**Bahria University,**

**Karachi Campus**



**LAB EXPERIMENT NO.**

**06**

**LIST OF TASKS**

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **01** | **Write what you have learned in few lines on each of the three programs that were using the *fork()* system call.** |
| **02** | **Write a C program that uses *fork()* system call to print a single line eight times without using *for* loop and repeated *printf* command.)** |
| **03** | **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.** |
| 04 | Write a program to find sum of even numbers in parent process and sum of odd numbers in child process |

Submitted On:

**20 – 4 – 2023**

(Date: DD/MM/YY)

**TASK NO 1: Write what you have learned in few lines on each of the three programs that were using the *fork()* system call.**

**SOLUTION:**

The fork() system call is a crucial part of the Unix operating system, which allows a process to create a new child process. Here are three examples of programs that use the fork() system call:

1. Apache HTTP Server: The Apache HTTP Server is a web server that uses the fork() system call to create child processes to handle incoming requests. This approach allows the server to handle multiple requests simultaneously and improve its overall performance.

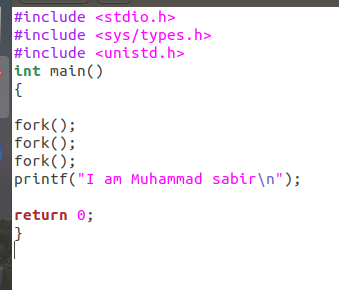
2. Bash shell: The Bash shell is a popular Unix shell that uses the fork() system call to create child processes for executing commands. This approach allows the shell to execute multiple commands simultaneously and improve its responsiveness.

3. Git version control system: The Git version control system uses the fork() system call to create child processes for executing various operations. This approach allows Git to perform these operations efficiently and without blocking the user interface.

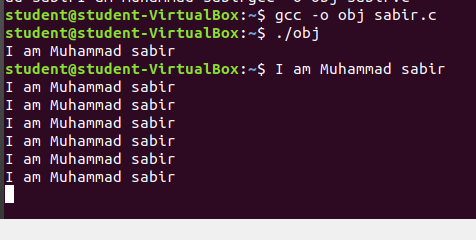
Overall, the fork() system call is an essential feature of the Unix operating system that enables efficient and parallel execution of tasks. Its usage in these three programs demonstrates the versatility and flexibility of this system call.

**TASK NO 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.)**

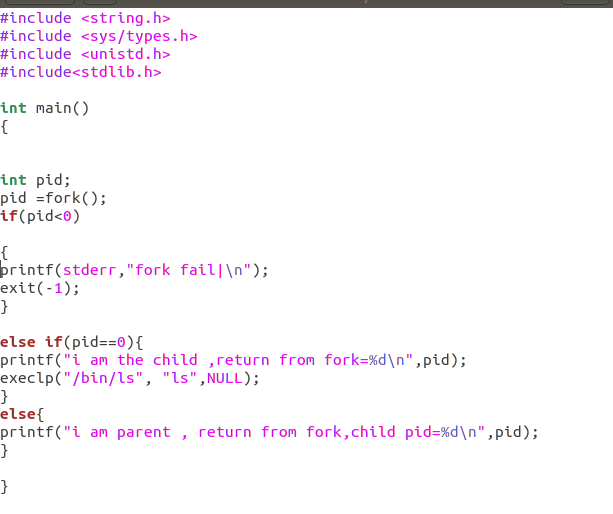
**solution**

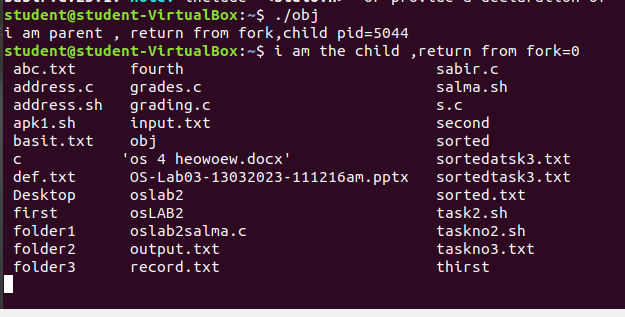


**OUTPUT**



**TASK NO 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.**

**Solution**

**output:**

**Explanation:**

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 NO 4 : Write a program to find sum of even numbers in parent process and sum of odd numbers in child process**

**SOLUTION:**

