

Week 8 Tutorial
NWEN 241
Systems Programming

Alvin Valera

`Alvin.valera@ecs.vuw.ac.nz`

How to know which system calls are invoked?

Two commands:

- a) **ltrace** – traces call to library functions
- b) **strace** -traces system calls

See details in Linux manual pages

Usage :

ltrace ./<program executable file>

ltrace -S ./<program executable file> (also display Kernel system calls)

Example: Does the `sin()` result in a system call?

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    double pi = 3.14;
    printf("start\n");
    double s = sin(pi);
    printf("sine of pi is %f\n");
    return 0;
}
```

Linux ps command

- Used to obtain information about processes that are running in the current shell

```
$ ps
  PID TTY          TIME CMD
 31843 pts/35        00:00:00 bash
 31850 pts/35        00:00:00 ps
```

Process ID

Every process is assigned a PID by the kernel

More about ps

- The command ps can display threads

```
$ ps -eLf
```

- To show process tree, us pstree

```
$ pstree
```

- What happened to the init process?

Let's write some code

- Write a simple code to demonstrate `fork()`
 - Child process to display its process id
 - Parent process to display its process id and child's process id
- What happens in the statement `fork() && fork();` ?
- Write a simple code to demonstrate `exec()`
 - Use `execl()` to invoke `"/bin/ls"`
 - Are the statements after `execl()` executed?
 - Can you rewrite this code such that after `exec()` call, the rest of the statements after it are executed?

Let's write some code

- Write a simple code to demonstrate `wait()` to let parent wait for child process to exit
- Write a simple code to study what the parent receives in the wait status
 - Let child process ask for code to pass to `exit()` system call
 - Display what will the `wait()` system call in the parent receive and how to make sense of it