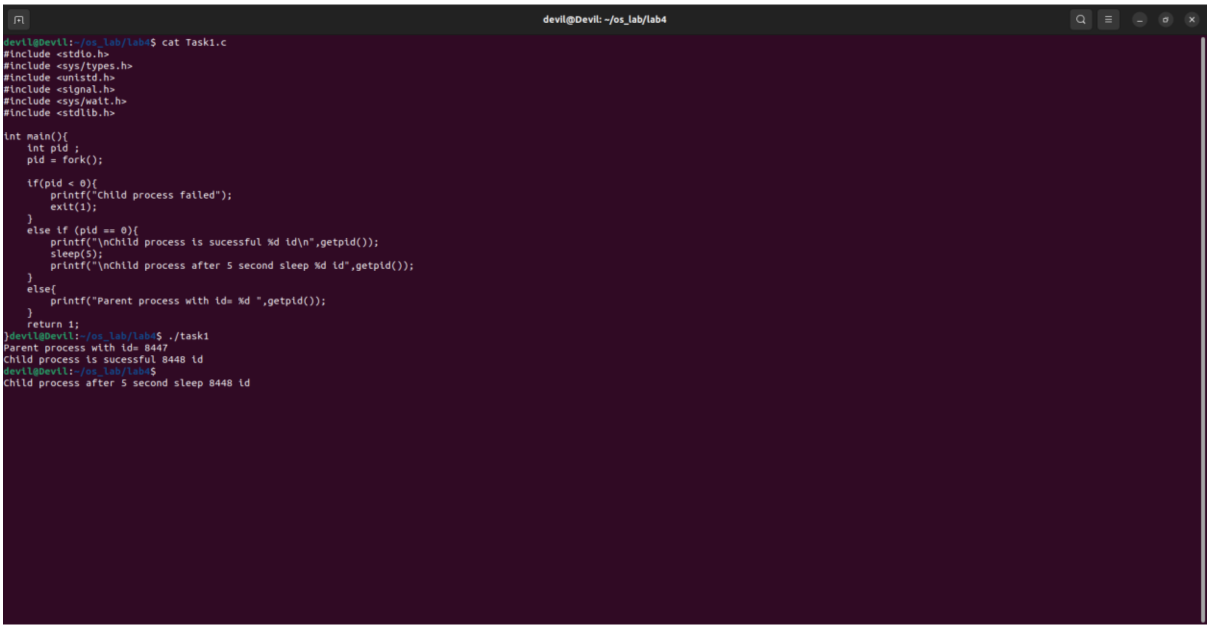
NAME : MUHAMMAD HAMZA

SEC : BS-SE-4A

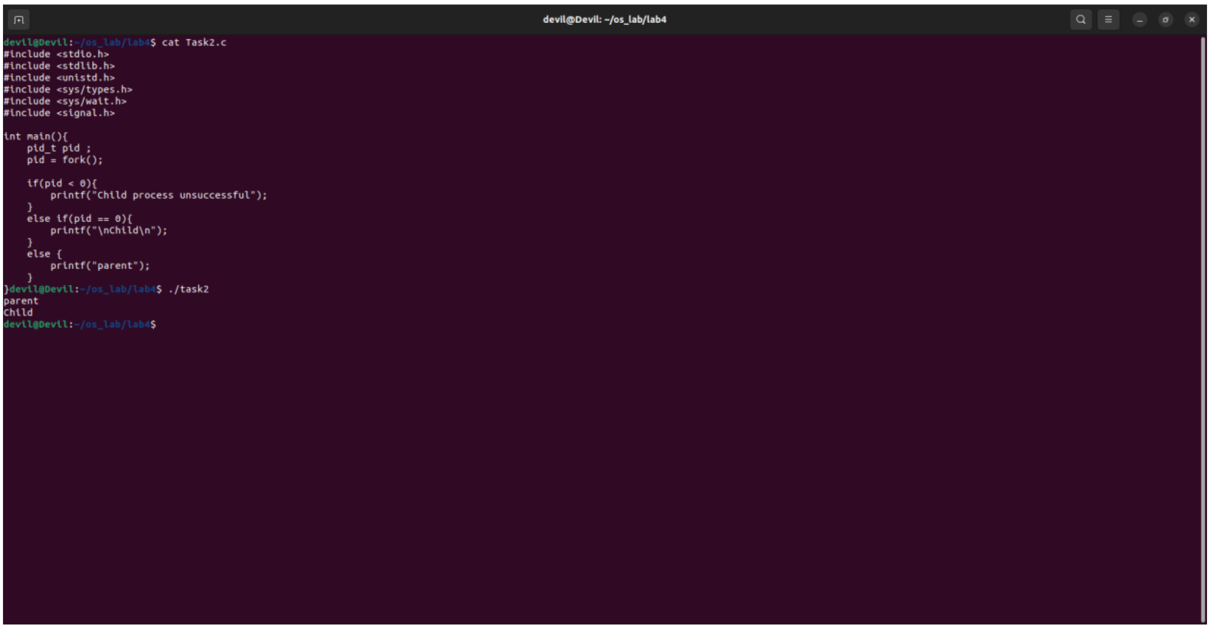
LAB 4

TASK 1



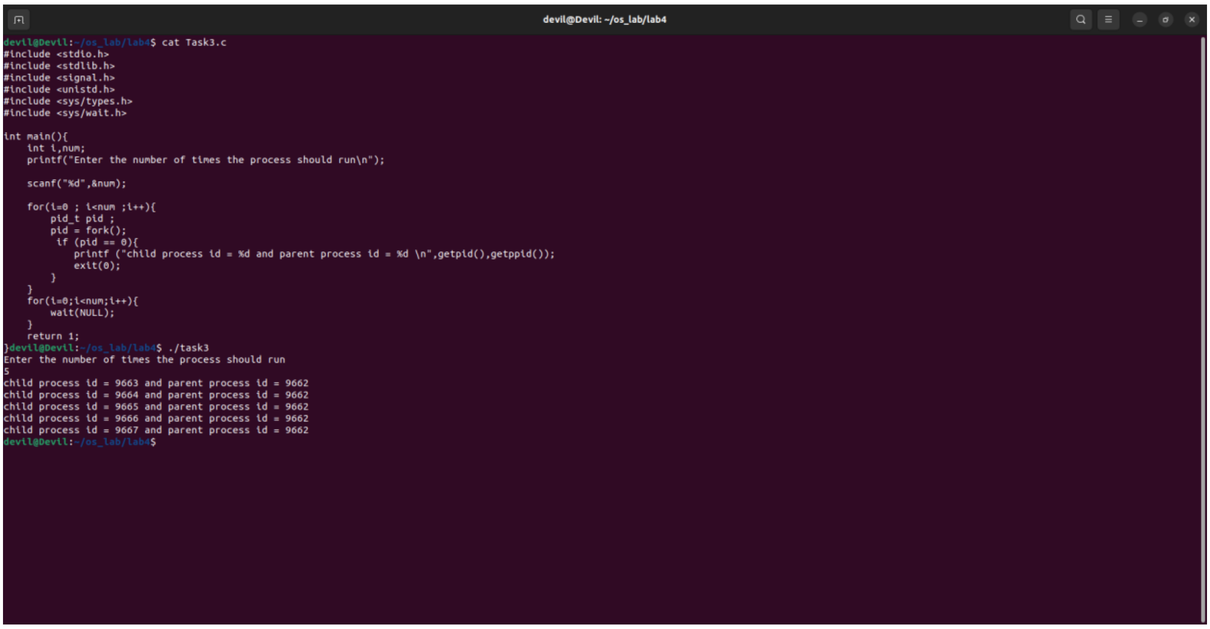
```
devil@Devil:~/os_lab/lab4$ cat Task1.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <signal.h>
#include <sys/wait.h>
#include <stdlib.h>

int main(){
    int pid ;
    pid = fork();

    if(pid < 0){
        printf("Child process failed");
        exit(1);
    }
    else if (pid == 0){
        printf("\nChild process is sucessful %d id\n",getpid());
        sleep(5);
        printf("\nChild process after 5 second sleep %d id",getpid());
    }
    else{
        printf("Parent process with id= %d ",getpid());
    }
    return 1;
}devil@Devil:~/os_lab/lab4$ ./task1
Parent process with id= 8447
Child process is sucessful 8448 id
devil@Devil:~/os_lab/lab4$
Child process after 5 second sleep 8448 id
```

TASK 2



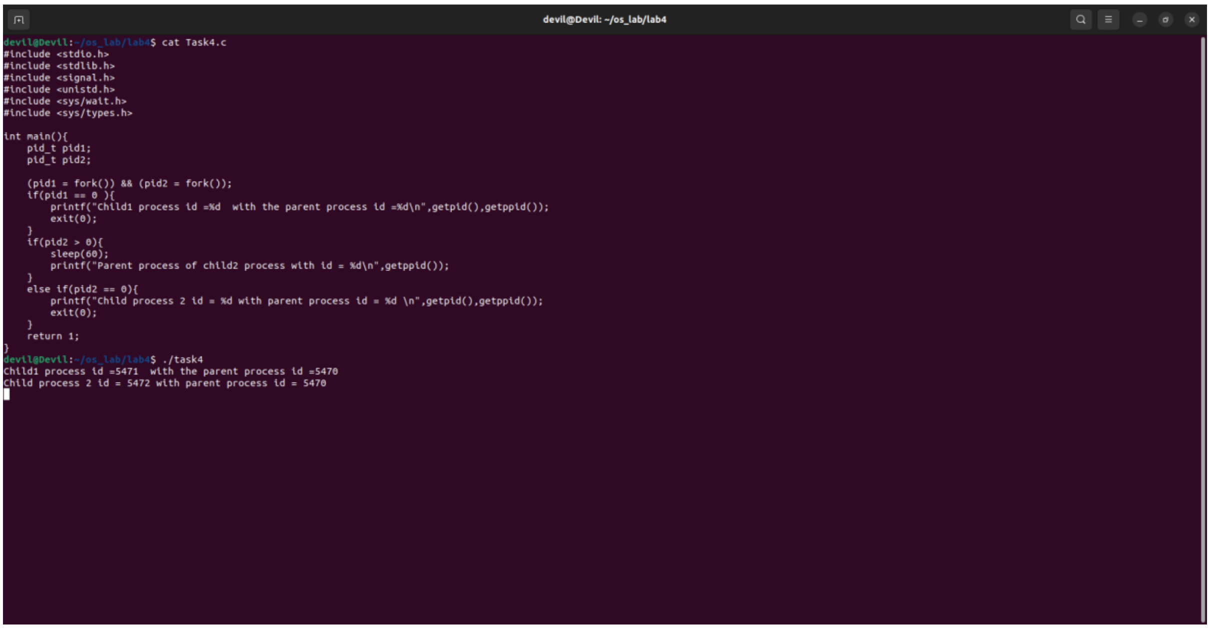
```
devil@Devil:~/os_lab/lab4$ cat Task2.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <signal.h>

int main(){
    pid_t pid ;
    pid = fork();

    if(pid < 0){
        printf("Child process unsuccessful");
    }
    else if(pid == 0){
        printf("\nChild\n");
    }
    else {
        printf("parent");
    }
}devil@Devil:~/os_lab/lab4$ ./task2
parent
Child
devil@Devil:~/os_lab/lab4$
```

TASK 3

