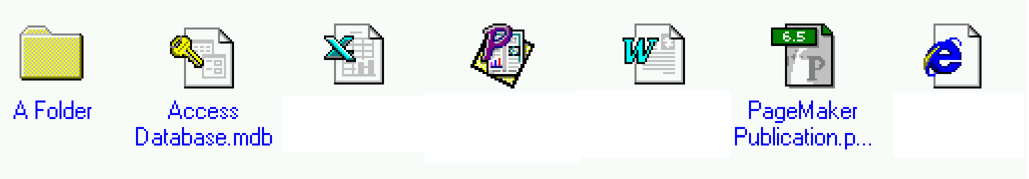
.xlt
.xlc
.rtf
.wri

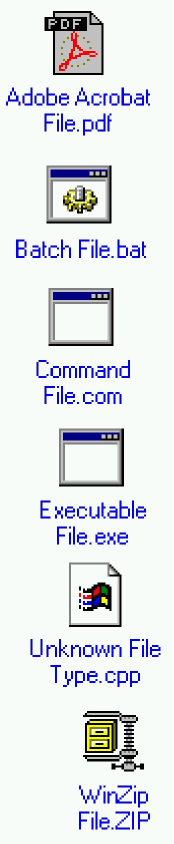
To find out which program uses these files in Microsoft Windows, open Explorer and find one of the files. Double click on the file and find out which program opens it up.

When a computer user types in "*testfile.bat*" at the command prompt, the ms-dos shell checks to make sure the file is actually there. If the file exists, then the shell checks what it is supposed to do with it, in this case a file with the extension testfile.bat is processed by the shell command.com so the file is opened and given to command.com.

The following table lists extensions common on a Windows 95 machine, can you find the Windows program which opens it, and guess at the purpose of the file ?

Extension	Purpose	Windows Program
XLS		
ASC		
MDB		
GIF		
JPG		
BMP		





File Use – GUI File Association

Graphical User Interfaces (GUIs) help the user by giving pictures of how the GUI believes the files are related to the programs known to the operating system. The diagrams on this page are a sample of pictures the Windows GUI displays when looking at files with certain extensions.

The file with a *.doc* extension is shown with a picture of the program that will open the file. Likewise the *.xls* extension and the *.pub* extension are shown with pictures of the programs that will open these files.

File Information – File Size, Date, and Time

As a general maintenance, record keeping matter, the OS will record additional information about a file. Like a good secretary, when MS-DOS stores your files it also records three other bits of information:

1. The size of the file
2. The Date it was created or last modified
3. The Time it was created or last modified.

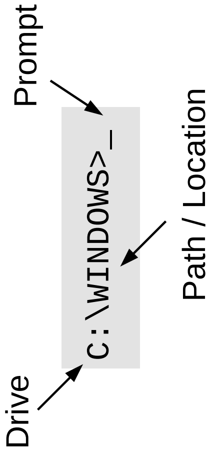
Files and Storage – An OS perspective

So far as the OS is concerned it does not really care about extensions or whether pretty pictures help us relate to what is supposed to be in the file. For that matter the OS does not really care whether the filename is readible or not.

What is important for the operating system is that it can grab the file from secondary storage and make it available in an effective time for the user application, while at the same time minimising the potential for its own files being damaged. Likewise the OS needs to be able to quickly store files onto the hard disk while not interrupting other work being performed on the computer.

The User Interface – Talking to the Computer

The user interface, or method by which the computer recieves instructions and provides feedback to a human user has generally been through the use of the computer screen, and computer keyboard. Providing a means for the computer user to interact with the purpose of the OS uses a program on top of the base operating system functions, or a program that simplifies the complexity of what the OS has to do.



The '*User Interface*' programs are often called *shells*. MS-DOS uses a '*text*' screen to communicate with the user. Microsoft Windows 95 uses a GUI shell called "*Explorer*," and the Macintosh uses a GUI shell called "*Finder*." The MS-DOS '*shell*' is provided by the program `command.com`

Each command for MS-DOS is entered through the keyboard when the computer displays the "*command prompt*". On most MS-DOS machines the command prompt looks like:

`C:\WINDOWS>_` or `C:\>_`

The command prompt defines:

1 What disk drive am I sitting on. *(the storage device)*
2 What directory am I in. *(the location on the storage device)*
3 A blinking "_" cursor to indicate where typing on the keyboard will begin displaying on the screen.

