Lab 3 – PROCESSES

A process is a program in execution. The process which creates another process is called parent process. The process which is created is called child process. We can identify process by their unique key called process identifier or pid (integer number). In Linux we can use fork() system call to create processes. By this system call new process is created containing copy of parent process. Both process (parent and child) continue executing instructions after fork(). The return number of fork() for new (child) process will be 0 (zero), whereas for parent process value will be nonzero positive process identifier. If fork() fails, it return a negative number. In this section, we create a simple program using fork() to create child process using

TASK 3.1

Using `getpid():` This function returns the pid of the current program. Use the following code and write the output.

```
int main(){
     int pid;
     pid = getpid();
     printf("Process ID is %d\n", pid);
     return 0;
}
```

TASK 3.2

What is the outcome of the following program?

```
int main(){
     long i;
     printf("Process ID is %d\n", getpid());
     for(i=0; i<=400;i++){
          printf("i is %d\n", i);
     }
     return 0;
}
```

TASK 3.3

Using `getppid()`: This function returns the pid of the parent process.

```
int main(){
     int ppid;
     ppid = getppid();
     printf("Parent Process ID is %d\n", ppid);
     return 0;
}
```

What is the outcome?