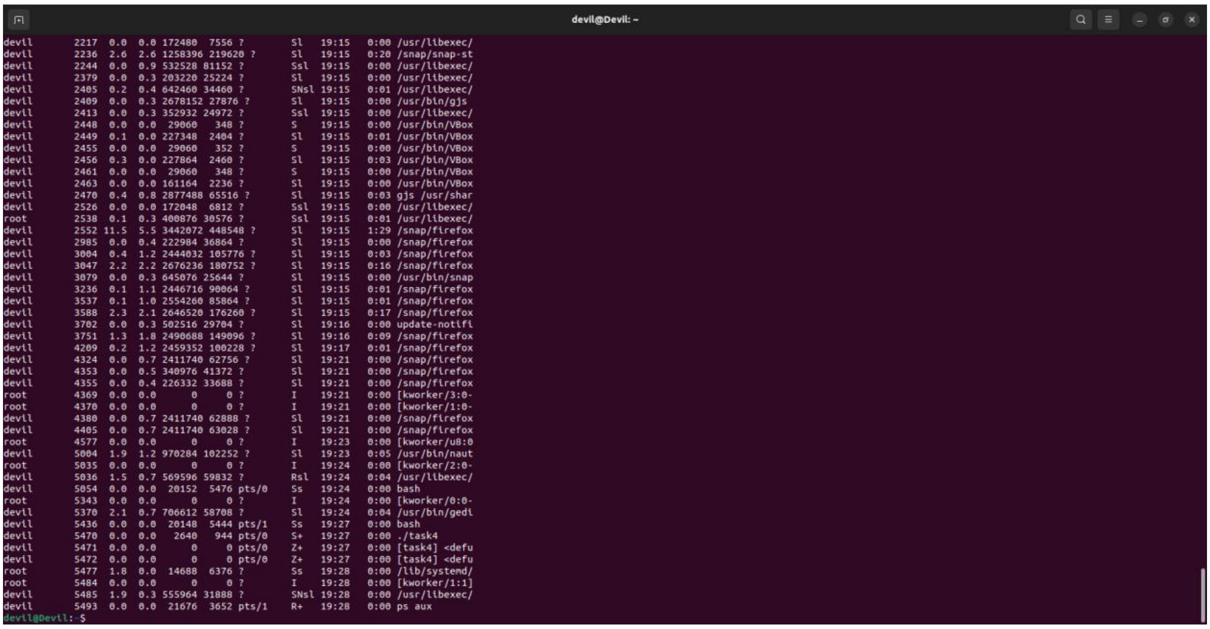


```
devil@Devil:~/os_lab/lab4$ cat Task3.c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int main(){
    int i,num;
    printf("Enter the number of times the process should run\n");

    scanf("%d",&num);

    for(i=0 ; i<num ;i++){
        pid_t pid ;
        pid = fork();
         if (pid == 0){
            printf ("child process id = %d and parent process id = %d \n",getpid(),getppid());
            exit(0);
        }
    }
    for(i=0;i<num;i++){
        wait(NULL);
    }
    return 1;
}devil@Devil:~/os_lab/lab4$ ./task3
Enter the number of times the process should run
5
child process id = 9663 and parent process id = 9662
child process id = 9664 and parent process id = 9662
child process id = 9665 and parent process id = 9662
child process id = 9666 and parent process id = 9662
child process id = 9667 and parent process id = 9662
devil@Devil:~/os_lab/lab4$
```

TASK 4

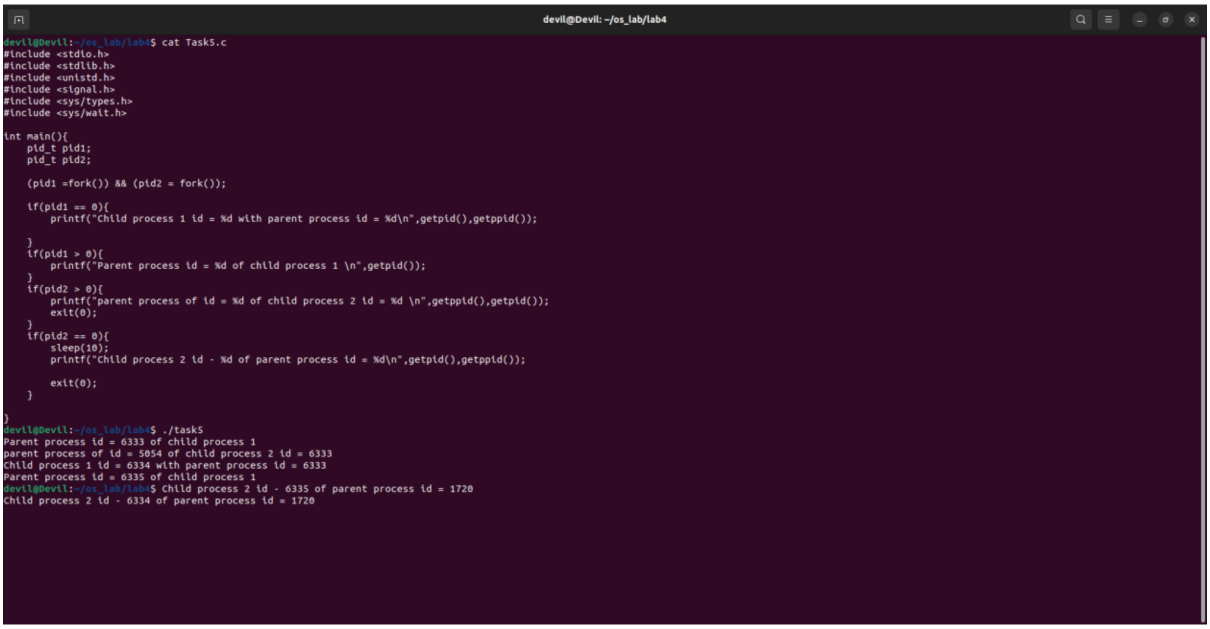


```
devil@Devil:~/os_lab/lab4$ cat Task4.c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>

int main(){
    pid_t pid1;
    pid_t pid2;

    (pid1 = fork()) && (pid2 = fork());
    if(pid1 == 0 ){
        printf("Child1 process id =%d  with the parent process id =%d\n",getpid(),getppid());
        exit(0);
    }
    if(pid2 > 0){
        sleep(60);
        printf("Parent process of child2 process with id = %d\n",getppid());