At this prompt you can type in anything you want, and if the operating system understands it as something useful it will do something about it, otherwise it will think your stupid.

```
C:\>I want to play games
Bad command or file name
```

Using the Keyboard – Telling the Computer What You Want

Since the text-based shell uses the keyboard for talking to the computer, a few special keys should be learned at this point.

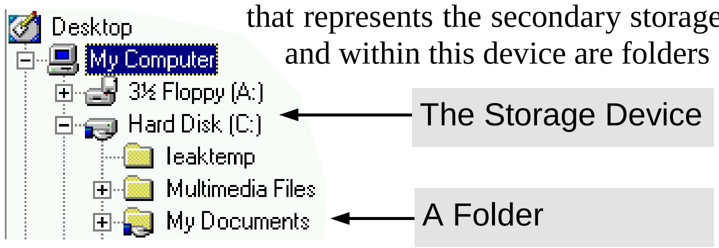
ENTER The shell does not know that you have completed your instruction list until you have hit the "enter" key. So hit the key when you want the computer to do something with what you have typed.

BACKSPACE Correction key. This key is usually found above the **ENTER** key with an arrow pointing to the left. You can use this key to delete the character before the cursor. Continue pressing the **BACKSPACE** key to delete additional characters until there are no characters in front of the cursor.

CTRL+ALT+DEL
Re-starts your computer (executes a warm boot).
When you restart your computer, or execute a "warm boot", all of the data in the conventional memory area of the computer is lost.

File System

As discussed earlier, our Operating Systems use a hierarchical filing system that is something like using a filing cabinet where files are stored in folders. Visually displayed, Windows Explorer and the Macintosh Finder have a topmost 'folder' that represents the secondary storage device (hard disk) where the files are stored and within this device are folders and files.



MS-DOS names the branches of the system, '*directories*' whereas in Windows 95 and the Macintosh names the branches '*folders.*'

ROOT – Going to the very top

At the very top of a storage device is the '*root*' folder. In our diagram going to the root is the same taking the mouse pointer and clicking on the "*Hard Disk (C:)*" picture.

Each storage device provides a '*root*' folder from which other folders can be placed for use by the operating system (and programs using the operating system.) The symbol for the 'root' folder/directory on the storage device hard disk c: is as follows:

C:\ - where C: is the label for the drive, and "\" is the label for the 'root' directory.

Naming that Location, Folders

Each folder, or directory has the same naming convention as for files discussed earlier. To describe the path, or location of a folder, we will take a look at two examples.

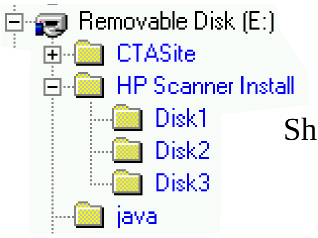
My Documents. The path to the folder "My Documents" is;

- the folder is on the storage device drive C:,
- the first folder is the root folder "\" and
- this folder is at the root.

Short-hand it is written out as:

```
C:\My Documents
```

In our second diagram, we want to write out the path to the folder named "Disk3".



The path to the folder "Disk3" is;

- the folder is on the storage device drive E:,
- the first folder is the root folder "\",
- the folder from the root is "HP Scanner Install"
- Disk3 is a folder inside this folder

Short hand it is written out as:

```
E:\HP Scanner Install\Disk3
```

Files – Looking Around, showing a Listing

Windows Explorer

A fundamental need for computer users is to be able to determine what files exist in a given location (directory or folder.) To show a list of files in a GUI shell we usually take the mouse pointer and double-click on the folder to open the folder.

In a text shell we need to type in a command to list the files. MS-DOS provides a “directory listing” command called “DIR”.

MS-DOS Prompt

```
DIR   DIRectory listing
```

To tell MS-DOS that you want a directory listing, you type “dir” at the MS-DOS prompt (without the “quotes”).

```
C:\WINDOWS>dir
```

Results:

List of files in the order that MS-DOS has it on the disk.

- The file names
- The file size
- The dates the files were created/last modified
- The times the files were created/last modified

When you execute the command DIR by typing it in at the MS-DOS Prompt, the computer will look at the “directory” and put onto the screen a list of files within the “directory” shown on the command prompt.

But the list is too long and scrolls off the screen.

