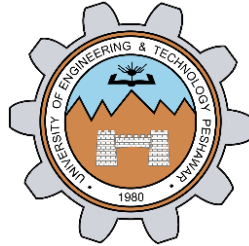


# **Command Line Arguments**

**LAB # 07**



**Spring 2023**

**CSE-204L Operating Systems Lab**

Submitted by: **Suleman Shah**

Registration No.: **21PWCSE1983**

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. Madiha Sher**

Date:

**19<sup>th</sup> May 2023**

**Department of Computer Systems Engineering**  
**University of Engineering and Technology, Peshawar**

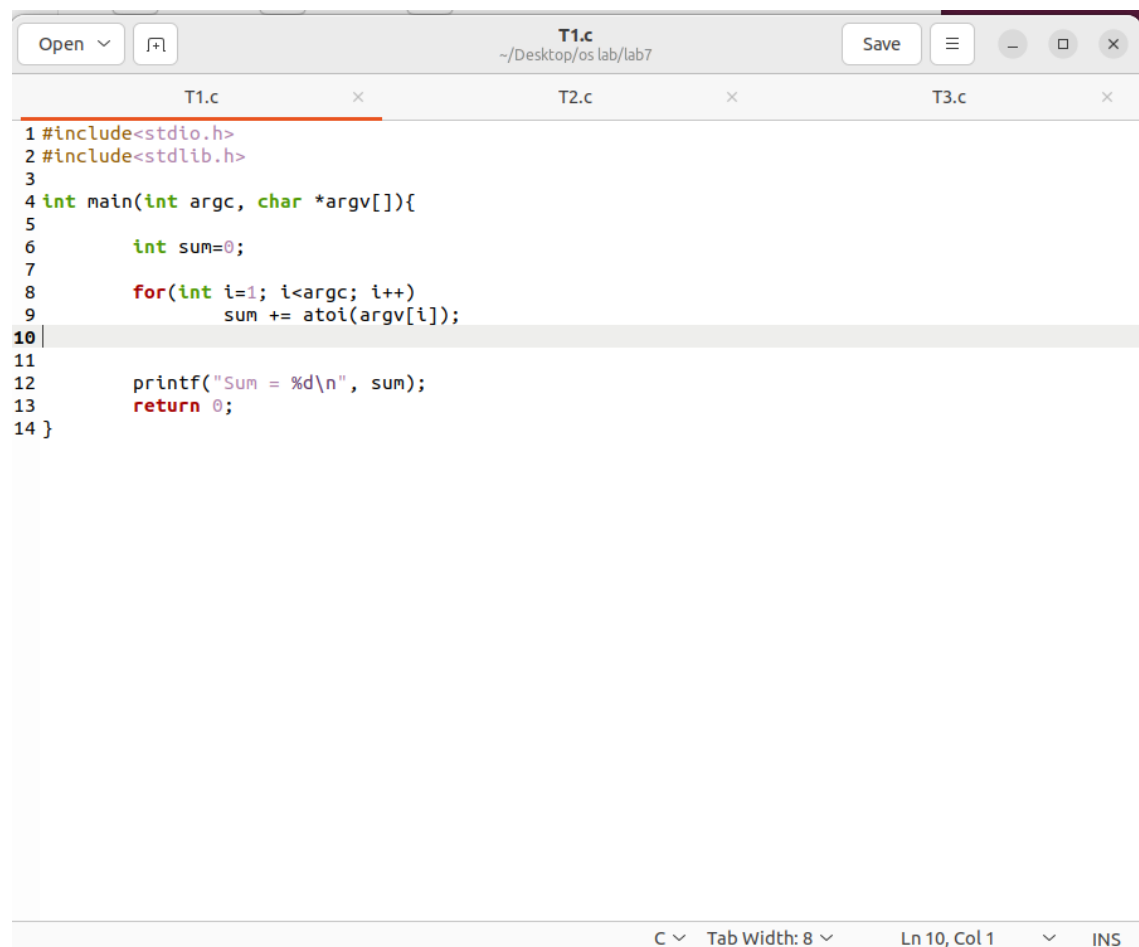
## Objectives:

- To comprehend the concept of command-line parameters and their usage in C programming.
- To acquire knowledge on accessing and manipulating command-line parameters in a C program.
- To delve into process creation and execution using command-line parameters.

## Task 1:

Write a C program that finds the sum of all CLA's.