    }
    else if(pid2 == 0){
        printf("Child process 2 id = %d with parent process id = %d \n",getpid(),getppid());
        exit(0);
    }
    return 1;
}
devil@Devil:~/os_lab/lab4$ ./task4
Child1 process id =5471  with the parent process id =5470
Child process 2 id = 5472 with parent process id = 5470
```



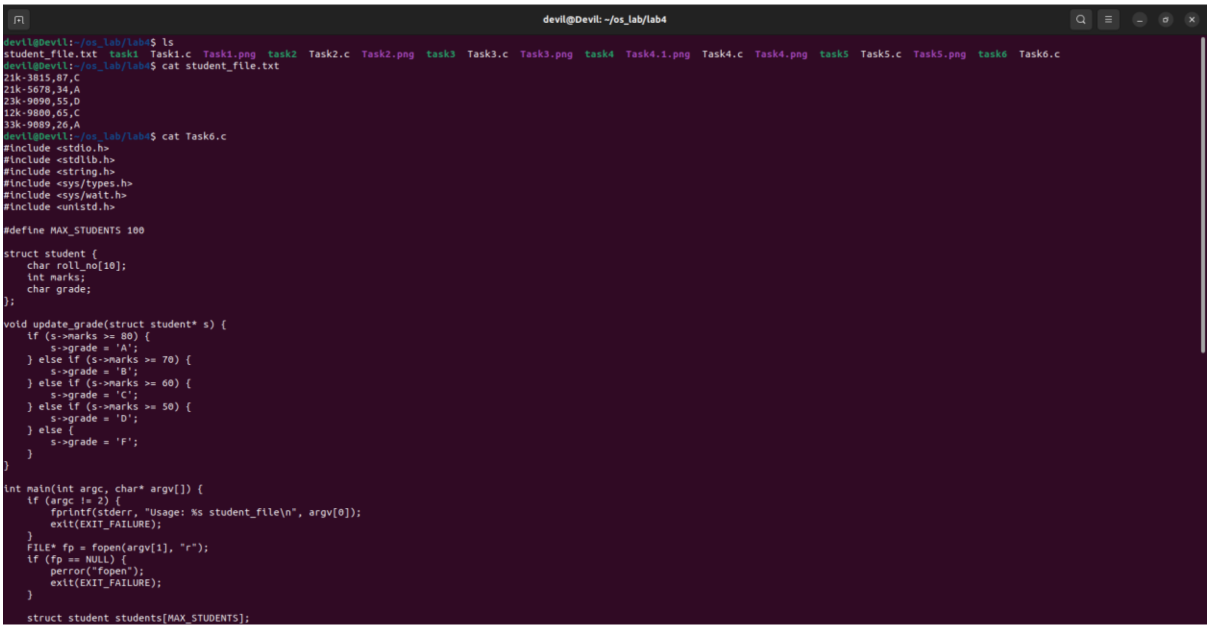
```
devil       2217  0.0  0.0 172480  7556 ?        Sl   19:15   0:00 /usr/libexec/
devil       2236  2.6  2.6 1258396 219620 ?      Sl   19:15   0:20 /snap/snap-st
devil       2244  0.0  0.9 532528 81152 ?        Ssl  19:15   0:00 /usr/libexec/
devil       2379  0.0  0.3 203220 25224 ?        Sl   19:15   0:00 /usr/libexec/
devil       2405  0.2  0.4 642460 34460 ?        SNsl 19:15   0:01 /usr/libexec/
devil       2409  0.0  0.3 2678152 27876 ?       Sl   19:15   0:00 /usr/bin/gjs
devil       2413  0.0  0.3 352932 24972 ?        Ssl  19:15   0:00 /usr/libexec/
devil       2448  0.0  0.0  29060   348 ?        S    19:15   0:00 /usr/bin/VBox
devil       2449  0.1  0.0 227348  2404 ?        Sl   19:15   0:01 /usr/bin/VBox
devil       2455  0.0  0.0  29060   352 ?        S    19:15   0:00 /usr/bin/VBox
devil       2456  0.3  0.0 227864  2460 ?        Sl   19:15   0:03 /usr/bin/VBox
devil       2461  0.0  0.0  29060   348 ?        S    19:15   0:00 /usr/bin/VBox
