Worksheet 5: C System Calls

Updated: 25th 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?

You should consult the lecture notes whilst attempting these exercises.

1. `strace`

Investigate the use of `strace` (man). Use `strace` to view the system calls to `ls`.

2. Files and Processes

(a) Write a c program called `cpdir` (using only system calls) that copies all the files (including subdirectories) from one directory to another. Both source and destination must be imported as command-line arguments. If the destination directory exists, your program should prompt the user to overwrite or exit. Each file should be copied in its own process (forked). Your program should read the directory, and for each file, `fork`, open the file for reading, open a file with the same name in the destination directory for writing, and copy the contents.

Don't forget to handle each child process as it finishes! The name of each file and subdirectory should remain the same.

Note: This program could cause chaos in your system. Take extra care when testing - ie., test small first, and backup your files!

(b) Write a `C` program that uses `fork`, `exec` and `pipe` to perform the equivalent of the shell command:

```
[user@pc]$ ls /dev | head -25
```

3. Named Pipes

Try out the named pipe example in the lecture notes.

Send multiple files e.g., **`*.c`**

4. `kill`

Kill off all the processes that you didnt handle correctly in the rest of the prac!

End of Worksheet