**UIT2512---Operating Systems Practices Lab**

# 2) Creating a Zombie Process using Python

# Name: Vasundhara.B

# Roll no: 3122 21 5002 119

# Here, you have to write a program in Python to create One Parent - One Child, where the Parent process terminates before the child completes its execution. The child returns the exit status to the parent, but the parent is not existing and causing the entry for the child process even after it becomes as the dead process.

# CODE:

import os

import time

def fibonacci(n):

    if n <= 0:

        return []

    elif n == 1:

        return [0]

    elif n == 2:

        return [0, 1]

    else:

        fib\_series = [0, 1]

        for i in range(2, n):

            next\_term = fib\_series[-1] + fib\_series[-2]

            fib\_series.append(next\_term)

        return fib\_series

# Create a child process

cpid = os.fork()

if cpid == 0:

    # Print child's PID and parent's PID

    print(f"\nChild: I have the PID: {os.getpid()}, my parent's PID: {os.getppid()}")

    fib\_series = fibonacci(10)

    print(f"\nChild: Generated Fibonacci series: {fib\_series}")

# Code for parent

else:

    print(f"\nParent: I have the PID: {os.getpid()}, my child's PID: {cpid}")

    print("\nParent: I am going to sleep!!")

    # Make the parent process sleep for 4 seconds

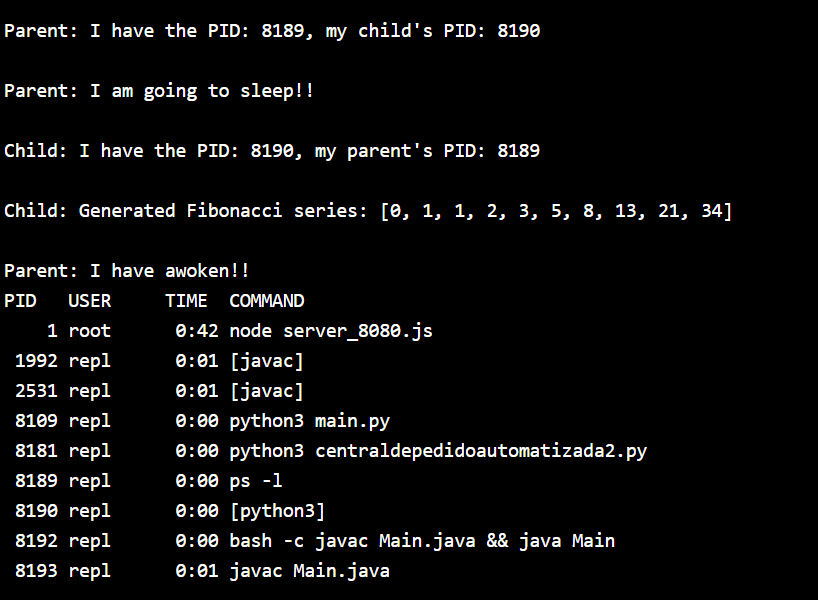
    time.sleep(4)

    print("\nParent: I have awoken!!")

    # Execute the 'ps' command with the '-l' flag

    os.execlp("ps", "ps", "-l")

**OUTPUT:**

****

**"ps -l" command:**

This command lists processes along with detailed information