devil       2463  0.0  0.0 161164  2236 ?        Sl   19:15   0:00 /usr/bin/VBox
devil       2470  0.4  0.8 2877488 65516 ?       Sl   19:15   0:03 gjs /usr/shar
devil       2526  0.0  0.0 172048  6812 ?        Ssl  19:15   0:00 /usr/libexec/
root        2538  0.1  0.3 400876 30576 ?        Ssl  19:15   0:01 /usr/libexec/
devil       2552 11.5  5.5 3442072 448548 ?      Sl   19:15   1:29 /snap/firefox
devil       2985  0.0  0.4 222984 36864 ?        Sl   19:15   0:00 /snap/firefox
devil       3004  0.4  1.2 2444032 105776 ?      Sl   19:15   0:03 /snap/firefox
devil       3047  2.2  2.2 2676236 180752 ?      Sl   19:15   0:16 /snap/firefox
devil       3079  0.0  0.3 645076 25644 ?        Sl   19:15   0:00 /usr/bin/snap
devil       3236  0.1  1.1 2446716 90064 ?       Sl   19:15   0:01 /snap/firefox
devil       3537  0.1  1.0 2554260 85864 ?       Sl   19:15   0:01 /snap/firefox
devil       3588  2.3  2.1 2646520 176260 ?      Sl   19:15   0:17 /snap/firefox
devil       3702  0.0  0.3 502516 29704 ?        Sl   19:16   0:00 update-notifi
devil       3751  1.3  1.8 2490688 149096 ?      Sl   19:16   0:09 /snap/firefox
devil       4209  0.2  1.2 2459352 100228 ?      Sl   19:17   0:01 /snap/firefox
devil       4324  0.0  0.7 2411740 62756 ?       Sl   19:21   0:00 /snap/firefox
devil       4353  0.0  0.5 340976 41372 ?        Sl   19:21   0:00 /snap/firefox
