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.
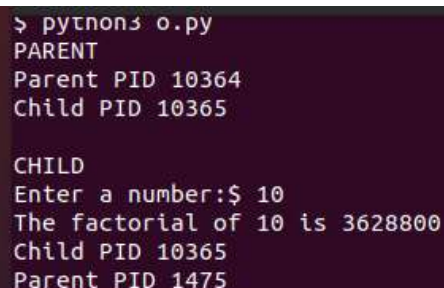
CODE:

```python
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:

```
$ python3 o.py
PARENT
Parent PID 10364
Child PID 10365

CHILD
Enter a number:$ 10
The factorial of 10 is 3628800
Child PID 10365
Parent PID 1475
```

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.

:

:

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:

```
10381 pts/1    00:00:00 ps
$ ps -e
   PID TTY         TIME CMD
     1 ?       00:00:01 systemd
     2 ?       00:00:00 kthreadd
     3 ?       00:00:00 rcu_gp
     4 ?       00:00:00 rcu_par_gp
     5 ?       00:00:00 slub_flushwq
     6 ?       00:00:00 netns
    10 ?       00:00:00 mm_percpu_wq
    11 ?       00:00:00 rcu_tasks_rude
```

```
  6464 ?       00:00:00 kworker/0:0H-kblockd
  6473 ?       00:00:00 gvfsd-dnssd
  7680 ?       00:00:00 kworker/2:0H-kblockd
  8235 ?       00:00:02 kworker/3:0-events
  8236 ?       00:00:02 kworker/0:3-events
  8718 ?       00:00:00 kworker/3:2H
  8818 ?       00:00:00 kworker/u8:2-events_unbound
  8838 ?       00:00:03 kworker/2:2-events
  8945 ?       00:00:04 kworker/1:2-events
  9705 ?       00:00:02 kworker/0:1-events
  9767 ?       00:00:02 kworker/3:2-events
 10134 ?       00:01:02 Isolated Web Co
 10137 ?       00:00:00 Isolated Web Co
 10179 ?       00:00:00 Web Content
 10228 ?       00:00:00 kworker/1:0H-radeon-crtc
 10264 ?       00:00:00 kworker/1:1-events
 10268 ?       00:00:00 kworker/2:1-events
 10277 ?       00:00:00 Web Content
 10305 ?       00:00:00 Web Content
 10360 ?       00:00:00 kworker/u8:0
 10365 pts/0   00:00:00 python3
 10377 pts/1   00:00:00 sh
 10381 pts/1   00:00:00 ps
```

➔ **Process 10365 exists.**

After execution:

```
 10179 ?       00:00:00 Web Content
 10228 ?       00:00:00 kworker/1:0H-radeon-crtc
 10264 ?       00:00:00 kworker/1:1-events
 10268 ?       00:00:00 kworker/2:1-events
 10277 ?       00:00:00 Web Content
 10305 ?       00:00:00 Web Content
 10360 ?       00:00:00 kworker/u8:0
 10377 pts/1   00:00:00 sh
 10387 pts/1   00:00:00 ps
```