Lab 04 OS

Utkarsh Bajaj, Roll No. 53, Reg No. 180905460

Ans 1) Program:

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

void main()

{

pid\_t pid;

int status;

char \*message;

int n;

printf("Fork program starting\n");

pid = fork();

if(pid == -1){

printf("Error\n");

exit(0);

} else if(!pid){

printf("I am the child\n");

exit(0);

} else {

wait(&status);

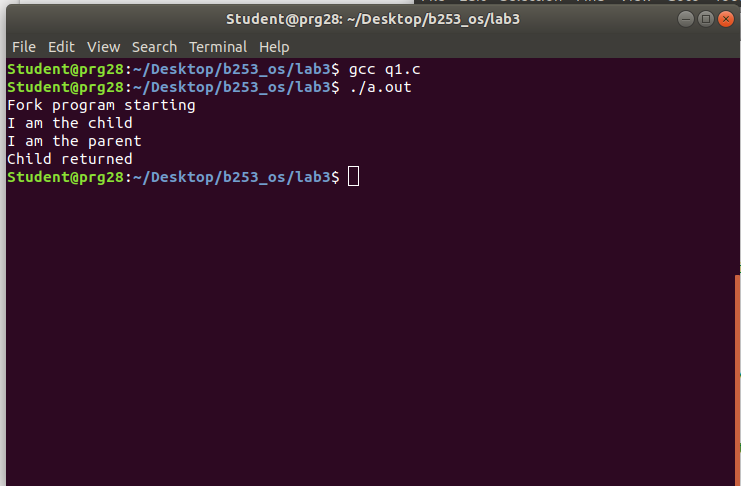
printf("I am the parent\n");

printf("Child returned now\n");

}

exit(0);

}



Ans 2) Program :

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

void main(){

int status;

pid\_t pid;

pid = fork();

if(pid < 0){

printf("Error\n");

exit(1);

} else if(!pid){

printf("Loading the binary executable in the child process\n");

printf("Executing child process\n");

execl("./p1", "p1", NULL);

} else {

wait(NULL);

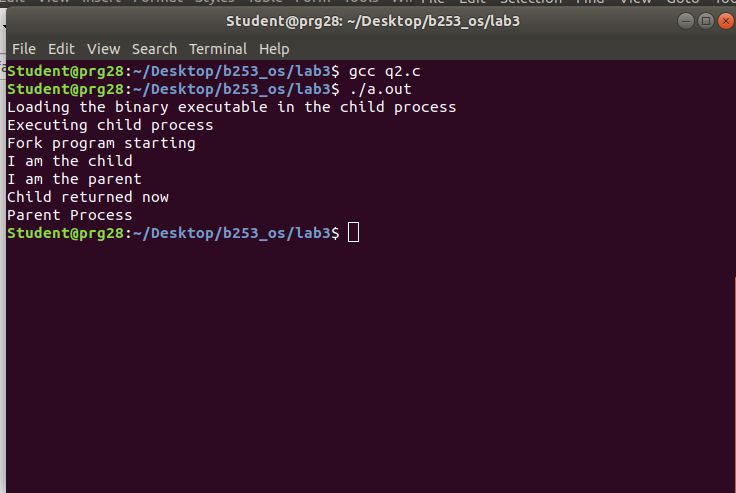
printf("Parent Process\n");

exit(0);

}

exit(0);

}



Ans 3) Program:

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

void main(){

int status;

pid\_t pid;

pid = fork();

if(pid < 0){

printf("Error\n");

exit(1);

} else if(!pid){

printf("This is the child process\n");

printf("Parent id: %d\n", getppid());

printf("Current id: %d \n", getpid());

exit(0);

} else {

wait(&status);

printf("This is the parent process\n");

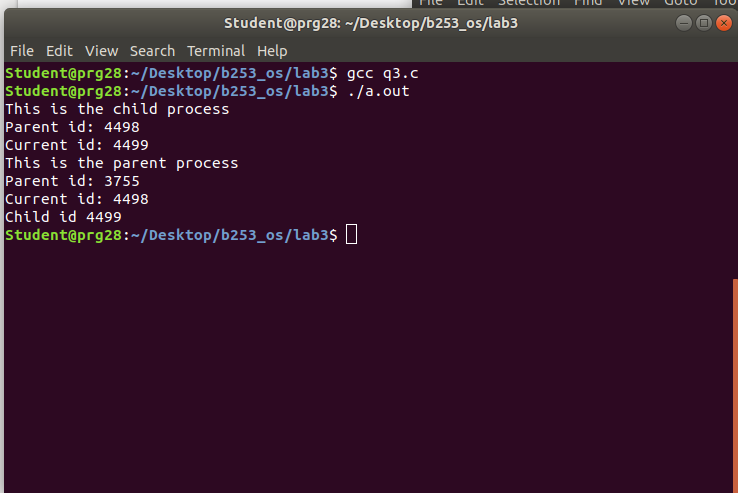
printf("Parent id: %d\n", getppid());

printf("Current id: %d\n", getpid());

printf("Child id %d\n", pid);

}

}



Ans 4) Program:

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

void main(){

int status;

pid\_t pid;

pid = fork();

if(pid < 0){

printf("Error\n");

exit(1);

} else if(!pid){

printf("Inside child\n");

exit(0);

} else {

printf("Inside parent\n");

sleep(1);

execl("/bin/ps", "ps", NULL);

}

exit(0);

}