devil       4355  0.0  0.4 226332 33688 ?        Sl   19:21   0:00 /snap/firefox
root        4369  0.0  0.0      0     0 ?        I    19:21   0:00 [kworker/3:0-
root        4370  0.0  0.0      0     0 ?        I    19:21   0:00 [kworker/1:0-
devil       4380  0.0  0.7 2411740 62888 ?       Sl   19:21   0:00 /snap/firefox
devil       4405  0.0  0.7 2411740 63028 ?       Sl   19:21   0:00 /snap/firefox
root        4577  0.0  0.0      0     0 ?        I    19:23   0:00 [kworker/u8:0
devil       5004  1.9  1.2 970284 102252 ?       Sl   19:23   0:05 /usr/bin/naut
root        5035  0.0  0.0      0     0 ?        I    19:24   0:00 [kworker/2:0-
devil       5036  1.5  0.7 569596 59832 ?        Rsl  19:24   0:04 /usr/libexec/
devil       5054  0.0  0.0  20152  5476 pts/0    Ss   19:24   0:00 bash
root        5343  0.0  0.0      0     0 ?        I    19:24   0:00 [kworker/0:0-
devil       5370  2.1  0.7 706612 58708 ?        Sl   19:24   0:04 /usr/bin/gedi
devil       5436  0.0  0.0  20148  5444 pts/1    Ss   19:27   0:00 bash
devil       5470  0.0  0.0   2640   944 pts/0    S+   19:27   0:00 ./task4
devil       5471  0.0  0.0      0     0 pts/0    Z+   19:27   0:00 [task4] <defu
devil       5472  0.0  0.0      0     0 pts/0    Z+   19:27   0:00 [task4] <defu
root        5477  1.8  0.0  14688  6376 ?        Ss   19:28   0:00 /lib/systemd/
