9. Grouping Commands and Special Functions

This lab deals with grouping commands as well as special functions such as Pipes, Tees, Filters and Wildcards.

Lab 9.1 - Grouping Commands

Rather than typing commands separately such as:

```
who
ls
cal
```

you can group them using a semicolon ;

Who; ls; cal –*which is the same as typing them in on three different lines*

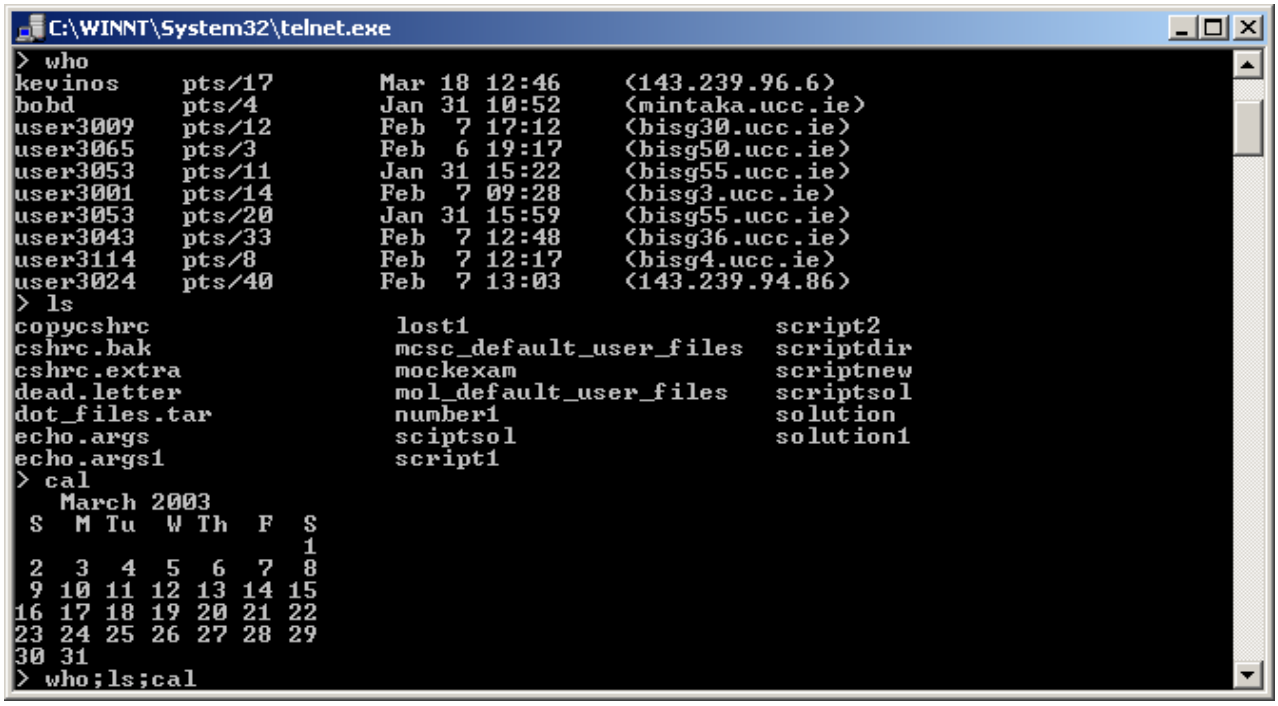

Figure 9.1: Grouping Commands Example

Can be very useful if you want to redirect the output into a file as you don't have to send the output of each command to a file and then append the output of the next command and so on.

```
(date; who; cal) > group
more group
```

Lab 9.2 - Pipes

So far you have been introduced to some basic UNIX commands that are essentially one instruction, cd , pwd , ls . UNIX is a powerful operating system and capable of much more.

You may want to open a file and have UNIX perform some other function on this file as you open it. To do so you use what is called piping commands. Piping connects the output of one program to the input of another program. You pipe commands together with the pipe symbol or command; |

Here is an example:

```
Cal 1999; date; ls; who  | more
```

This command tells UNIX you want to carry out the list of commands. You use the pipe so that when these commands are done, the output will be displayed to the screen, one page at a time, rather than just flashing past as it is printed to the screen. You can pipe as many commands together as you want, just use the pipe symbol between each command.

Lab 9.3 - Tees

A tee allows you to save the output from a command in a file at the same time the output is piped to another command

```
(cal 2002; cal 2003) | tee calfile | more
```

Lab 9.4 - Filters

A filter takes a stream of data from its standard input, transforms the data in some way and sends the results to the standard output (screen). The head and tail commands are examples of filters.

Lab 9.5 - Wildcards

Wildcards can alleviate the nuisance of typing and retyping file names.

Lab 9.5.1 - Asterix *

The asterix * is one of the most commonly used wildcards in unix

ls f*

This will return all the files in your directory starting with f

ls file*

This will return all the files in your directory starting with file

ls *d

This will return all the files in your directory ending with d

ls *ou*

This will return all the files in your directory with ou in their name

It can be used with many different commands such as rm or cat for example

rm file*

Which will remove anything starting with file

Again be carefull not to type rm *

Lab 9.5.2 - Question mark ?

The question mark ? is another wildcard that matches just one character at a time.

Ls ?file

It will only return anything that has file directly after that one letter

Ls test?

It will only return anything that has test directly before that one letter

Lab 9.5.3 - Square Brackets [ ]

Square Brackets [ ] are the final wildcard used that matches any character inside of them.

```
Ls file[abc ]
```

It will return files called filea fileb or filec

You can also combine wildcards

```
Ls [atf]*
```

This will return anything with a, t or f at the beginning.