13. Job Processing

Lab 13.1 - Processes

Each process submitted to the kernel is given a unique process ID (PID).

The output of the ps command, without any options specified, will include the process ID, the terminal from which the process was started, the amount of time the process has been running, and the name of the command that started the process.

The main options when using ps are:

ps –l *// Display detailed information about each process*

ps –u *// Display more detailed information about each process*

ps –a *// List information about all processes*

A process may be in the foreground, in the background, or be suspended. In general the shell does not return the UNIX prompt until the current process has finished executing.

Some processes take a long time to run and hold up the terminal. Backgrounding a long process has the effect that the UNIX prompt is returned immediately, and other tasks can be carried out while the original process continues executing.

Lab 13.2 - Sleep Command

The sleep command creates a process that sleeps for a certain specified time. It waits for a certain specified number of seconds before returning the shell.

Sleep 15 – *waits 15 seconds before returning the shell for you to carry on*

(sleep 30; echo I am awake now) – *used to delay execution of another command*

waits for the process to finish and executes the echo command

Lab 13.2.1 - Running background processes

To background a process, type an **&** at the end of the command line. For example, the command sleep waits a given number of seconds before continuing. Type

```
sleep 10
```

This will wait 10 seconds before returning the command prompt %. Until the command prompt is returned, you can do nothing except wait.

To run sleep in the background, type

```
sleep 10 &
[1] 6259
```

The **&** runs the job in the background and returns the prompt straight away, allowing you do run other programs while waiting for that one to finish.

The first line in the above example is typed in by the user; the next line, indicating job number and PID, is returned by the machine. The user is be notified of a job number (numbered from 1) enclosed in square brackets, together with a PID and is notified when a background process is finished. Backgrounding is useful for jobs which will take a long time to complete.

Lab 13.2.2 – Backgrounding a current foreground process

At the prompt, type

```
sleep 100
```

You can suspend the process running in the foreground by holding down the [control] key and typing z (written as **^Z**) Then to put it in the background, type

```
bg
```

Note: do not background programs that require user interaction e.g. pine

Lab 13.3 - Listing suspended and background processes

When a process is running, backgrounded or suspended, it will be entered onto a list along with a job number. To examine this list, type

```
jobs
```

An example of a job list could be

[1] Suspended sleep 100
[2] Running netscape
[3] Running nedit

To restart (foreground) a suspended processes, type

```
fg %jobnumber
```

For example, to restart sleep 100, type

```
fg %1
```

Typing **fg** with no job number foregrounds the last suspended process.

Lab 13.4 - Killing a process

kill (terminate or signal a process)

It is sometimes necessary to kill a process (for example, when an executing program is in an infinite loop)

To kill a job running in the foreground, type **^C** (control c). For example, run **sleep 100** then kill it with **^C**

To kill a suspended or background process, type

kill %*jobnumber*

For example, run **sleep 100 &** then type **jobs** to see its job number. If it is job number 4, type

kill %4

To check whether this has worked, examine the job list again to see if the process has been removed.

ps (process status)

Alternatively, processes can be killed by finding their process numbers (PIDs) and using kill PID_number.

Run **sleep 100 &** again, then type

ps

PID TT S TIME COMMAND
20077 pts/5 S 0:05 sleep 100
21563 pts/5 T 0:00 netscape
21873 pts/5 S 0:25 nedit

To kill off the process sleep 100, type

kill 20077

and then type **ps** again to see if it has been removed from the list.

If a process refuses to be killed, uses the **-9** option, i.e. type

kill -9 20077

Note: It is not possible to kill off other users' processes !!!

Exercise 13.1

Create a file called **months** with the months written into it using vi

Sort the file in reverse order and save the result into a file called **months2** using redirection

Start up a process to sleep for 300 seconds and then print out the date onto the screen.

Suspend that process

Move it to the background

Set up another process in the background to sleep for 50 seconds and then print out the text "hello there process two done" onto the screen.

Kill the first suspended process you created.

Bring the second process into the foreground