root        5484  0.0  0.0      0     0 ?        I    19:28   0:00 [kworker/1:1]
devil       5485  1.9  0.3 555964 31888 ?        SNsl 19:28   0:00 /usr/libexec/
devil       5493  0.0  0.0  21676  3652 pts/1    R+   19:28   0:00 ps aux
devil@Devil:~$
```

TASK 5

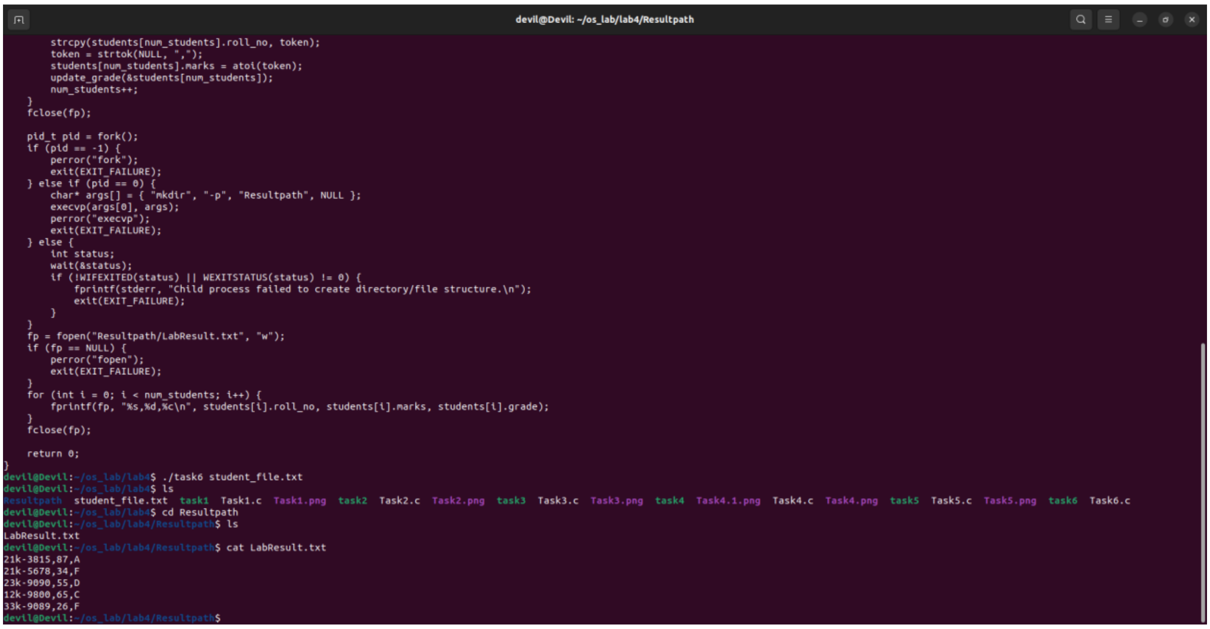


```
devil@Devil:~/os_lab/lab4$ cat Task5.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/wait.h>

int main(){
    pid_t pid1;
    pid_t pid2;

    (pid1 =fork()) && (pid2 = fork());

    if(pid1 == 0){
        printf("Child process 1 id = %d with parent process id = %d\n",getpid(),getppid());

    }
    if(pid1 > 0){
        printf("Parent process id = %d of child process 1 \n",getpid());
    }
    if(pid2 > 0){
        printf("parent process of id = %d of child process 2 id = %d \n",getppid(),getpid());
        exit(0);
    }
    if(pid2 == 0){
        sleep(10);
        printf("Child process 2 id - %d of parent process id = %d\n",getpid(),getppid());

        exit(0);
    }

}
devil@Devil:~/os_lab/lab4$ ./task5
Parent process id = 6333 of child process 1
parent process of id = 5054 of child process 2 id = 6333
Child process 1 id = 6334 with parent process id = 6333
Parent process id = 6335 of child process 1
devil@Devil:~/os_lab/lab4$ Child process 2 id - 6335 of parent process id = 1720
Child process 2 id - 6334 of parent process id = 1720
```

TASK 6



```
devil@Devil:~/os_lab/lab4$ ls
student_file.txt  task1  Task1.c  Task1.png  task2  Task2.c  Task2.png  task3  Task3.c  Task3.png  task4  Task4.1.png  Task4.c  Task4.png  task5  Task5.c  Task5.png  task6  Task6.c
devil@Devil:~/os_lab/lab4$ cat student_file.txt
21k-3815,87,C
21k-5678,34,A
23k-9090,55,D
12k-9800,65,C
33k-9089,26,A
devil@Devil:~/os_lab/lab4$ cat Task6.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

#define MAX_STUDENTS 100

struct student {
    char roll_no[10];
    int marks;
    char grade;
};

void update_grade(struct student* s) {
    if (s->marks >= 80) {
        s->grade = 'A';
    } else if (s->marks >= 70) {
        s->grade = 'B';
    } else if (s->marks >= 60) {
        s->grade = 'C';
    } else if (s->marks >= 50) {
        s->grade = 'D';
    } else {
        s->grade = 'F';
    }
}

int main(int argc, char* argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s student_file\n", argv[0]);
        exit(EXIT_FAILURE);
    }
    FILE* fp = fopen(argv[1], "r");
    if (fp == NULL) {
        perror("fopen");
        exit(EXIT_FAILURE);
    }

    struct student students[MAX_STUDENTS];
```



```
        strcpy(students[num_students].roll_no, token);
        token = strtok(NULL, ",");
        students[num_students].marks = atoi(token);
        update_grade(&students[num_students]);
        num_students++;
    }
    fclose(fp);

