The exec Function

LAB # 04

Fall 2023

CSE-302L Systems Programming Lab

Submitted by: **Ali Asghar**

Registration No.: **21PWCSE2059**

Class Section: **C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Abdullah Hamid

Date:

1st February 2024

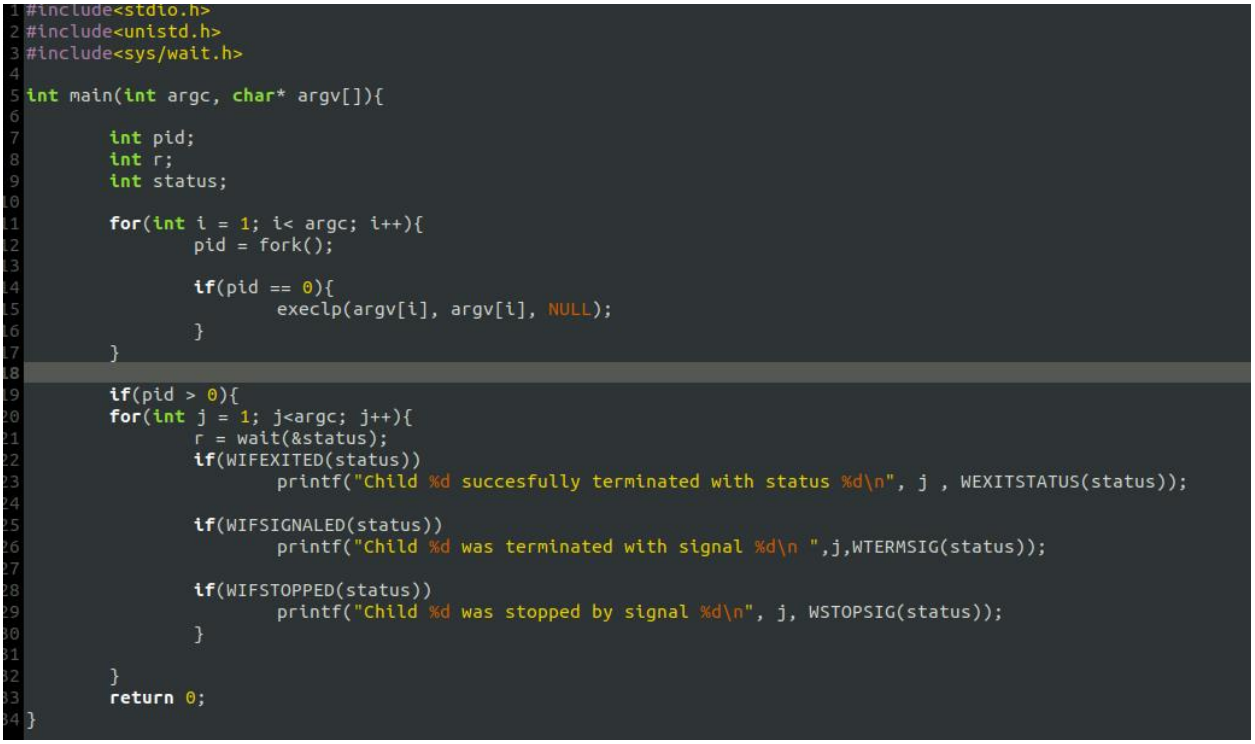
Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task 1:

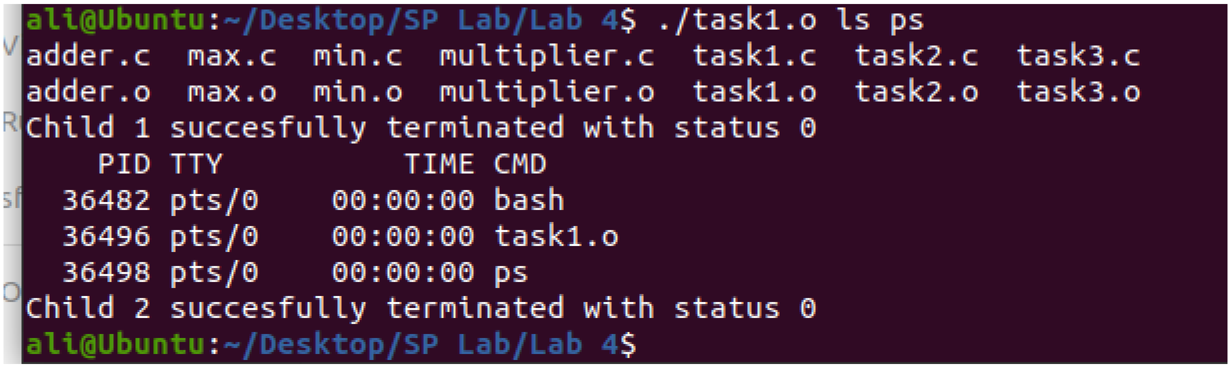
Write a program that takes N UNIX commands as arguments, creates N child processes, each of them implementing their respective commands. Parent process shall wait for all the child processes and receive and print the exit status of the child processes.

Code:



```c
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>

int main(int argc, char* argv[]){

        int pid;
        int r;
        int status;

        for(int i = 1; i< argc; i++){
                pid = fork();

                if(pid == 0){
                        execlp(argv[i], argv[i], NULL);
                }
        }

        if(pid > 0){
        for(int j = 1; j<argc; j++){
                r = wait(&status);
                if(WIFEXITED(status))
                        printf("Child %d succesfully terminated with status %d\n", j , WEXITSTATUS(status));

                if(WIFSIGNALED(status))
                        printf("Child %d was terminated with signal %d\n ",j,WTERMSIG(status));

                if(WIFSTOPPED(status))
                        printf("Child %d was stopped by signal %d\n", j, WSTOPSIG(status));
                }

        }
        return 0;
}
```

Output:



```
ali@Ubuntu:~/Desktop/SP Lab/Lab 4$ ./task1.o ls ps
adder.c   max.c  min.c  multiplier.c  task1.c  task2.c  task3.c
adder.o   max.o  min.o  multiplier.o  task1.o  task2.o  task3.o
Child 1 succesfully terminated with status 0
    PID TTY          TIME CMD
  36482 pts/0    00:00:00 bash
  36496 pts/0    00:00:00 task1.o
  36498 pts/0    00:00:00 ps
Child 2 succesfully terminated with status 0
ali@Ubuntu:~/Desktop/SP Lab/Lab 4$
```

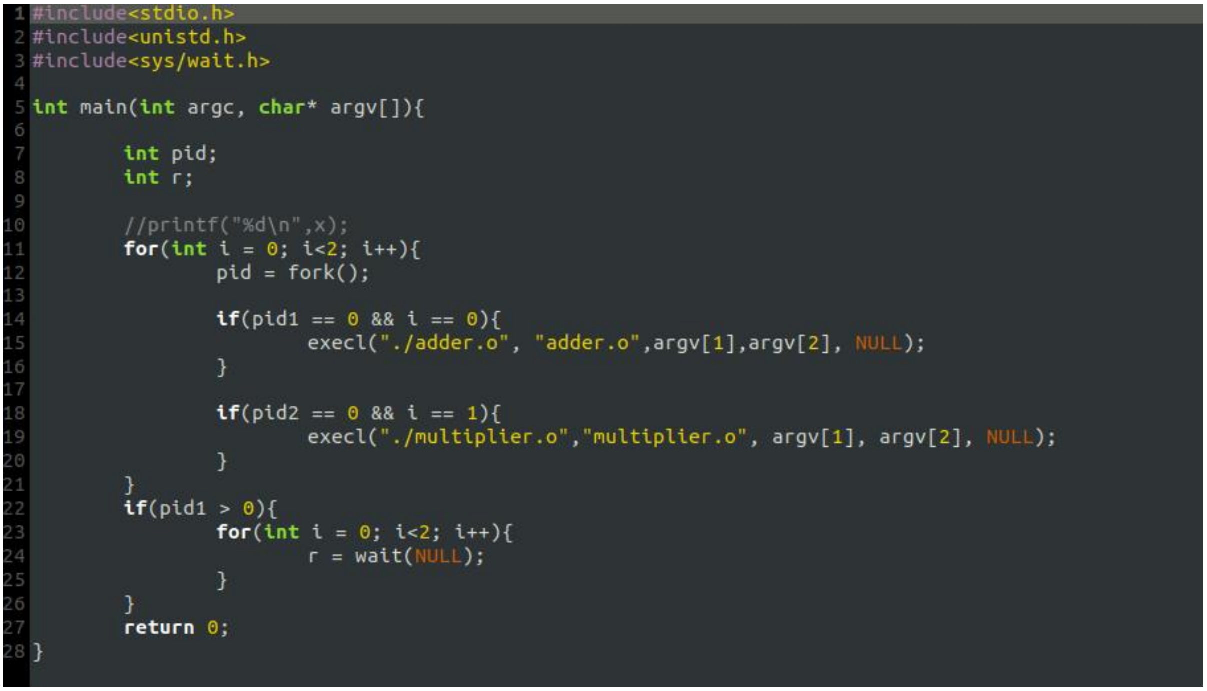
Task 2:

a. Write a program that takes integers as arguments and adds them.

b. Write a program that takes integers as arguments and multiplies them.

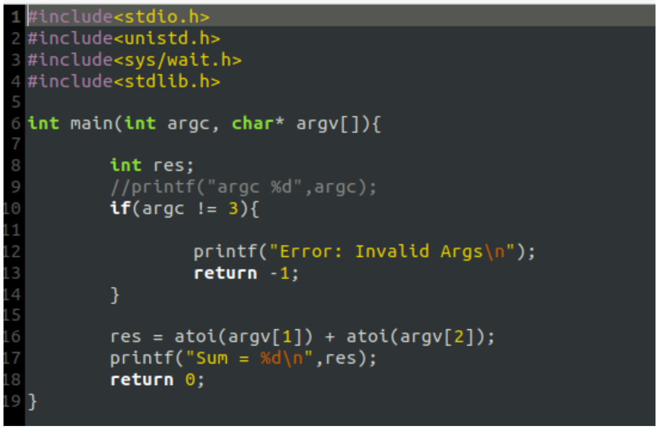
c. Write a program that takes integers as arguments & adds & multiplies them using the above two programs.

Code:

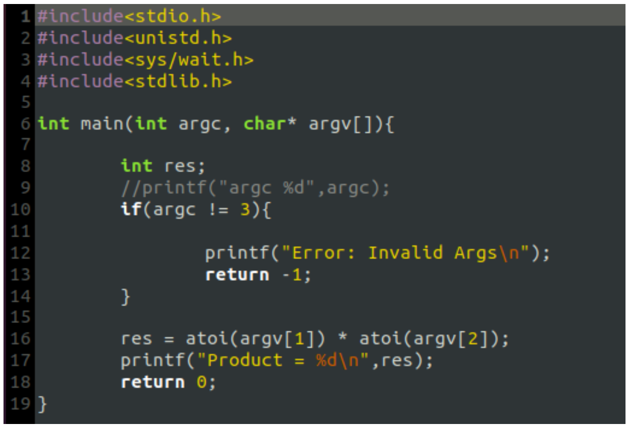


```c
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>

int main(int argc, char* argv[]){

        int pid;
        int r;

        //printf("%d\n",x);
        for(int i = 0; i<2; i++){
                pid = fork();

                if(pid1 == 0 && i == 0){
                        execl("./adder.o", "adder.o",argv[1],argv[2], NULL);
                }

                if(pid2 == 0 && i == 1){
                        execl("./multiplier.o","multiplier.o", argv[1], argv[2], NULL);
                }
        }
        if(pid1 > 0){
                for(int i = 0; i<2; i++){
                        r = wait(NULL);
                }
        }
        return 0;
}
```

Task2.c



```c
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>

int main(int argc, char* argv[]){

        int res;
        //printf("argc %d",argc);
        if(argc != 3){

                printf("Error: Invalid Args\n");
                return -1;
        }

        res = atoi(argv[1]) + atoi(argv[2]);
        printf("Sum = %d\n",res);
        return 0;
}
```

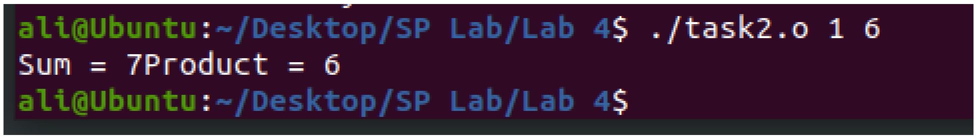


```c
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>

int main(int argc, char* argv[]){

        int res;
        //printf("argc %d",argc);
        if(argc != 3){

                printf("Error: Invalid Args\n");
                return -1;
        }

        res = atoi(argv[1]) * atoi(argv[2]);
        printf("Product = %d\n",res);
        return 0;
}
```

adder.c

multiplier.c

Output:

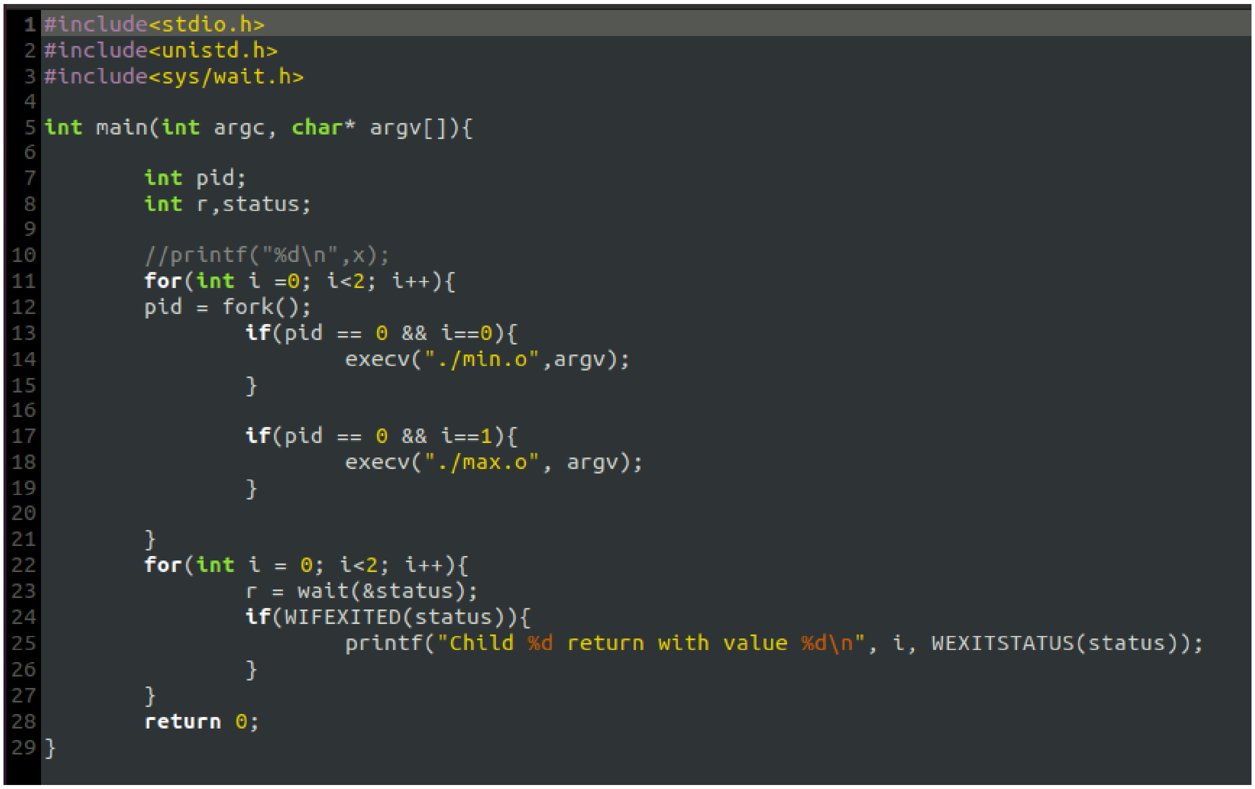


```
ali@Ubuntu:~/Desktop/SP Lab/Lab 4$ ./task2.o 1 6
Sum = 7Product = 6
ali@Ubuntu:~/Desktop/SP Lab/Lab 4$
```

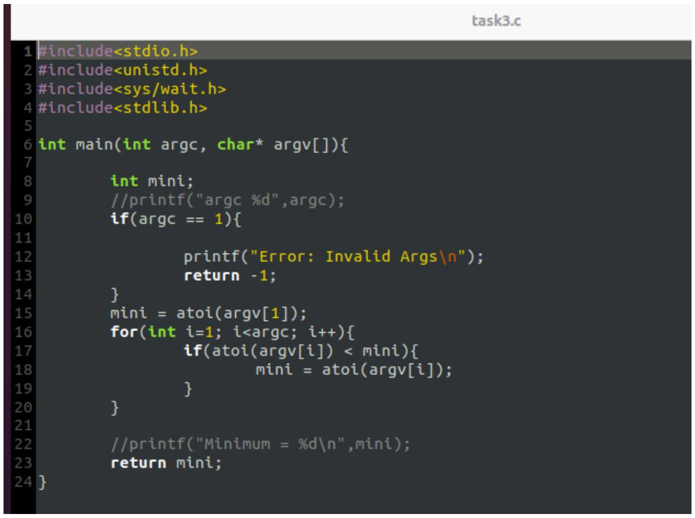
Task 3:

Write a program **"minmax.c"** that takes an array as command line arguments. Program executes **min.c** and **max.c** programs in its two child processes. One child process calculates and returns the min value and other calculates and returns the max value in the array. The program **"minmax.c"** shall receive the values returned by the child processes and display these values.

Code:

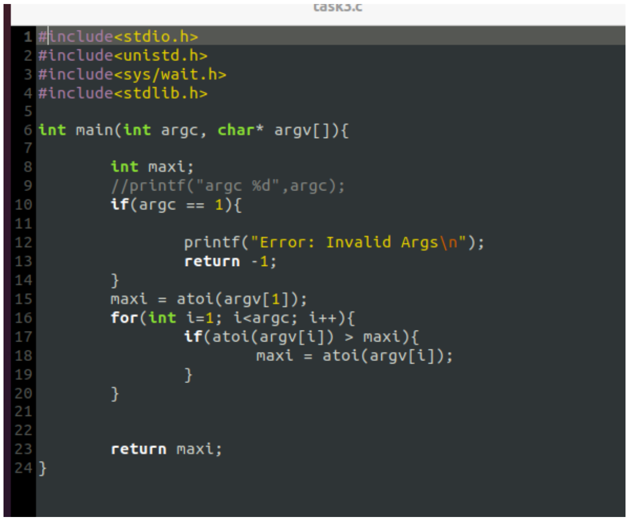


```c
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>

int main(int argc, char* argv[]){

        int pid;
        int r,status;

        //printf("%d\n",x);
        for(int i =0; i<2; i++){
        pid = fork();
                if(pid == 0 && i==0){
                        execv("./min.o",argv);
                }

                if(pid == 0 && i==1){
                        execv("./max.o", argv);
                }

        }
        for(int i = 0; i<2; i++){
                r = wait(&status);
                if(WIFEXITED(status)){
                        printf("Child %d return with value %d\n", i, WEXITSTATUS(status));
                }
        }
        return 0;
}
```



task3.c

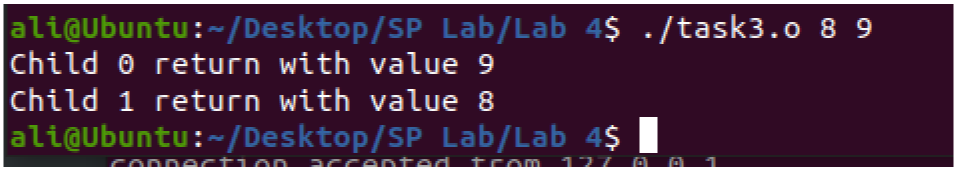
```
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>

int main(int argc, char* argv[]){

        int mini;
        //printf("argc %d",argc);
        if(argc == 1){

                printf("Error: Invalid Args\n");
                return -1;
        }
        mini = atoi(argv[1]);
        for(int i=1; i<argc; i++){
                if(atoi(argv[i]) < mini){
                        mini = atoi(argv[i]);
                }
        }

        //printf("Minimum = %d\n",mini);
        return mini;
}
```



task3.c

```
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>

int main(int argc, char* argv[]){

        int maxi;
        //printf("argc %d",argc);
        if(argc == 1){

                printf("Error: Invalid Args\n");
                return -1;
        }
        maxi = atoi(argv[1]);
        for(int i=1; i<argc; i++){
                if(atoi(argv[i]) > maxi){
                        maxi = atoi(argv[i]);
                }
        }


        return maxi;
}
```

Output:



```
ali@Ubuntu:~/Desktop/SP Lab/Lab 4$ ./task3.o 8 9
Child 0 return with value 9
Child 1 return with value 8
ali@Ubuntu:~/Desktop/SP Lab/Lab 4$ 
   connection accepted from 127.0.0.1
```