LAB #06

Process Creation, Execution and Termination

Spring 2023

CSE-204L Operating Systems Lab

Submitted by: MUHAMMAD SADEEQ

Registration No.: 21PWCSE2028

Section: C

“On my honor, as a student of the University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work”

Submitted to:

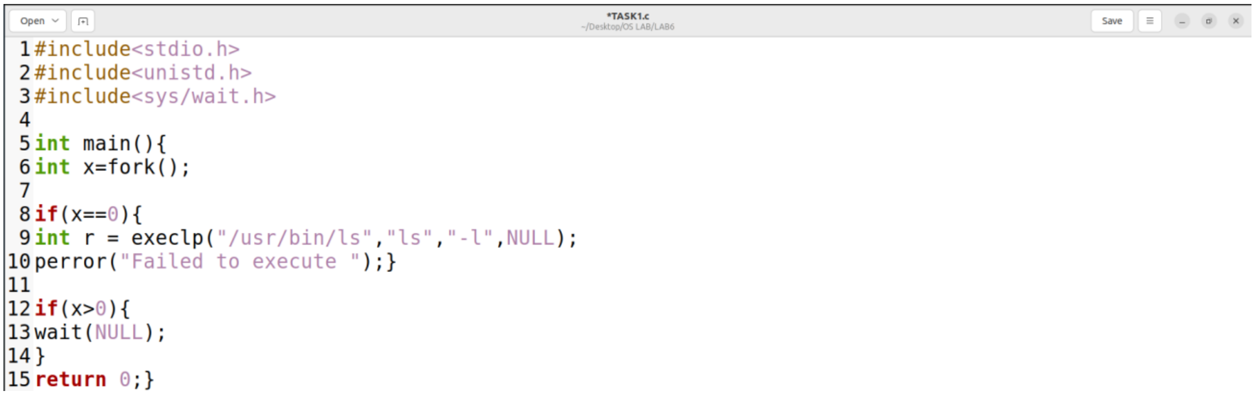
Engr. Madiha Sher

(18 May 2023)

Department of Computer systems engineering
University of Engineering and Technology,
Peshawar

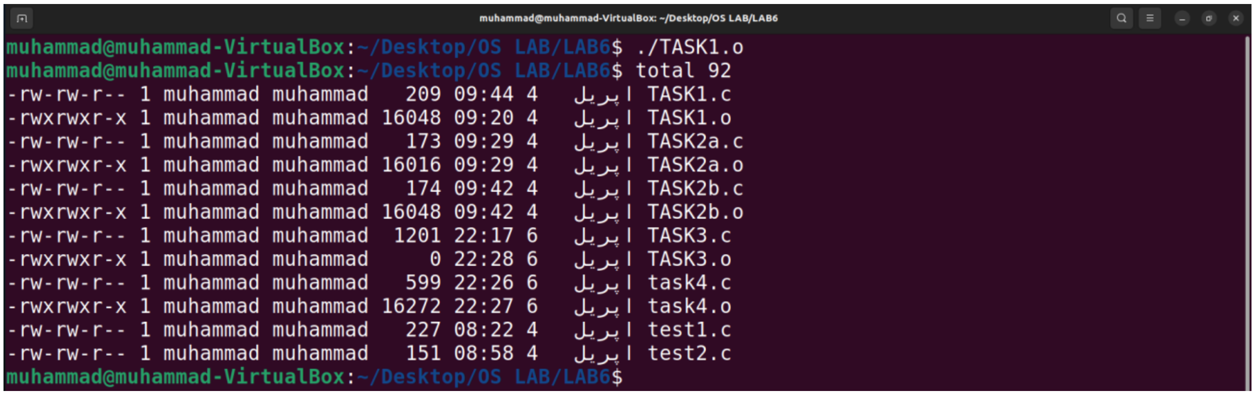
TASK 1

CODE:



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

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

if(x==0){
int r = execlp("/usr/bin/ls","ls","-l",NULL);
perror("Failed to execute ");}

if(x>0){
wait(NULL);
}
return 0;}
```

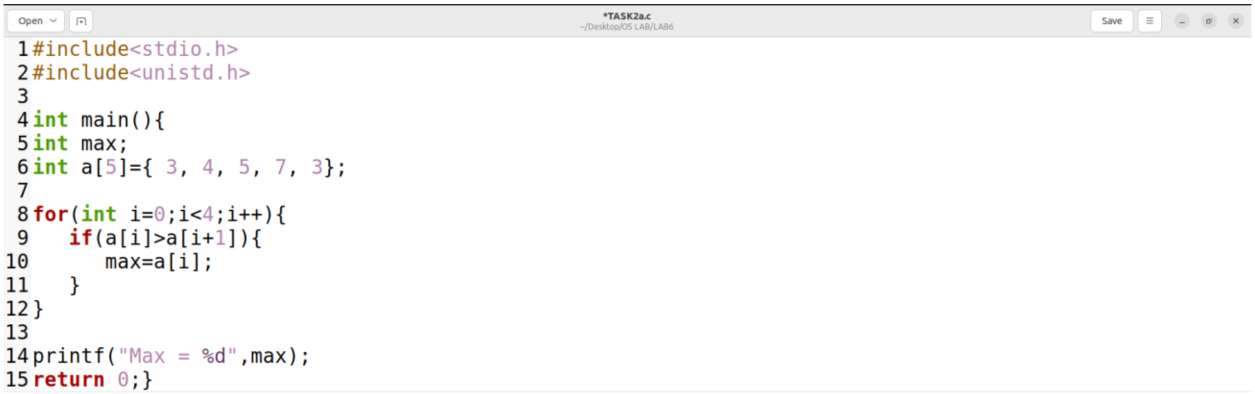
Output:



```
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ ./TASK1.o
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ total 92
-rw-rw-r-- 1 muhammad muhammad   209 09:44 4   اپریل TASK1.c
-rwxrwxr-x 1 muhammad muhammad 16048 09:20 4   اپریل TASK1.o
-rw-rw-r-- 1 muhammad muhammad   173 09:29 4   اپریل TASK2a.c
-rwxrwxr-x 1 muhammad muhammad 16016 09:29 4   اپریل TASK2a.o
-rw-rw-r-- 1 muhammad muhammad   174 09:42 4   اپریل TASK2b.c
-rwxrwxr-x 1 muhammad muhammad 16048 09:42 4   اپریل TASK2b.o
-rw-rw-r-- 1 muhammad muhammad  1201 22:17 6   اپریل TASK3.c
-rwxrwxr-x 1 muhammad muhammad     0 22:28 6   اپریل TASK3.o
-rw-rw-r-- 1 muhammad muhammad   599 22:26 6   اپریل task4.c
-rwxrwxr-x 1 muhammad muhammad 16272 22:27 6   اپریل task4.o
-rw-rw-r-- 1 muhammad muhammad   227 08:22 4   اپریل test1.c
-rw-rw-r-- 1 muhammad muhammad   151 08:58 4   اپریل test2.c
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$
```

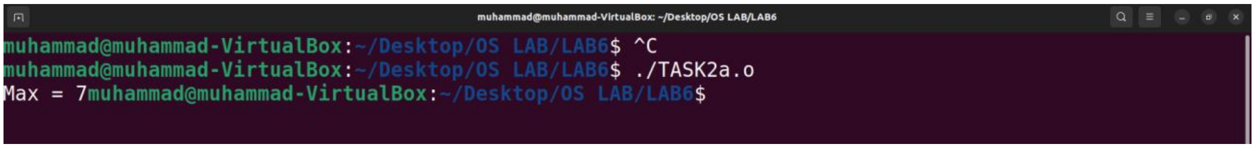
TASK 2

CODE No.1:



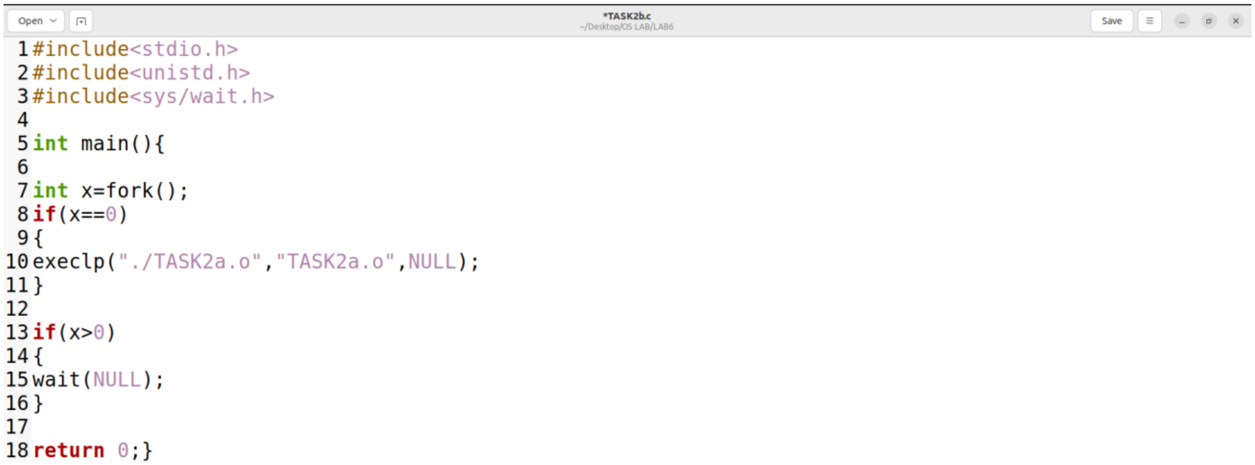
```c
#include<stdio.h>
#include<unistd.h>