## Code Screenshot:

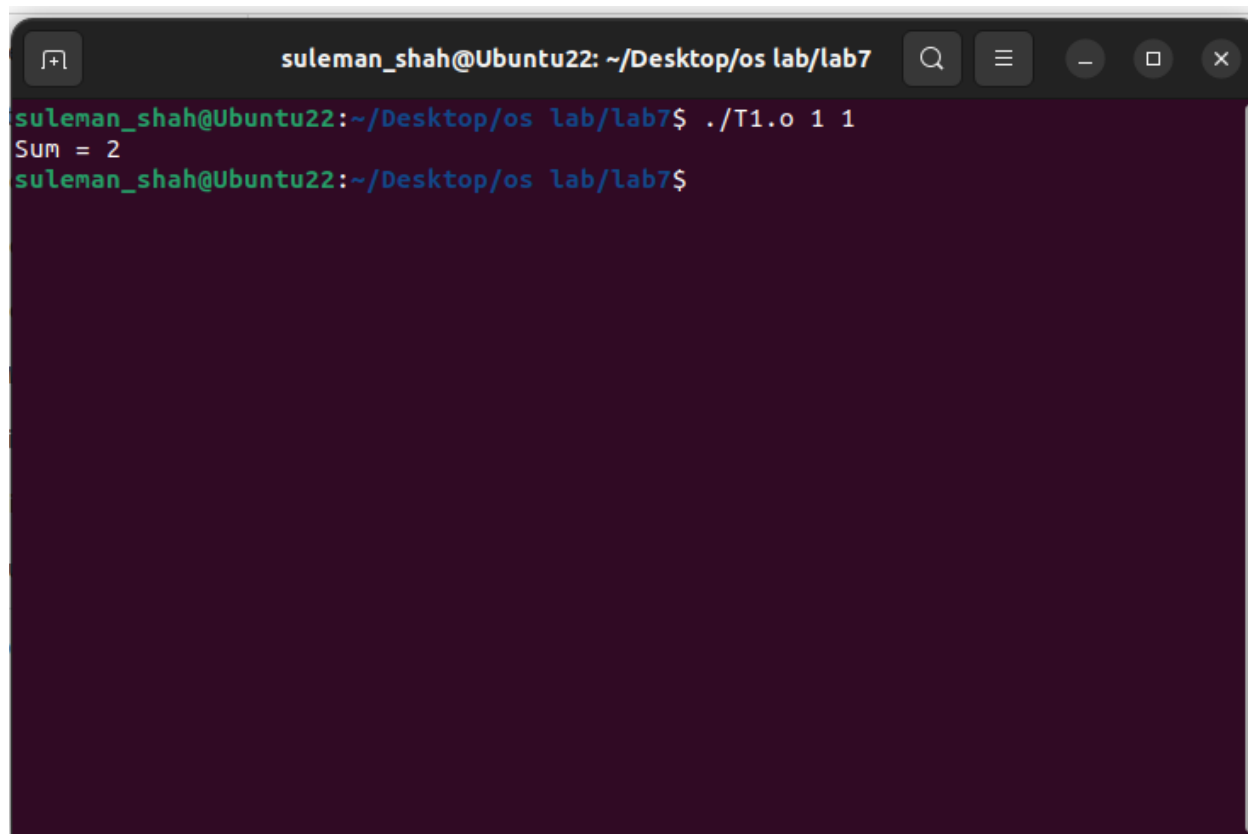


The screenshot shows a code editor window with three tabs: T1.c, T2.c, and T3.c. The T1.c tab is active and contains the following C code:

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main(int argc, char *argv[]){
5
6     int sum=0;
7
8     for(int i=1; i<argc; i++)
9         sum += atoi(argv[i]);
10
11
12     printf("Sum = %d\n", sum);
13     return 0;
14 }
```

The editor interface includes a top bar with 'Open', 'Save', and window control buttons. The bottom status bar shows 'C', 'Tab Width: 8', 'Ln 10, Col 1', and 'INS'.

## Output Screenshot:



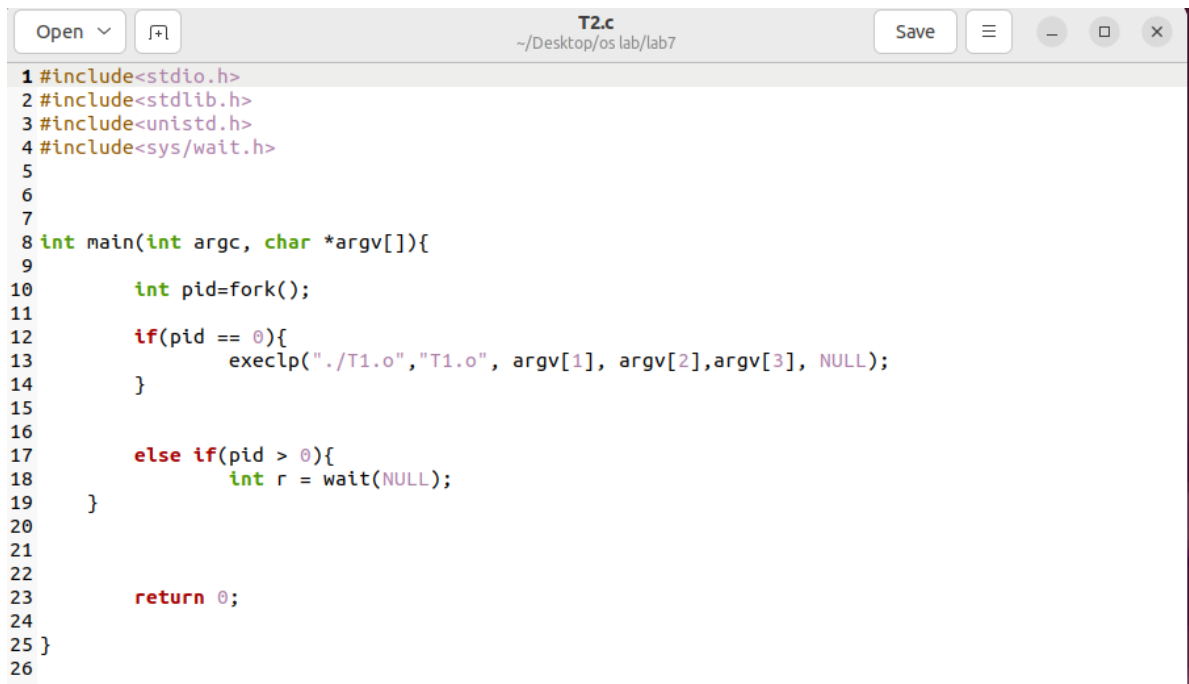
```
suleman_shah@Ubuntu22: ~/Desktop/os lab/lab7
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$ ./T1.o 1 1
Sum = 2
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$
```

The screenshot shows a terminal window with a dark background. The title bar at the top reads "suleman\_shah@Ubuntu22: ~/Desktop/os lab/lab7". The terminal content shows a user prompt "suleman\_shah@Ubuntu22:~/Desktop/os lab/lab7\$" followed by the command "./T1.o 1 1". The output of the command is "Sum = 2". Below the output, the prompt "suleman\_shah@Ubuntu22:~/Desktop/os lab/lab7\$" is shown again, indicating the command has completed.

## Task 2:

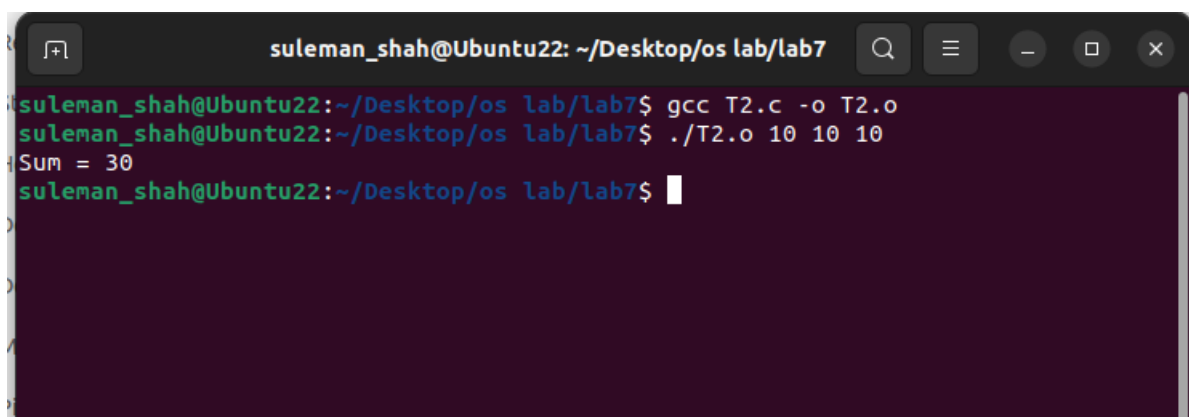
Write a C program that creates a child process & execute task 1 in child process using `execlp()` system call. Parent process shall wait for the child process.

### Code Screenshot:



```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<sys/wait.h>
5
6
7
8 int main(int argc, char *argv[]){
9
10     int pid=fork();
11
12     if(pid == 0){
13         execlp("./T1.o", "T1.o", argv[1], argv[2],argv[3], NULL);
14     }
15
16     else if(pid > 0){
17         int r = wait(NULL);
18     }
19
20
21
22     return 0;
23 }
24
25
26
```

### Output Screenshot:

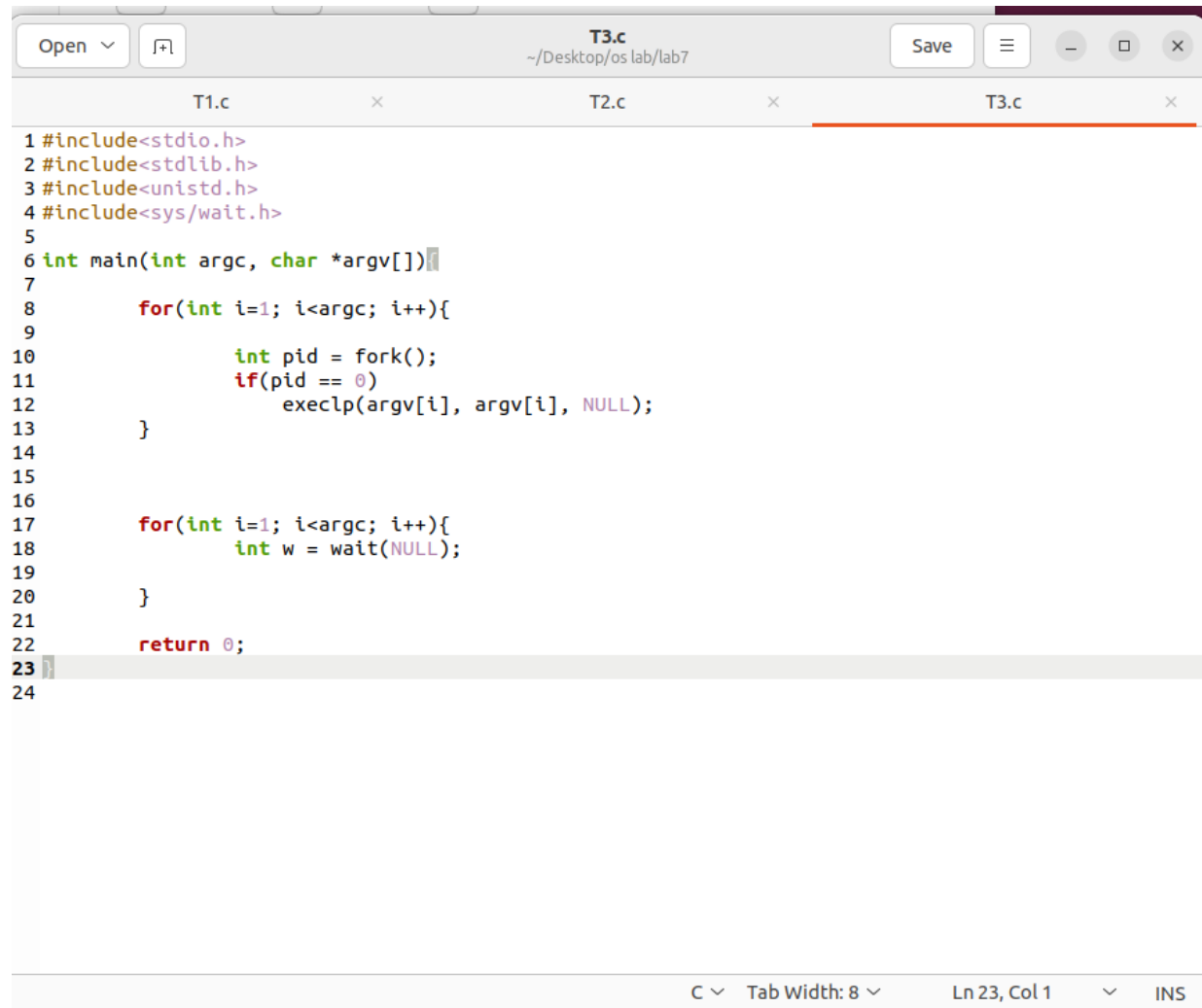


```
suleman_shah@Ubuntu22: ~/Desktop/os lab/lab7
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$ gcc T2.c -o T2.o
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$ ./T2.o 10 10 10
Sum = 30
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$
```

### Task 3:

Write a C program that takes built-in command on CLA's and create separate child process for each command & execute these commands in child process. Parent shall wait for the child processes.

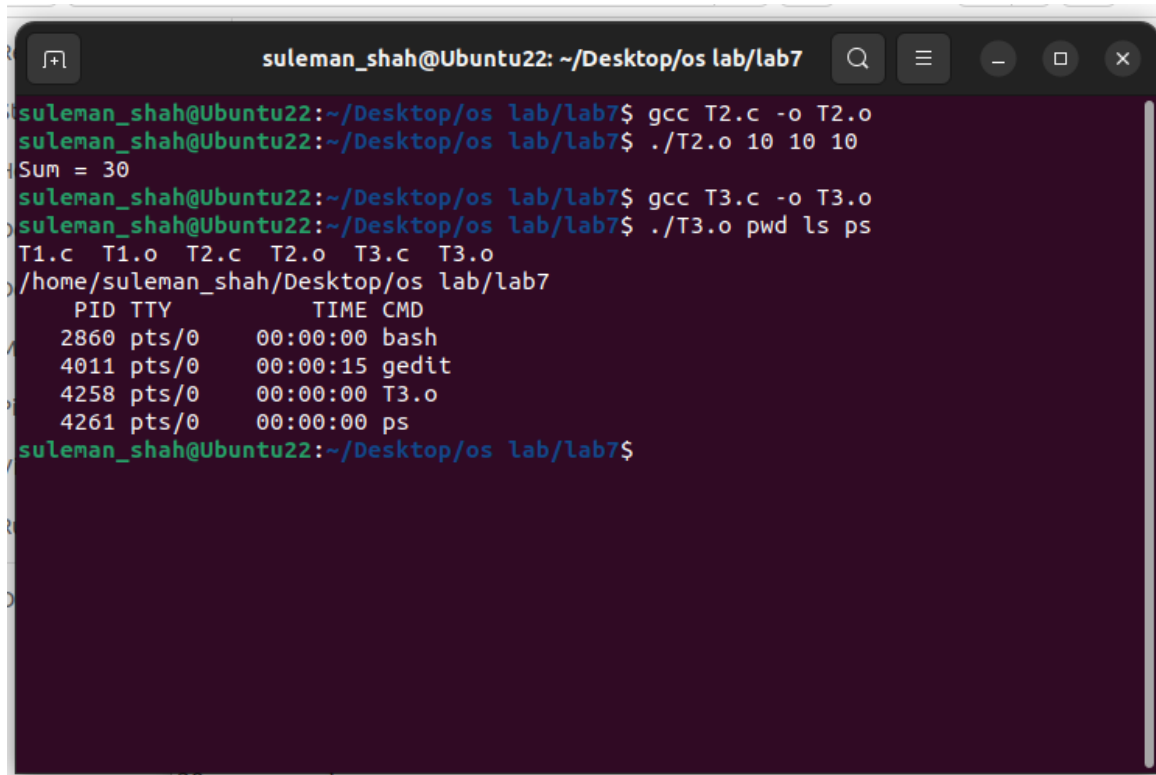
### Code Screenshot:



```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<sys/wait.h>
5
6 int main(int argc, char *argv[])
7 {
8     for(int i=1; i<argc; i++){
9         int pid = fork();
10        if(pid == 0)
11            execlp(argv[i], argv[i], NULL);
12    }
13
14    for(int i=1; i<argc; i++){
15        int w = wait(NULL);
16    }
17
18    return 0;
19 }
```

The screenshot shows a code editor window titled 'T3.c' with the file path '~/Desktop/os lab/lab7'. The editor contains a C program that forks child processes to execute shell commands. The code includes headers for stdio, stdlib, unistd, and sys/wait. The main function iterates over the command-line arguments, forking a child process for each. The child process executes the command using execlp, and the parent process waits for all child processes to complete before returning 0.

## Output Screenshot:



```
suleman_shah@Ubuntu22: ~/Desktop/os lab/lab7
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$ gcc T2.c -o T2.o
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$ ./T2.o 10 10 10
Sum = 30
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$ gcc T3.c -o T3.o
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$ ./T3.o pwd ls ps
T1.c  T1.o  T2.c  T2.o  T3.c  T3.o
/home/suleman_shah/Desktop/os lab/lab7
  PID TTY          TIME CMD
  2860 pts/0        00:00:00 bash
  4011 pts/0        00:00:15 gedit
  4258 pts/0        00:00:00 T3.o
  4261 pts/0        00:00:00 ps
suleman_shah@Ubuntu22:~/Desktop/os lab/lab7$
```