Bahria University

Karachi Campus



LAB EXPERIMENT NO.

**06**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Write what you have learned in few lines on each of the three programs that were using the ***fork()*** system call. |
| 2 | Write a C program that uses ***fork()*** system call to print a single line eight times without using ***for*** loop and repeated ***printf*** command. |
| 3 | Code the C program given below and explain what it does along with providing a snapshot of the output. Investigate and write about the usage of ***execlp()*** system call. |
| 4 | Write a program to find sum of even numbers in parent process and sum of odd numbers in child process. |

Submitted On:

20/04/2023

(Date: DD/MM/YY)

**Task 1**: Write what you have learned in few lines on each of the three programs that were using the ***fork()*** system call.

**1-**In Linux, process is created by duplicating parent process. This is called forking. You invoke the fork system call with fork() function.It is a system call that creates a new process under Linux operating system. It takes no argument. The purpose of fork() is to create a new process which becomes the child process to the caller. After the new child process is created, both processes will execute next instruction following the fork system call, therefore we have to distinguish the parent process from the child which can be done by evaluating the returned value of fork() function.If the returned value is negative, it means that the child process creation was unsuccessfulIf the returned value is zero, the child process is created with pid = 0.If the returned value is positive, the child process is created with the process with a process ID to the parent process (The returned process ID is of type pid\_t defined in sys.type.h).

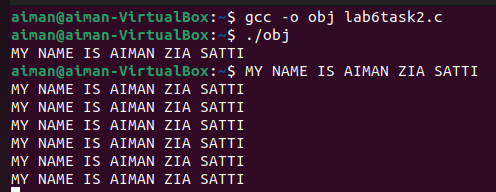
**2-**The progression with each ***fork( )*** that is added is 2 raised to the power of n, , where *n* is the number of calls to ***fork( )***. The program below will create four processes.

**3-**There are two system calls (functions) which can get the Process ID one is to get the process ID of the current process and another one to get the ID of its parent process. These are:

‘getpid()’ returns the PID of current process

‘getppid()’ returns the PID of parent’s process

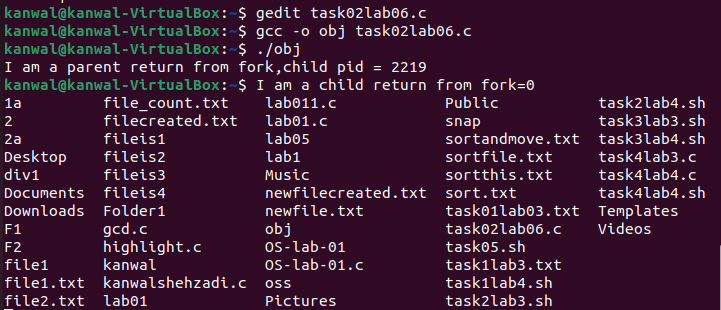
**Task 2**: Write a C program that uses ***fork()*** system call to print a single line eight times without using ***for*** loop and repeated ***printf*** command.

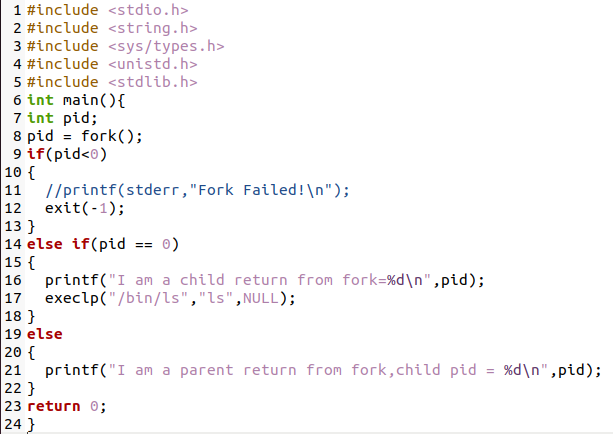


Graphical user interface, text, application

Description automatically generated

**Task 3**: Code the C program given below and explain what it does along with providing a snapshot of the output. Investigate and write about the usage of ***execlp()*** system call.





The execlp() family of commands can be used to execute an application from a process. The system call execlp() replaces the executing process by a new process image which executes the application specified as its parameter. Arguments can also be specified.In simple words execlp() is a function that takes single or multiple strings and a Null pointer as parameter. First string defines the path of an executable file that we want to be executed and then the next strings are the commands that we want to run.Execlp(“Path/Directory”, ”Command1”,...,NULL).The given program first runs the else statement and because of fork it runs second time and till that time it moves in child class and the pid will have 0 value in it. The else if statement becomes true so it reaches execlp(). Execlp() is moving to path ‘/bin/ls’ and executing the command ‘ls’ which is listing all the files.

**Task 4**: Write a program to find sum of even numbers in parent process and sum of odd numbers in child process.