int main(){
int max;
int a[5]={ 3, 4, 5, 7, 3};

for(int i=0;i<4;i++){
   if(a[i]>a[i+1]){
      max=a[i];
   }
}

printf("Max = %d",max);
return 0;}
```

Output:



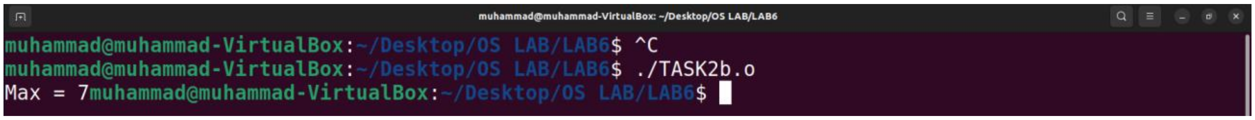
```
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ ^C
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ ./TASK2a.o
Max = 7muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$
```

CODE No.2:



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

int main(){

int x=fork();
if(x==0)
{
execlp("./TASK2a.o","TASK2a.o",NULL);
}

if(x>0)
{
wait(NULL);
}

return 0;}
```

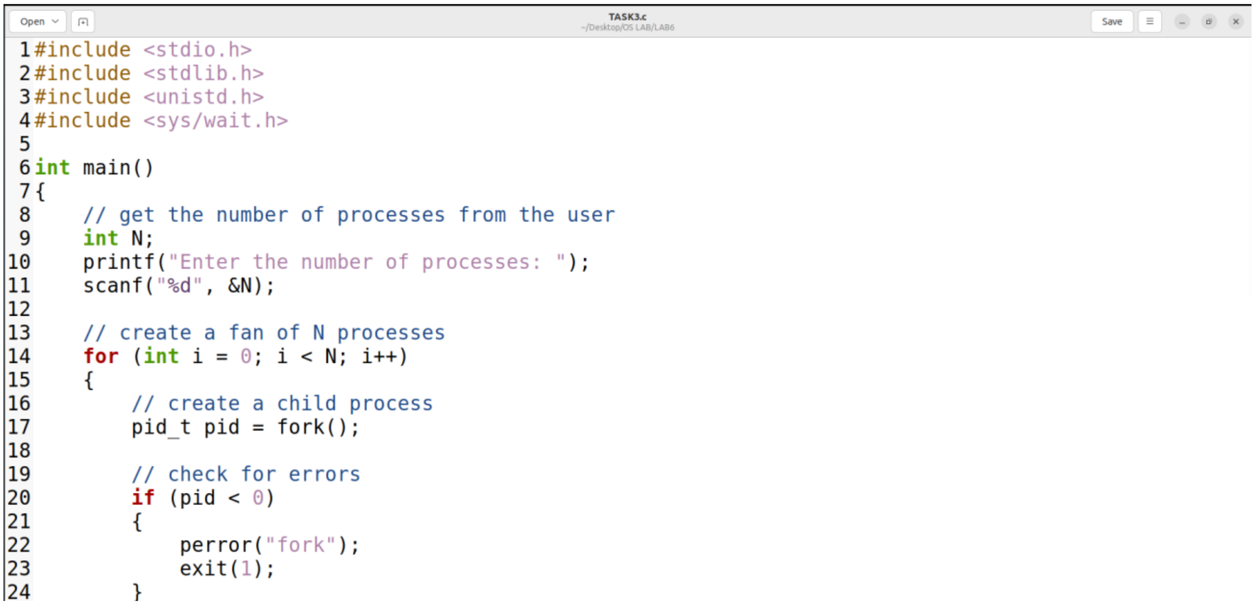
Output:



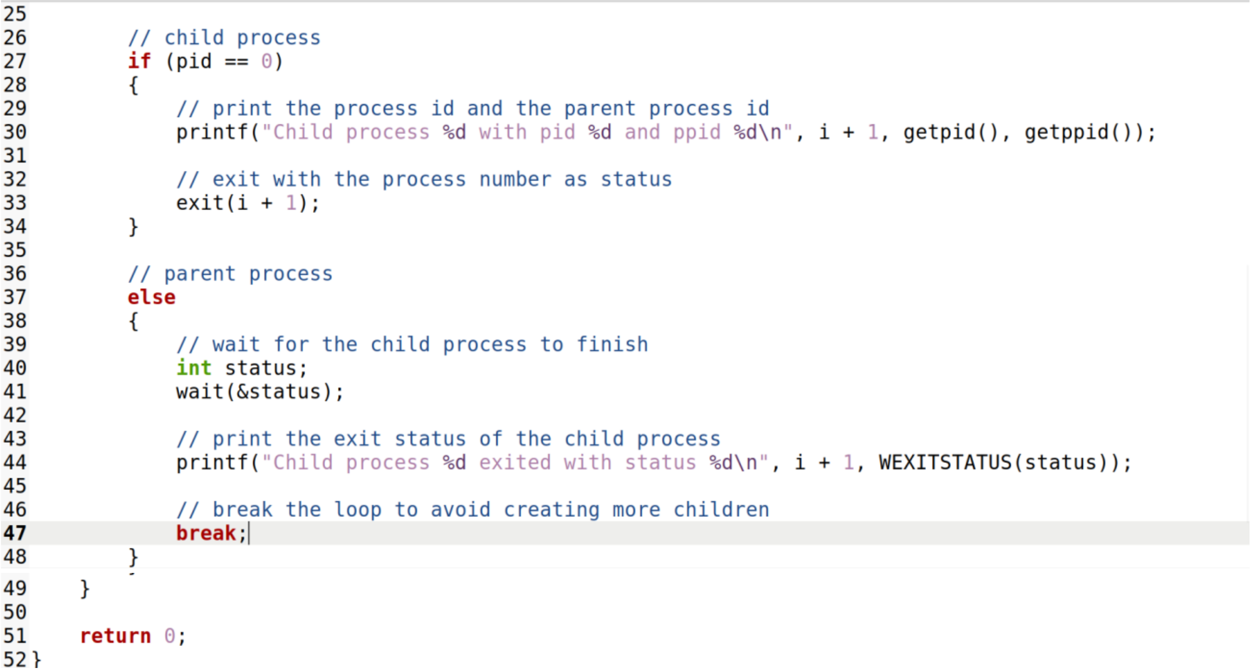
```
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ ^C
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ ./TASK2b.o
Max = 7muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ 
```

