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

# 1) Creating an Orphan 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. Hence, the child becomes as an orphan process, at it is taken care a different process. You have to identify the new parent process of that child process in your code and understand new process.

# CODE:

import os,time

p = os.fork()

if(p==0):

   print("CHILD")

   factorial=1

   num=int(input("Enter a number:" ))

   for i in range(1,num + 1):

       factorial = factorial\*i

   print("The factorial of",num,"is",factorial)

   #time.sleep(3)

   print("Child PID {}".format(os.getpid()))

   print("Parent PID {}\n".format(os.getppid()))

else:

   print("PARENT")

   print("Parent PID {} ".format(os.getpid()))

   print("Child PID {}\n".format(p))

# OUTPUT:

# A screenshot of a computer Description automatically generated

# pstree -s -p (ppid) command: The pstree command is used to display the processes in a tree-like structure, showing their parent-child relationships. By specifying the -s and -p options, one can see the process names and their Process IDs (PIDs) in the tree.

# A screenshot of a computer Description automatically generated

# :

# :

# A screenshot of a computer Description automatically generated

# ps -e command: The ps -e command is used to list all currently running processes on a Unix-based operating system. It displays a list of processes, along with their associated information.

# While execution:

# A screenshot of a computer Description automatically generated

# A screenshot of a computer Description automatically generated

# Process 10365 exists.

# After execution:

# A screenshot of a computer Description automatically generated