If you have typed in the above command and there were no errors, then a long string of lines will have scrolled up and up until all you have is the last page of something that was very long.

Two special Directories in each folder (except the root) is the single dot “.” and the double dot “..”

The single dot “.” is a folder name that refers to the ‘current’ folder.

The double dot “..” is a folder name that refers to the ‘parent folder’ or the folder in which the current folder is located.

If the current folder is C:\WINDOWS then the folder single dot “.” refers to the WINDOWS folder, and the folder double dot “..” refers to the parent folder, where windows is located which is the “\” root folder.

DIR will list the dot folders “.” and “..” as type <DIR>, directories.

```
C:\WINDOWS>dir /p
```

The /p tells dir to pause the listing after each screenful.

```
C:\WINDOWS>dir /w
```

The /w tells dir to display the listing in ‘wide-screen’ mode. Which really means, leave all the other junk out and put only the list of files into columns so you can see it.

```
C:\WINDOWS>dir [path]
```

The [path] tells dir to give the listing, not of the current directory, but of the [path] specified in the command. For example, while still sitting in drive C: we can force a listing of the files in drive A:’s root directory by typing

```
C:\WINDOWS>dir a:\
```

File Listings – Customisation using the Parameters

The DIR command can use "parameters" (words that follow the command to tell the command extra things to do) to allow the user to customise the result they get.

```
DIR *.EXE
```

By executing the "DIR *.EXE" command we are telling MS-DOS to give us a listing of all files that begin with "*" (which means any letter or group of letters/ characters) and ends with the extension .EXE. The asterix is called a "wildcard".

Files - Taking a Peak at What is Inside

So how can we take a look at what is in a file?

Generally, to view the contents of a file you would use the program that created the file to open the file and display its content.

TYPE. A simplistic method of viewing the contents of a file is to use the MS-DOS command "type". This command "types" the contents of a file onto the screen for viewing.

For example, to view the contents of the file a:john.txt on a disk in drive B:, you would use the following command:

```
type john.txt
```

FOLDERS – MAKING AND REMOVING FOLDERS

Folders, or Directories are easily created in Windows Explorer, by just selecting the File | New | Folder menu command. For the text based MS-DOS shell another command is required for making directories, called MD (**M**ake **D**irectory).

The command is typed in at the MS-DOS prompt, followed by the 8.3 *folder name* that you want to give the directory:

```
MD dirname
```

Example:

Suppose you are in the root directory (\) and you want to create a directory named QUEEN to store you data. We know the command would be: `md QUEEN`

```
C:\>MD QUEEN
```

To check whether things have worked correctly we can type in our dir listing command for our folder.

```
C:\>dir queen
QUEEN       <DIR>         03-07-94  11:36p
```

The directory QUEEN has been successfully created by MS-DOS, as indicated by the results from the **dir** command.

DIRectories - UNcreating them

To remove unwanted folders in Windows Explorer you highlight the folder and then press the delete key on the keyboard. To get rid of, remove unwanted directories **RD** (Remove Directory) or RMDIR (**R**e**M**ove **DIR**ectory). The general format for the command is:

```
RD directory-name
```

To remove the directory QUEEN that was created you would type in:

```
rmdir queen
```

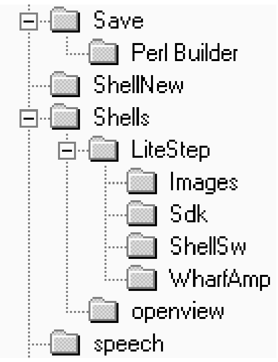
DIRectories - Moving in, out, around

To move around the different folders in Windows Explorer you double-click on folders to open them, or single click to just view a listing of files in that folder. To move into a different folder in MS-DOS we have to change the directory the command current resides using the `CD` - Change Directory or CHDIR (**CH**ange **DIR**ectory) command.

```
CD [path]
```

Changing to the root directory. The root directory is the top of the directory hierarchy for a drive. To return to the root directory, type the following command:

```
cd \
```

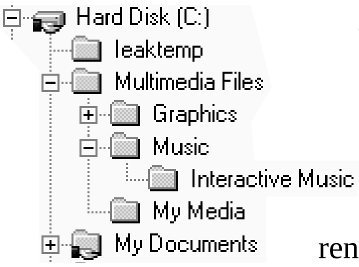


Referring to our diagram, if the current directory is `Shells` and we want to move to the folder `WharfAmp`, then the path we need to move through is,

- move through to the LiteStep folder
- move again through to the WharfAmp folder

```
cd LiteStep\WharfAmp
```

In our second diagram, we are sitting in the `Music` folder and want to go to the `Graphics` folder, which is on the same level we are. The path we need to move through is,



- move back one folder (..)
- move through to `Graphics` folder

```
cd ..\Graphics
```

The above movement, is moving through the directories 'relative' to the current position in the file system.

