Worksheet 2: Process Control, Scripts and Filesystems

Updated: 24th February, 2020

Note: You are encouraged to keep notes on your observations and answers to the worksheet exercises and lecture examples.

Assistance is only given when students can demonstrate their own attempt of resolving any problems.

Make sure you include things that went wrong. Your notes will serve as evidence of attempt, and often you can learn a lot analysing things that did not work.

Your notes should include answers to the following questions:

- What programs or websites did you use in the practical today?
- What do you need to remember about your practical work today?
- How does your program work?

1. Infinite Loop

Write a c program that runs forever (an example is given below). You should include in your program input from **`stdin`** and output to **`stdout`** and **`stderr`**.

Compile your program twice with different executables, e.g., `infinity1` and `infinity2`

```c
#include <stdio.h>
#include <unistd.h>

int main()
{
    printf("starting...\n");
    while(1)
    {
        printf(".");
        sleep(5);
    }
    return 0;
}
```

Note: This program may not give the expected output. Why not? Fix it.

Comment out the input and output. Execute the first processes, halt it (not terminate) use the `ps` command and the `jobs` command to see what is happening with the process. Now execute the second process, halt it as well, and investigate both processes. Re-start both processes in the background, and use `ps` to see what's happening.

2. The top Command

Investigate the `top` command, use it to view the processes. Investigate `nice` and `renicen` modify the priority of one process, use `top` to see the effect.

3. Termination

Terminate both processes. Remove the comments from `stdin`, `stdout` and `stderr` from your programs. Modify them so you can see which process requires input or is outputting. Execute one in the foreground, and one in the background. Explain what is happening.

Now run both in the background, and use `vi` to edit any text file. What happens?

Terminate both processes, re-execute them in a way as to stop them mis-behaving

4. Scripting

Write a shell script to execute your processes as above. Execute the following on the command line:

```
for file in `ls`;
do
    echo -n $file;  # Also try chmod instead of echo
done
```

Try without semi-colons

5. Rename

Write a command that will rename all of the `.htm` files in a directory to `.html`

6. Explain extents

Find two acceptable (high quality references) articles on `extents`. Use these articles to explain the possible shortcomings of `inodes`, and how `extents` resolve these issues.

End of Worksheet