    pid_t pid = fork();
    if (pid == -1) {
        perror("fork");
        exit(EXIT_FAILURE);
    } else if (pid == 0) {
        char* args[] = { "mkdir", "-p", "Resultpath", NULL };
        execvp(args[0], args);
        perror("execvp");
        exit(EXIT_FAILURE);
    } else {
        int status;
        wait(&status);
        if (!WIFEXITED(status) || WEXITSTATUS(status) != 0) {
            fprintf(stderr, "Child process failed to create directory/file structure.\n");
            exit(EXIT_FAILURE);
        }
    }
    fp = fopen("Resultpath/LabResult.txt", "w");
    if (fp == NULL) {
        perror("fopen");
        exit(EXIT_FAILURE);
    }
    for (int i = 0; i < num_students; i++) {
        fprintf(fp, "%s,%d,%c\n", students[i].roll_no, students[i].marks, students[i].grade);
    }
    fclose(fp);

    return 0;
}
devil@Devil:~/os_lab/lab4$ ./task6 student_file.txt
devil@Devil:~/os_lab/lab4$ ls
Resultpath  student_file.txt  task1  Task1.c  Task1.png  task2  Task2.c  Task2.png  task3  Task3.c  Task3.png  task4  Task4.1.png  Task4.c  Task4.png  task5  Task5.c  Task5.png  task6  Task6.c
devil@Devil:~/os_lab/lab4$ cd Resultpath
devil@Devil:~/os_lab/lab4/Resultpath$ ls
LabResult.txt
devil@Devil:~/os_lab/lab4/Resultpath$ cat LabResult.txt
21k-3815,87,A
21k-5678,34,F
23k-9090,55,D
12k-9800,65,C
33k-9089,26,F
devil@Devil:~/os_lab/lab4/Resultpath$
```