OPERATING SYSTEMS LAB 07

COMMAND LINE ARGUMENTS

Spring 2024

Submitted by: **Hassan Zaib Jadoon**

Registration No: **22pwsce2144**

Class Section: **A**

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

Student Signature: Mohsin Sajjad

Submitted to:

Engr. Abdullah Hamid

Month Day, Year (03 May, 2024)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

OBJECTIVES:

- Understand what command line arguments are.
- Learn how to use command line arguments to control the behavior of a program.
- Practice writing programs that accept and use command line arguments.

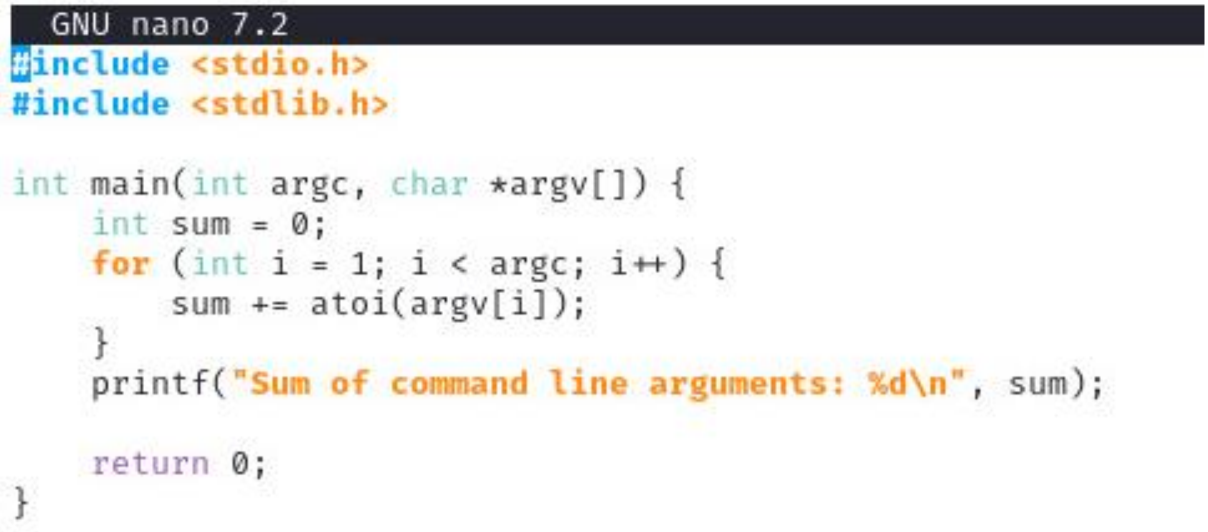
Title:

COMMAND LINE ARGUMENTS

Task1:

Write a program to calculate the sum of CLAs.

Code:

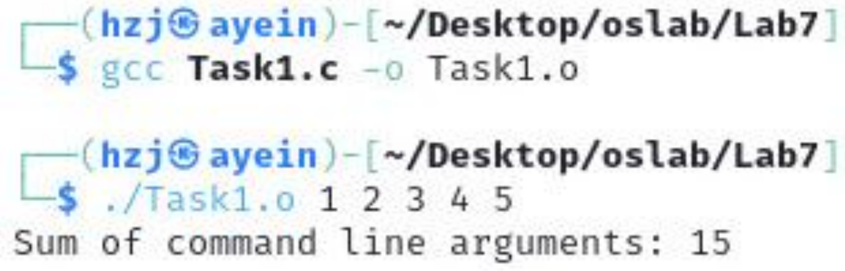


```
  GNU nano 7.2
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    int sum = 0;
    for (int i = 1; i < argc; i++) {
        sum += atoi(argv[i]);
    }
    printf("Sum of command line arguments: %d\n", sum);

    return 0;
}
```

OUTPUT:



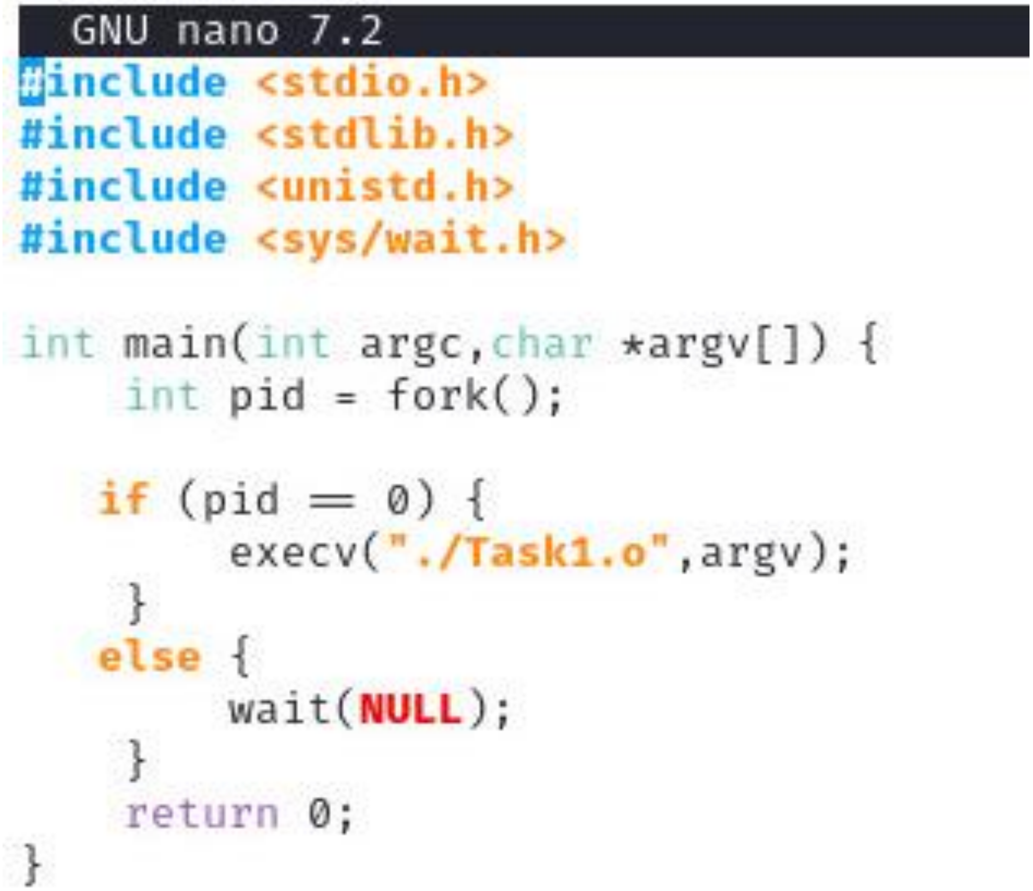
```
┌──(hzj㉿ayein)-[~/Desktop/oslab/Lab7]
└─$ gcc Task1.c -o Task1.o

┌──(hzj㉿ayein)-[~/Desktop/oslab/Lab7]
└─$ ./Task1.o 1 2 3 4 5
Sum of command line arguments: 15
```

Task2:

Write a program that create a child process and execute Task1 in child process. Parent process shall wait for child process termination.

Code:

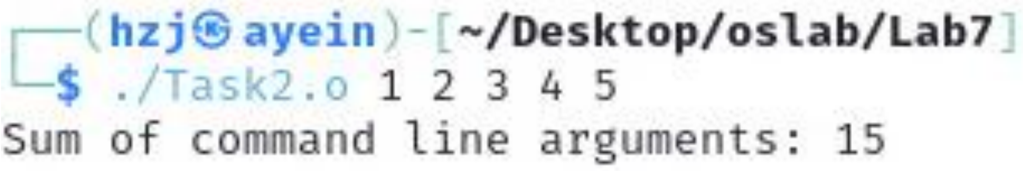


```
  GNU nano 7.2
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main(int argc,char *argv[]) {
    int pid = fork();

   if (pid == 0) {
        execv("./Task1.o",argv);
    }
   else {
        wait(NULL);
    }
    return 0;
}
```

Output:

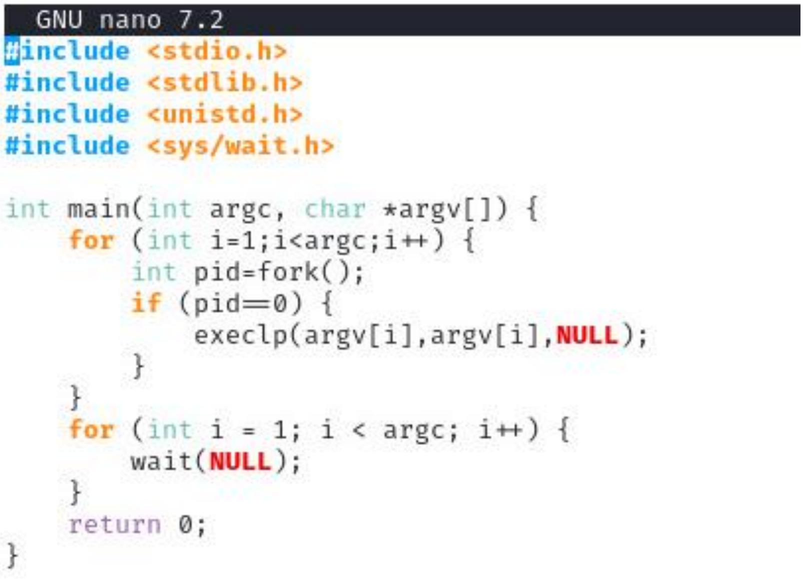


```
┌──(hzj㉿ayein)-[~/Desktop/oslab/Lab7]
└─$ ./Task2.o 1 2 3 4 5
Sum of command line arguments: 15
```

TASK 03:

Write a program that takes a list of commands as CLAs. Create a separate child for each command and execute that command in child process. Parent process shall wait for all child process.

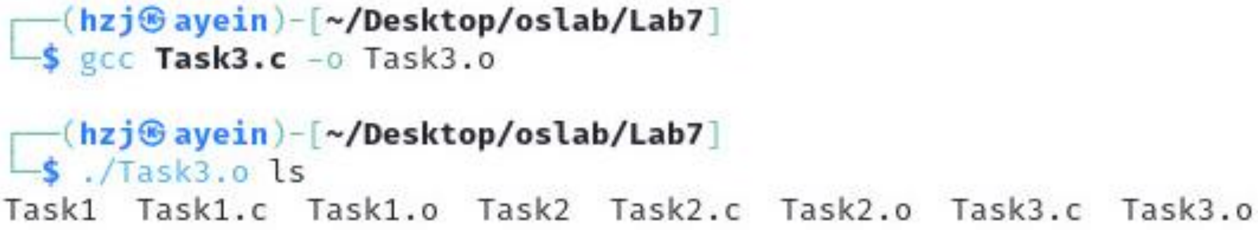
Code:



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

int main(int argc, char *argv[]) {
    for (int i=1;i<argc;i++) {
        int pid=fork();
        if (pid==0) {
            execlp(argv[i],argv[i],NULL);
        }
    }
    for (int i = 1; i < argc; i++) {
        wait(NULL);
    }
    return 0;
}
```

Output:



```
┌──(hzj㉿ayein)-[~/Desktop/oslab/Lab7]
└─$ gcc Task3.c -o Task3.o

┌──(hzj㉿ayein)-[~/Desktop/oslab/Lab7]
└─$ ./Task3.o ls
Task1   Task1.c   Task1.o   Task2   Task2.c   Task2.o   Task3.c   Task3.o
```

CSE 302L: Operating Systems Lab

LAB ASSESSMENT RUBRICS

Marking Criteria	**Exceeds expectation (2.5)**	**Meets expectation (1.5)**	**Does not meet expectation (0)**	**Score**
1. Correctness	Program compiles (no errors and no warnings). Program always works correctly and meets the specification(s). Completed between 81-100% of the requirements.	Program compiles (no errors and some warnings). Some details of the program specification are violated, program functions incorrectly for some inputs. Completed between 41-80% of the requirements.	Program fails to or compile with lots of warnings. Program only functions correctly in very limited cases or not at all. Completed less than 40% of the requirements.	
2. Delivery	Delivered on time, and in correct format (disk, email, hard copy etc.)	Not delivered on time, or slightly incorrect format.	Not delivered on time or not in correct format.	
3. Coding Standards	Proper indentation, whitespace, line length, wrapping, comments and references.	Missing some of whitespace, line length, wrapping, comments or references.	Poor use of whitespace, line length, wrapping, comments and references.	
4. Presentation of document	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting and excellently organized.	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting.	No name, date, or assignment title included. No task titles, no objectives, no output screenshots, poor formatting.	

Instructor:

Name: Engr. Abdullah Hamid   Signature: ______________________