TASK 3

CODE:

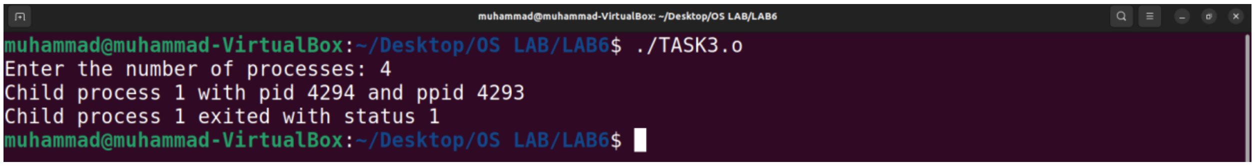


```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main()
{
    // get the number of processes from the user
    int N;
    printf("Enter the number of processes: ");
    scanf("%d", &N);

    // create a fan of N processes
    for (int i = 0; i < N; i++)
    {
        // create a child process
        pid_t pid = fork();

        // check for errors
        if (pid < 0)
        {
            perror("fork");
            exit(1);
        }
```



```c

        // child process
        if (pid == 0)
        {
            // print the process id and the parent process id
            printf("Child process %d with pid %d and ppid %d\n", i + 1, getpid(), getppid());

            // exit with the process number as status
            exit(i + 1);
        }

        // parent process
        else
        {
            // wait for the child process to finish
            int status;
            wait(&status);

            // print the exit status of the child process
            printf("Child process %d exited with status %d\n", i + 1, WEXITSTATUS(status));

            // break the loop to avoid creating more children
            break;
        }
    }

    return 0;
}
```

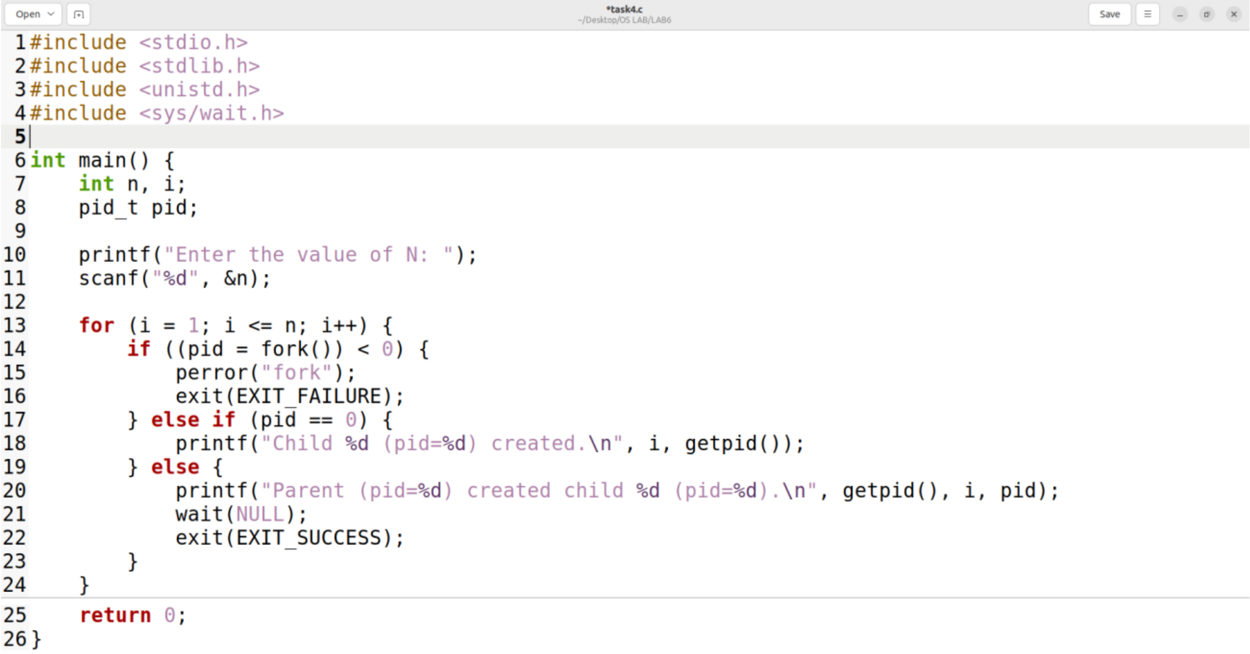
Output:



```
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ ./TASK3.o
Enter the number of processes: 4
Child process 1 with pid 4294 and ppid 4293
Child process 1 exited with status 1
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ 
```

