National University of Computer and Emerging Sciences

Laboratory Manual # 02
Operating Systems Lab

Lab Instructor	Fatima Ali
Section	BCS-4A, BCS-4B, BCS-6B
Semester	Spring – 26

Instructions:

- Submit a word file containing screenshots of your outputs with question number.
- In case of any explanation, you can add a multiline comment or add details under the screenshot of the output.
- Submit your code files with question number and roll number.

Objectives:

- Practicing Fork & Exec system call (basics)

1. Exercise: [05]

Write a program that creates a child process using fork().

- Create a global counter
- The child process should increment and print a counter in a loop.
- The parent process should also increment and print the same counter.
- Explain why their values are different.

2. Exercise: [05]

Write a program that calls fork() 3 times (consecutively).

- How many processes are created in total? Draw a process tree.
- Comment the previous code and update the program to create only three child processes. Write print statements in each child. Also draw a process tree.
- Make sure to use wait() properly.

3. Exercise: [05]

Write a program where:

- The child prints its id and exits immediately.
- The parent sleeps for 20 or less seconds without calling wait().
- Use the ps command to observe the zombie process.
- Take and submit Screen shot
- Fix the issue using wait().

To observe zombie process you can use

```
ps -aux | grep defunct
ps -l | grep Z
```

4. Exercise: [05]

Write a program where:

- A parent forks a child.
- The child uses execlp() to run the cat file.txt command. [Create a txt file in same folder and place some random text in it]
- The parent waits for the child to finish.

5. Exercise: [05]

Write a program that creates three child processes.

- Each child runs a different command using exec() (i.e. pwd, date, rm).
- Parent waits for all children.