

**Usman Institute of Technology**

**Department of Computer Science Fall 2022**

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

Course Instructor: Ma’am Shabina Mushtaq

Date: 17-Nov-2022

# Lab Tasks:

1. Using a Linux system, write a program that forks a child process that ultimately becomes a zombie process. This zombie process must remain in the system for at least 10 seconds.

#!/bin/python3

import os,time

id=os.fork()

if id == 0:

    print("The child is running")

    time.sleep(10)

else:

    print("The parent is running")

    os.wait()

Output:



1. Write a program that creates a child process which further creates its two child processes. Store the process id of each process in an array called Created Processes. Also display the process id of the terminated child to understand the hierarchy of termination of each child process.

#!/bin/python3

import os,time

created\_processes = []

parent = os.fork()

if parent == 0:

    child\_1 = os.fork()

    if child\_1 == 0:

        print("Child is running with pid ", os.getpid())

    else:

        status = os.wait()

        created\_processes.append(status[0])

        print("Parent is running with pid ", os.getpid())

        child\_2 = os.fork()

        if child\_2 == 0:

            print("Child is running with pid ", os.getpid())

        else:

            status = os.wait()

            created\_processes.append(status[0])

else:

    status = os.wait()

    created\_processes.append(status[0])

    print("Parent is running with pid ", os.getpid())

    child\_3 = os.fork()

    if child\_3 == 0:

        print("Child is running with pid ", os.getpid())

    else:

        status = os.wait()

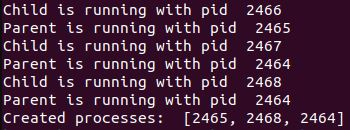
        created\_processes.append(status[0])

        print("Parent is running with pid ", os.getpid())

        created\_processes.append(os.getpid())

        print("Created processes: ", created\_processes)

Output:



1. Write a program in which a parent process will initialize an array, and child process will sort this array. Use wait() and sleep() methods to achieve the synchronization such that parent process should run first.

#!/bin/python3

import os, time

arr = [1, 3, 2, 5, 4]

parent = os.fork()

if parent == 0:

    print("Child is running")

    print("Sorting...")

    arr.sort()

    print("Sorted array: ", arr)

else:

    print("Parent is running")

    print("Array initialized")

    os.wait()

Output:

