UNIX commands relating to files

	Command	Description
1	cat	A way of displaying the contents of a text file. (Command comes from the word 'con**cat**enate').
2	more	A way of displaying the contents of a text file. (This is a 'file perusal' filter and displays files a page at a time).
3	cp	**C**o**p**y file
4	mv	**M**o**v**e a file (also used to rename a file).
5	rm	**R**e**m**ove a file. (Can also be used with the appropriate option to remove a directory and all its contents)
6	touch	This can be used to quickly create an empty file (or to date stamp an existing file).
7	file	To determine the file type.
8	head	Displays the first few lines of a file.
9	tail	Displays the last few lines of a file.
10	wc	Used to display how many lines, words and characters are held in a file.
11	find	Find the location of a file.
12	du	Used to display how big a file is (disk usage). The figure given is the number of memory blocks. Each block on our system is 1024 bytes.
13	df	Used to display the amount of disk space that is free (unused).

Commands, arguments and options.
Shell commands lines entered at the shell command prompt consist of one or more words separated by spaces.
The first word is the ***command*** and the remaining words (if any) are the ***command arguments*** (sometimes known as ***command parameters***).
The command arguments are the 'things' that the command acts upon.
An option is a special kind of argument (it is usually a letter prefixed with a minus sign) that tells the command how it should behave or what it has to do.
The following table shows how commands, arguments and options work. The example assumes a user named os who has a home directory /home/os and a subdirectory under that named /home/os/cwork.

Command	Details
cd	Change to the home directory (/home/os). The cd command on its own will always return a user to his/her home directory.
ls	List the contents of the directory
ls cwork	List the contents of the directory named cwork
ls -t cwork	List the contents of the directory named cwork but show the information sorted by modification time before sorting alphabetically.

Examples of command options...

	Command	Description
1	ls mywork	List the contents of a directory named mywork
2	ls **-l** mywork	List directory contents in **l**ong format (this displays additional useful information)
3	ls **–p** mywork	List directory and **p**ut a slash (/) after each entry that is a directory name.
4	ls **–s** mywork	List directory and show the **s**ize in blocks.
5	ls **–t** mywork	List directory and sort by the **t**ime modified before sorting alphabetically.
6	ls **–u** mywork	List directory and sort by the time of the last access instead of last modification. (i.e. sort by the last time **u**sed)
7	ls **–x** mywork	List directory and display the entries sorted across the screen rather than down it.
8	ls **–R** mywork	List directory and **r**ecursively list the contents of any subdirectory encountered.

So... although there are only a few basic commands that you need to know to get started, each command can be used with many options that control how the command behaves. It is impossible to remember all the commands and their options so a user will frequently need to refer to the on-line help pages referred to as the 'man pages' (i.e. the **man**uals)

Using the man pages

The ***man pages*** are simple to use. For example, if you want to know all about the **ls** command simply enter the command:

man ls

Everything you ever wanted to know (and don't want to know) about the ls command is then displayed on the screen When viewing a man page remember to:

- Use the return key to scroll down through the information one line at a time
- Use the space bar to scroll down through the information one page at a time
- Use **q** to quit the man page and return to the dollar prompt.

Unix system administrators refer to man pages all the time. There are too many Unix commands and options for anyone to remember them. Take your time when reading man pages as they often contain the information you are often looking for.

Exercise 3 - Using UNIX commands relating to files.

Before starting:

- Move into the directory named **testdir**.
- List the contents of that directory.
- Move down into the directory named **play**.
- List the contents of that directory (it should be empty at this stage)

	Instruction and/or Unix command	Observations
1	**man ls > lsmanpage**	Create a file containing the contents of the man page for the **ls** command. Note the use of the command output redirection symbol (>). This sends the ouput of the command 'man ls' to the file instead of to the screen.
2	**Ls**	List the contents of the directory.
3	**file lsmanpage**	Use the file command to find out what type of file **lsmanpage** is.
4	**cat lsmanpage**	Display the contents of the file name **lsmanpage**. It will fly by so quickly that you will probably only see the last few lines of it.
5	**more lsmanpage**	Use the **more** command to see the contents of a file a page at a time. When viewing a file using this command you will need to use the return key to scroll down the file by one line at a time, the spacebar to scroll a page at a time and use q to finish viewing.
6	**cp lsmanpage lsmanpage2** **ls**	Make a copy of the file List the contents of the directory (it should contain **lsmanpage** and **lsmanpage2**).
7	**mv lsmanpage2 lsmanpage3** **ls**	Rename the file. List the contents of the directory (it should contain **lsmanpage** and **lsmanpage3**)
8	**head lsmanpage** **tail lsmanpage**	Display the first 10 lines of the file. Dislay the last 10 lines of the file.
9	**wc lsmanpage**	Make a note of the 3 numbers that are displayed and write down what they represent.
10	**du lsmanpage**	Write down the number of disk blocks that this file uses. How many bytes does this represent?
12	**rm lsmanpage3** **ls**	Remove a file. List the contents of the directory (it should now only contain **lsmanpage**).
13	**cat > mydetails** **Surname: Bloggs** **Firstname: John** **Age: 21** ***<now press Ctrl/Z to terminate text entry>***	The cat command can be used with the redirection operator (>) to create a new text file 'on-the-fly'. Enter the command **cat >mydetails** then press the return key, then continue to enter text on the next blank line and to terminate text entry press the Ctrl and Z keys at the same time.
14	**cat mydetails**	Now use the cat command without any redirection to display the contents of the file.
15	**cat -n mydetails**	Display the text file and show line **n**umbers.
16	**cat -n lsmanpage** **cat -n lsmanpage | more**	Display the original large text file and show line **n**umbers. Again the file goes by too quickly. Try the **cat -n** command again but send its output to another command, ie. the **more** command. This is referred to as piping (think of a pipe between 2 commands). The vertical bar character (to the left of the z key) is the pipe symbol.
17	**cat -n lsmanpage > num-lsmanpage** **ls** **more num-lsmanpage**	Redirect the output of the **cat -n lsmanpage** command to another file named **num-lsmanpage**. List the contents of the directory. Display the contents of the new file using the **more** command.
18	**cat > mycar**	Create another file using the on-the-fly technique.

	Car make: Ford **Model: Escort** **Colour: Silver** **<*now press Ctrl/Z to terminate text entry*>**	
19	**cat mycar >> mycar2** **cat mycar >> mycar2**	Use the >> operator to **append** the contents of one file to another file. Examine the contents of mycar and mycar2.
20	**touch quickone** **ls** **du quickone** **cat quickone**	Touch can be used to quickly create an empty text file.