Another path that can be taken , using an absolute movement is

- start at the drive where the folder is
- start from the root folder
- move to the first folder Multimedia Files
- move to the next folder Graphics

```
cd “c:\Multimedia Files\Graphics”
```

Notice we have used “quotation marks” in this command.

Because MS-DOS does not allow spaces in a file name, but Windows does, we use the quotation mark to tell MS-DOS that the (space) is supposed to be there.

MS-DOS Prompt

Continuing with the same diagram, now that we are inside the `Graphics` folder we want to move to the `My Documents` folder. The relative path we have to travel is:

- move back one folder (..) takes us to the Multimedia Files
- move back another folder (..) takes us to the ‘root’ folder
- move through to `My Documents` folder

```
cd “..\..\My Documents”
```

The absolute path to the same location

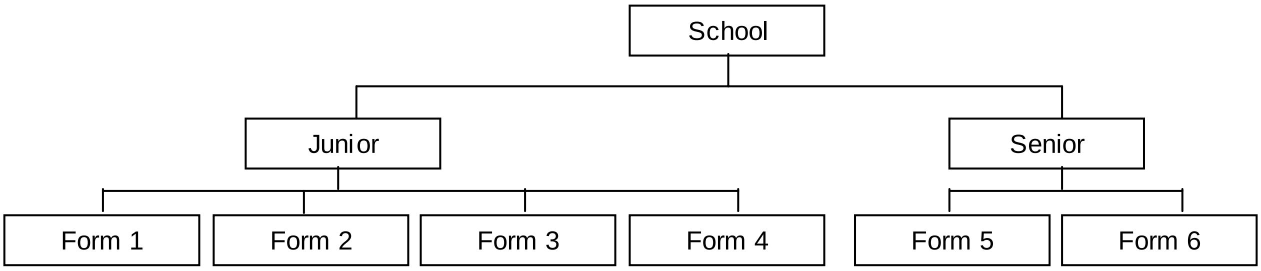
- start at the drive where the folder is
- start from the root folder
- move to the first folder My Documents

```
cd “c:\My Documents”
```

Windows 95/98 Users can read a good summary, review of how to find files on your computer, looking around in a booklet that comes with Windows 95 *Introducing Microsoft Windows 95.*

The section to look through *“Seeing What’s on Your Computer.”*

Figure 1: A directory structure

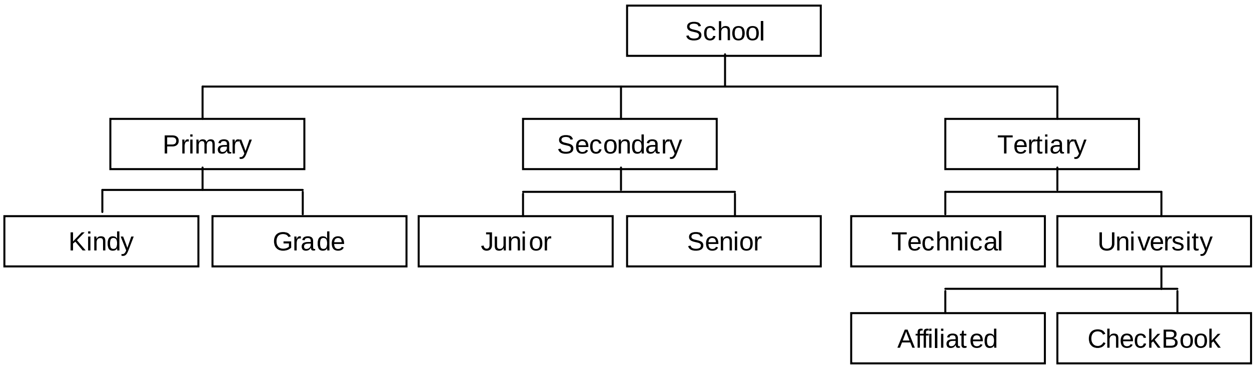


Exercise 1

Using the above diagram as a reference for writing command instructions to move from

