**ASSIGNMENT – 5**

1. Write a shell script to show the working of fork and process.

**Code**:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

int main()

{

printf("Hello World\n");

printf("I am %d before forking proccess\n",(int)getpid());

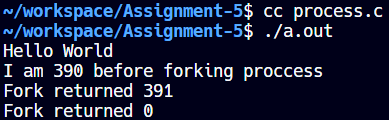
pid\_t pid = fork();

printf("Fork returned %d\n",(int)pid);

return 0;

}

**Output**:



1. Write a shell script to create a process using fork and identify the child process and parent process.

**Code**:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

int main()

{

printf("Hello World\n");

printf("I am %d before forking proccess\n",(int)getpid());

pid\_t pid = fork();

printf("Fork returned %d\n",(int)pid);

if(pid < 0)

{

perror("Fork failed\n");

}

else if(pid==0)

{

printf("I am child process pid %d\n",(int)getpid());

}

else

{

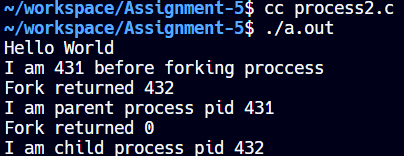
printf("I am parent process pid %d\n",(int)getpid());

}

return 0;

}

**Output**:



1. Write a shell script to find the output and behavior of the following C program using the fork() system call, and how does it demonstrate the creation and identification of parent and child processes.

**Code**:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

int main()

{

pid\_t pid, mypid, myppid;

pid = getpid();

printf("Before fork, pid is %d\n", pid);

printf("Called fork() system call\n");

pid = fork();

if(pid < 0)

{

perror("Fork failed\n");

return 1;

}

else if(pid == 0)

{

printf("I am child process, pid is %d\n", pid);

mypid = getpid();

myppid = getppid();

printf("My pid is %d and my parent pid is %d\n", mypid, myppid);

}

else

{

sleep(2);

printf("I am parent process.\n");

mypid = getpid();

printf("My pid is %d and my parent pid is %d\n", mypid, myppid);

}

return 0;

}

**Output**:

