Course COMP-8567
Assignment 02
Winter 2024
Due Date: Mar/03/2024
50 Marks

CRITICAL POINTS TO NOTE (before you start this assignment)

- MOST IMPORTANT: While creating process trees (through fork()) and to keep the processes alive in order to test a2prc, DO NOT USE INFINITE LOOPS , BUT USE REASONABLE SLEEP (3-4 MINUTES)
- Since using infinite loops leads to fork bombs and is a very serious matter, unfortunately, students who use infinite loops for this assignment will be marked a ZERO
  - Please take it seriously since I do not want any comments/concerns from the system administrator. The system administrator can easily share the usernames associated with fork() bombs.
- Before you logout for the day, make sure you enter the command:

  $killall -u *username* (very important since this will take care of any fork() bomb related issues!)

Write a C program a2prc.c that searches for a process in the process tree (rooted at a specified process) and prints the requested information based on the input parameters (it might be a good idea to look at the sample runs included later)

Synopsis :

a2prc [*process_id]* [*root_process*] [*OPTION*]

- Lists the PID and PPID of *process_id* if *process_id* belongs to the process tree rooted at *root_process* else does not print anything

  root process is the PID of a process that is a descendant of any Bash under the same user //Note: multiple BASH terminals under the same user can be kept open.

process_id is the PID of a process that is a descendant of any Bash under the same user

Note: In any of the following options, if process_id does not belong to the process tree rooted at root_process , you need to print *" Does not belong to the process tree"*

OPTION (Note: Print suitable messages for both success and failure scenarios)

-rp process_id is killed **if** it belongs to the process tree rooted at root_process

-pr the *root_process* is killed (if it is valid)

-xn lists the PIDs of all the **non-direct** descendants of *process_id*

-xd lists the PIDs of all the **immediate** descendants of *process_id*

-xs lists the PIDs of all the **sibling processes** of *process_id*

-xt *process_id* is paused with **SIGSTOP**

-xc **SIGCONT** is sent to all processes that have been paused earlier

- xz Lists the PIDs of all **descendents of** *process_id* **that are defunct**

- xg lists the PIDs of all the **grandchildren** of *process_id*

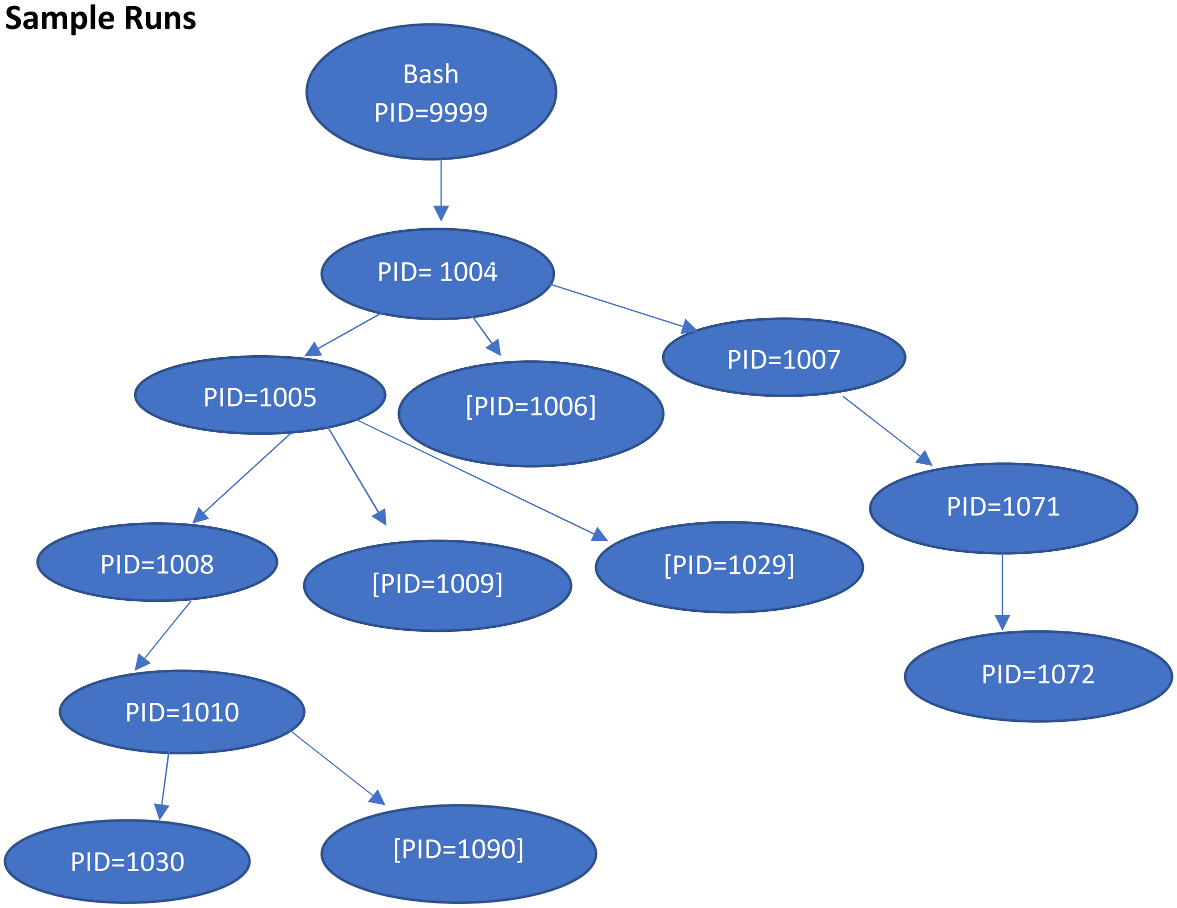
- zs prints the status of *process_id* (Defunct/ Not Defunct)

SAMPLE RUN:

THE FOLLOWING SAMPLE RUN (PROCESS TREE) IS JUST AN EXAMPLE, YOUR PROGRAM SHOULD WORK ON ANY PROCESS TREE CREATED FROM ANY BASH (UNDER THE SAME USER).

- **YOU MIGHT HAVE CREATED 8 PROCESSES THAT ARE CURRENTLY RUNNING FROM BASH1, 16 PROCESSES THAT ARE CURRENTLY RUNNING FROM BASH 2, 8 PROCESSES CURRENTLY RUNNING FROM BASH 3 …SO ON…You have to RUN a2prc from a new BASH and be able to perform the actions listed above.**

Sample Runs



Note: In the above example, [PID=1006], [PID=1009], [PID=1029] and [PID=1090] are defunct (zombie) processes at the time of execution of the following programs

```
$ a2prc 1009 1004
1009 1005

$ a2prc 1072 1004
1072 1071


$ a2prc 1005 1007
Does not belong to the process tree

$ a2prc 1020 1005
Does not belong to the process tree




$ a2prc 1005 1004  -xn
1010
1030
1090

$ a2prc 1010 1008  -xn
No non-direct descendants

$ a2prc 1005 1004  -xd
1008
1009
1029

$ a2prc 1030 1008  -xd
No direct descendants
```

```
$ a2prc 1005 1004 -xz
1009
1029
1090

$ a2prc 1009 1005 -xz
No descendant zombie process/es



$ a2prc 1030 1004 -xs
1090

$ a2prc 1071 1005 -xs
Does not belong to the process tree

$ a2prc 1072 1004 -xs
No sibling/s

$ a2prc 1010 1008 -zs
Not defunct

$ a2prc 1090 1008 -zs
Defunct

$ a2prc 1008 1005 -xg
1030
1090

$ a2prc 1010 1008 -xg
No grandchildren
```

Comments and explanation of the program

-You are required to include adequate and appropriate comments to explain the working of the program.
-Please see the assignment rubrics for more information

Submission:

Submission Instructions (Note: Plagiarism Detection Tool: MOSS)

You need to submit the following:
1. a2prc_firstname_lastname_SID.c
3. Zoom/Google Drive recording link explaining the following (not more than 15 minutes)

- Overall working of the code and various modules (around 8-9 minutes)
- Execution of the code under various inputs/conditions as per the requirements of the assignment (around 6-7 minutes)
- Other form of links/MP4 files will NOT be acceptable.
- Include the link in the COMMENTS section.