UNIX Tutorial Two

2.1 Copying Files

cp (copy)

`cp file1 file2` is the command which makes a copy of **file1** in the current working directory and calls it **file2**

What we are going to do now, is to take a file stored in an open access area of the file system, and use the `cp` command to copy it to your unixstuff directory.

First, `cd` to your **unixstuff** directory.

```
% cd ~/unixstuff
```

Then at the UNIX prompt, type,

```
% cp /vol/examples/tutorial/science.txt .
```

Note: Don't forget the dot **.** at the end. Remember, in UNIX, the dot means the current directory.

The above command means copy the file **science.txt** to the current directory, keeping the name the same.

(Note: The directory **/vol/examples/tutorial/** is an area to which everyone in the school has read and copy access. If you are from outside the University, you can grab a copy of the file here. Use 'File/Save As..' from the menu bar to save it into your **unixstuff** directory.)

Exercise 2a

Create a backup of your **science.txt** file by copying it to a file called **science.bak**

2.2 Moving files

mv (move)

`mv file1 file2` moves (or renames) **file1** to **file2**

To move a file from one place to another, use the mv command. This has the effect of moving rather than copying the file, so you end up with only one file rather than two.

It can also be used to rename a file, by moving the file to the same directory, but giving it a different name.

We are now going to move the file science.bak to your backup directory.

First, change directories to your unixstuff directory (can you remember how?). Then, inside the **unixstuff** directory, type

```
% mv science.bak backups/.
```

Type ls and ls backups to see if it has worked.

2.3 Removing files and directories

rm (remove), rmdir (remove directory)

To delete (remove) a file, use the **rm** command. As an example, we are going to create a copy of the **science.txt** file then delete it.

Inside your **unixstuff** directory, type

```
% cp science.txt tempfile.txt
% ls
% rm tempfile.txt
% ls
```

You can use the **rmdir** command to remove a directory (make sure it is empty first). Try to remove the **backups** directory. You will not be able to since UNIX will not let you remove a non-empty directory.

Exercise 2b

Create a directory called **tempstuff** using **mkdir** , then remove it using the **rmdir** command.

2.4 Displaying the contents of a file on the screen

clear (clear screen)

Before you start the next section, you may like to clear the terminal window of the previous commands so the output of the following commands can be clearly understood.

At the prompt, type

```
% clear
```

This will clear all text and leave you with the % prompt at the top of the window.

cat (concatenate)

The command cat can be used to display the contents of a file on the screen. Type:

```
% cat science.txt
```

As you can see, the file is longer than than the size of the window, so it scrolls past making it unreadable.

less

The command less writes the contents of a file onto the screen a page at a time. Type

```
% less science.txt
```

Press the [**space-bar**] if you want to see another page, and type [**q**] if you want to quit reading. As you can see, **less** is used in preference to **cat** for long files.

head

The **head** command writes the first ten lines of a file to the screen.

First clear the screen then type

```
% head science.txt
```

Then type

```
% head -5 science.txt
```

What difference did the -5 do to the head command?

tail

The `tail` command writes the last ten lines of a file to the screen.

Clear the screen and type

```
% tail science.txt
```

Q. How can you view the last 15 lines of the file?

2.5 Searching the contents of a file

Simple searching using less

Using `less`, you can search though a text file for a keyword (pattern). For example, to search through **science.txt** for the word **'science'**, type

```
% less science.txt
```

then, still in `less`, type a forward slash [/] followed by the word to search

```
/science
```

As you can see, `less` finds and highlights the keyword. Type [**n**] to search for the next occurrence of the word.

grep (don't ask why it is called grep)

`grep` is one of many standard UNIX utilities. It searches files for specified words or patterns. First clear the screen, then type

```
% grep science science.txt
```

As you can see, `grep` has printed out each line containg the word **science**.

Or has it ????

Try typing

```
% grep Science science.txt
```

The **grep** command is case sensitive; it distinguishes between Science and science.

To ignore upper/lower case distinctions, use the -i option, i.e. type

```
% grep -i science science.txt
```

To search for a phrase or pattern, you must enclose it in single quotes (the apostrophe symbol). For example to search for spinning top, type

```
% grep -i 'spinning top' science.txt
```

Some of the other options of grep are:

-v display those lines that do NOT match
-n precede each matching line with the line number
-c print only the total count of matched lines

Try some of them and see the different results. Don't forget, you can use more than one option at a time. For example, the number of lines without the words science or Science is

```
% grep -ivc science science.txt
```

wc (word count)

A handy little utility is the **wc** command, short for word count. To do a word count on **science.txt**, type

```
% wc -w science.txt
```

To find out how many lines the file has, type

```
% wc -l science.txt
```

Summary

Command	Meaning
`cp file1 file2`	copy file1 and call it file2
`mv file1 file2`	move or rename file1 to file2
`rm file`	remove a file
`rmdir directory`	remove a directory
`cat file`	display a file
`less file`	display a file a page at a time
`head file`	display the first few lines of a file
`tail file`	display the last few lines of a file
`grep 'keyword' file`	search a file for keywords
`wc file`	count number of lines/words/characters in file
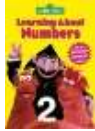
M.Stonebank@surrey.ac.uk, © 9th October 2000