**Operating System**

**Lab Report**

**Lab 07**



|  |  |
| --- | --- |
| Group Members Name & Reg #: | **Muhammad Haris Irfan**  **(FA18-BCE-090)** |
|  |  |
| Class | Operating System (**BCE-5B**) |
| Instructor’s Name | Sir Wasif |

**In Lab Tasks**

**Question 1: Activity 1:**

In this activity, you are required to perform tasks given below:

1. Print something and Check id of the parent process

2. Create a child process and print child process id in parent process

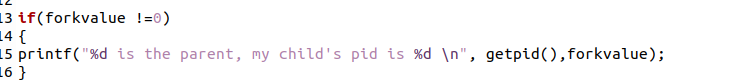
3. Create a child process and print child process id in child process

**Solution:  
1)** The Terminal Output is given below,





**2)** The Terminal Output is given below,





**3)** The Terminal Output is given below,

Text

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 2: Activity 2:**

**1. Create a process and make it an orphan.**

Hint: To, illustrate this insert a sleep statement into the child’s code. This ensured that the parent process terminated before its child.

**Steps to create an orphan process:**

1. print something and get its pid and ppid

2. create a child process

3. Now as a parent process print parent id and id of child process

4. Make child sleep for 5 seconds

5. Now while child is sleeping parent will terminate. Print parent id of child to make sure it is orphaned (PPID has been changed)

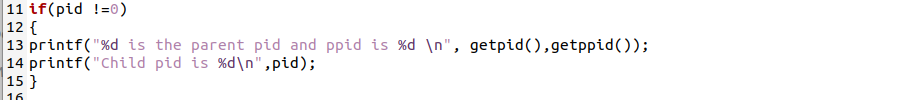
Solution:

1) The Terminal Output is given below,



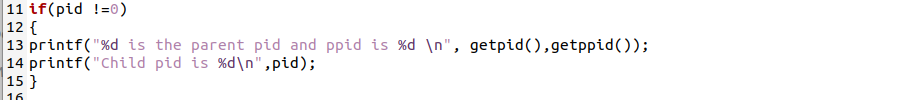


2) The Terminal Output is given below,





3) The Terminal Output is given below,





4) The Terminal Output is given below,



Text

Description automatically generated5) The Terminal Output is given below,Graphical user interface, text, application, email

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 3: Activity 3:**

**Create a process and make it a Zombie.**

1. Execute fork to create a child

2. In parent process (using if statement) Create an infinite loop so that it never terminates and never executes wait ()

3. Make parent sleep for 100 sec

4. Terminate child process exit using exit.

Now this child is a zombie because no parent is waiting for him

To view status of processes, use the following commands on command line:

1. Execute the c code in background using ./abc &

2. View process status using ‘ps -lf’

3. You will see zombie process with state ‘Z’ in STAT column

4. Get parent id and kill parent process using “kill (parent id)” command

5. again execute ‘ps -lf’

6. Note that Zombie is gone now

Solution:

1) The Terminal Output is given below,



2) The Terminal Output is given below,



3) The Terminal Output is given below,



4) The Terminal Output is given below,



1) The Terminal Output is given below,

Text

Description automatically generated

2) The Terminal Output is given below,

Text

Description automatically generated

3) The Terminal Output is given below,

Text

Description automatically generated

4) The Terminal Output is given below,

Text

Description automatically generated

5) The Terminal Output is given below,Text

Description automatically generated

6) The Terminal Output is given below,

Text

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 4: Activity 4:**

Write a C/C++ program in which a parent process creates a child process using a fork() system call. The child process takes your age as input and parent process prints the age.

Solution:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question 5: Activity 5:**

1. Write a C/C++ program that asks user to enter his name and his university name. Within the same program, execute another program that asks the user to enter his degree name and department name.

2. Hint: write 2 separate programs and execute using execv()

Solution:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

\_\_\_\_\_\_THE END\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_