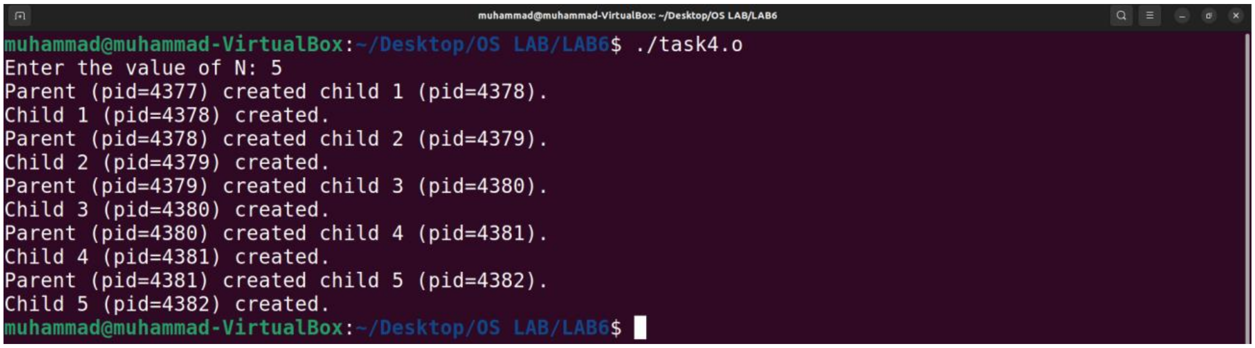
TASK 4

CODE:



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

int main() {
    int n, i;
    pid_t pid;

    printf("Enter the value of N: ");
    scanf("%d", &n);

    for (i = 1; i <= n; i++) {
        if ((pid = fork()) < 0) {
            perror("fork");
            exit(EXIT_FAILURE);
        } else if (pid == 0) {
            printf("Child %d (pid=%d) created.\n", i, getpid());
        } else {
            printf("Parent (pid=%d) created child %d (pid=%d).\n", getpid(), i, pid);
            wait(NULL);
            exit(EXIT_SUCCESS);
        }
    }
    return 0;
}
```

Output:



```
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ ./task4.o
Enter the value of N: 5
Parent (pid=4377) created child 1 (pid=4378).
Child 1 (pid=4378) created.
Parent (pid=4378) created child 2 (pid=4379).
Child 2 (pid=4379) created.
Parent (pid=4379) created child 3 (pid=4380).
Child 3 (pid=4380) created.
Parent (pid=4380) created child 4 (pid=4381).
Child 4 (pid=4381) created.
Parent (pid=4381) created child 5 (pid=4382).
Child 5 (pid=4382) created.
muhammad@muhammad-VirtualBox:~/Desktop/OS LAB/LAB6$ 
```