- Directory Form 1 to Form 4
- Directory Form 2 to Form 5
- Directory Form 3 to Junior
- Directory Form 4 to Senior

Figure 2: A directory structure

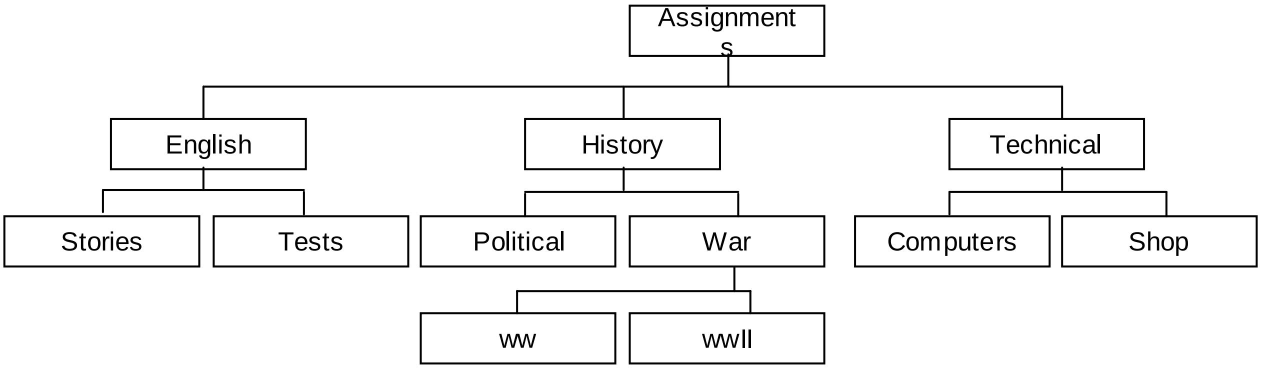


Exercise 2

Using the above diagram as a reference for writing command instructions to move from

- Directory Kindy to Senior
- Directory Primary to Secondary
- Directory Junior to Affiliated
- Directory CheckBook to Technical

Figure 3: A directory structure

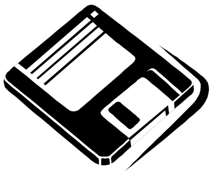


Exercise 3

Using the above diagram as a reference for writing command instructions to move from

- Directory ww to wwII
- Directory Stories to War
- Directory Computers to ww
- Directory Tests to Computers

PREPARING DISKS TO HOLD INFORMATION



If you have a new disk that has never been used, you must prepare it for storing information. On microcomputers, you do this by running a program that prepares the diskette by laying out the areas where data is to be placed. The program "formats" or "initialises" the disk so the disk's file organisation is understood by the operating system.

During the format process the disk is cleared, the logical layout of the diskette is defined and an area is set aside for an index of the contents of the diskette *(obviously the initial content is very minimal or has nothing in it but the index.)* The most common floppy disk layout is the MS-DOS layout 'format' with the index known as the File Allocation Table (FAT.) The FAT maintains a list of areas currently used, and which file is using it, as well as the areas currently available for storing additional files.

MS-DOS uses FAT

Windows98 uses FAT32 a more recent update to FAT.

WindowsNT uses NTFS. NT File System.

OS/2 uses HPFS. High Performance FS

Macintosh uses HFS. Hierarchical File System, and

Linux uses ext2 –

Available space is also called "free space".

The format process also checks the disk to ensure each major area (known as a sector) can be written to and read from reliably. If one of the disk areas cannot reliably store files, then it is marked in the index (FAT) as a "bad sector" and will not be used.

Preparing a Disk in Windows 95/98/NT

To prepare, format, a floppy diskette in Windows 95/NT, the steps are as follows.

My Computer

3½ Floppy (A:)

- Insert the diskette to be formatted into the Floppy Disk Drive
- Open "My Computer" by double clicking on the icon
- Click, select, the floppy disk drive icon where you have placed the diskette
- Select the file menu
- Select the "**Format**..." command
- On the following dialog box, select [Start] button to begin formatting.

As Windows 95/NT formats the diskette it will show a graphical bar as it progresses. When the format process is complete, a dialog box is displayed with summary information of the work that it has done.

