Dan Otieno.

CPE434-01.

HOMEWORK 1.

01/24/23.

1. **What are system calls used for? What are the system calls that the command interpreter must execute so that a new process starts on a Linux or Unix system?**
    a. ***System calls provide an interface to the services made available by an operating system and are accessed using application programming interface or API. In Unix/Linux systems, the fork call creates a process, exec loads a program into the process and shell waits for the process to terminate or continues with user commands.***
2. **Forking a new process in Linux is claimed to use a copy-on-write policy so that although the text heap and stack are the same for a parent and child before and after the fork they change as soon as a data item is written by either the parent or child. write a small program that verifies whether this is true or not. upload the commented code together with a sample screen shot to verify this statement is true.**

```
using namespace std;
#include <iostream>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

/*******Function for fork********/
void forkprogram()
{
    int val = 10; //Initialize val, I just picked 10 instead of 0.
    pid_t pid = fork();

    /* Val is not updated by writing any data,
    will use this to check if Heap and Stack are the same*/
    cout << "PID: " << getpid() << "\tVal before data is written: " << val << endl;

    if(pid != 0) // Parent process to update val variable.
    {
            val += 5;
            cout << "PID: " << getpid() << "\tval after data is written by parent process: " << val
<< endl;
    }
    else
    {
            // else, print variable using child process.
            cout << "PID: " << getpid() << "\tChild process prints val after parent process: " <<
val << endl;
    }
```

```
}
/***********************************************/
/*********************MAIN**********************/
int main()
{
    forkprogram(); //Call function in Main.

    return 0;
}
```