MS-DOS - FORMAT

MS-DOS Prompt

The MS-DOS program/command to prepare a disk for use is:
`FORMAT.COM`

For example, the following command formats a floppy disk in drive A:

```
FORMAT A:
```

You should always specify the drive that contains the disk you want to format.

As it formats the disk, MS-DOS displays the percentage of the disk that has been formatted. After the disk is formatted, you are prompted to give the disk a "volume label". Type the name you want to give the disk, or press ENTER if you don't want a label.

MS-DOS then displays information about how the disk was formatted:

```
C:\>format a:
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
Formatting 720K
Format complete.

Volume label (11 characters, ENTER for none)?

      730,112 bytes total disk space
       61,440 bytes in bad sectors
      668,672 bytes available on disk

        1,024 bytes in each allocation unit.
          653 allocation units available on disk.

Volume Serial Number is 2657-11CE

Format another (Y/N)?
```

The Windows operating systems and MS-DOS provide the following summary information.

Bytes total disk space Indicates the storage capacity of the disk.

Bytes used by the system Appears if you have transferred system files to the disk. This line shows how much space is used by the system files.

Bytes in bad sectors Indicates how much of the disk is unusable because of bad sectors. If there are no bad sectors, this line is omitted.

Bytes available on disk Indicates the total disk space minus the space taken up by the system files and bad sectors.

Bytes in each allocation unit and *allocation units available on disk*. Indicates the minimum storage unit the disk has been configured to store and how many units are available. If you multiply the two numbers on these lines, the result is the same as “bytes available on disk” number.

Volume serial number Indicates the serial number assigned to the disk. This number is unique for each disk to help the operating system differentiate between different floppy disks.

Creating a System Startup, or System Boot Disk

A system startup disk, or system boot disk is a diskette with enough operating system files on it that it can be used to start up a microcomputer. Review our earlier discussion on turning on the PC.

Windows 95/98

Windows 95/98 provides two different ways of creating system startup, or system boot disks. With *Windows Explorer* up on the screen, and the floppy diskette selected, go to the menu: **File | Format** ... command, the dialog box provides the different options discussed earlier for formatting a disk.

If the diskette has already been formatted, then the user can select that only the system files are to be copied. (Copy system files only)

If the diskette also needs to be formatted then the system files can be copied as part of the formatting process.
(copy system files)

MS-DOS – Doing it on the command line

In MS-DOS, preparing a disk is performed using the program *"format.com"*. The parameter **/s** tells format to transfer the "system" files to the disk. For example, the following command formats the disk in drive a, then copies system files to the disk:

```
format a: /s
```

To make a disk that is already formatted a system disk, use the **sys** command. For example, the following command copies system files from the current drive to a formatted disk in drive b:

```
sys a:
```

Getting HELP

MS-DOS commands accept the parameter **/?** which forces the command to display a "help" screen of command information. For example the following command will display more helpful information on the format command.

```
format /?
```

The computer will display the following information.

```
C:\>format /?
Formats a disk for use with MS-DOS.

FORMAT drive: [/V[:label]] [/Q] [/U] [/F:size] [/B | /S] [/C]
FORMAT drive: [/V[:label]] [/Q] [/U] [/T:tracks /N:sectors] [/B | /S] [/C]
FORMAT drive: [/V[:label]] [/Q] [/U] [/1] [/4] [/B | /S] [/C]
FORMAT drive: [/Q] [/U] [/1] [/4] [/8] [/B | /S] [/C]

[ etc. etc. etc. ]
```

FILE MANAGEMENT

In this section we will take a look at the common activities related to managing files; these being copying, moving, and deleting files.

SOURCES AND REFERENCES

Microsoft MS-DOS 5.0 Reference Manual, Microsoft Press
Tornsdorf, Manfred & Helmut Tornsdorf, DOS 6.0 Complete, Abacus, 1993
Student Manual DOS Introduction Course, Education Development Centre, 1992
Siliva, Tonga. Liahona High School 1997 Term 4 Final Exam, Computer Studies Form 5 (TSC)

http://www.qsc.edu.to - Queen Salote’s SchoolNET Website
http://www.tongatapu.net.to - **Tonga** on the **'NET**

Queen Salote’s SchoolNET Website does not require Internet access as it is not connected to the world wide Internet but uses the same technology within Queen Salote College and participating schools.

http://www.qsc.edu.to is available on all networked computers at Queen Salote College.