

Team 17
Introduction to Computer Systems
Coding Project 1 (Problem 2 - 18 points)
April 5, 2022

Name	Email
Kevin Zhang	kyz2@scarletmail.rutgers.edu
Justine Catli	jlc562@scarletmail.rutgers.edu
Sumant Pottepalem	ssp213@scarletmail.rutgers.edu
Harsh Patel	hp440@scarletmail.rutgers.edu
Jawad Abdussalam	jja153@scarletmail.rutgers.edu
Felix Shames	fs410@scarletmail.rutgers.edu
Brandon Yuen	by161@scarletmail.rutgers.edu
Taqiya Ehsan	te137@scarletmail.rutgers.edu

All had equal contributions.
Project content begins on the next page.

Code is provided in a separate C file.

Input

There are no artificial inputs in this code, however, these are the PIDs of the given processes.

A = 321992

B = 321994

C = 321993

D = 321995

Output

"I am Process A (321992), the root of the tree"

"I am Process C (321993), leaf of process A (321992)"

"I am Process B (321994), leaf of process A (321992)"

"I am Process D (321995), leaf of process B (321994)"

"Process A (321992) has terminated with return code 0"

"Process D (321995), child of Process B (1), has terminated with return code 10"

```
te137@engsoft:~$ ./Proj1Prob2
I'm process A (321992), the root of the tree
I'm process C (321993), leaf of process A (321992)
I'm process B (321994), leaf of process A (321992)
I'm process D (321995), leaf of process B (321994)
Proj1Prob2(321992)─┬─Proj1Prob2(321993)
                  └─Proj1Prob2(321994)─Proj1Prob2(321995)─sh(321996)─pstree(+
Process A (321992) has terminated with return code 0
Process D (321995), child of Process B (1) has terminated with return code 10
```

Design Decisions

- There were many factors that went into the structure of our code. The main goal of our code was to go step by step in order to find which node is the correct one. We start off by first finding the root of the tree which is A, which thereafter forks into two nodes, b and c. We can see that C no longer forks so that was next in line for us to code and recognize. The leaf of node A that no longer forks any more is C. Then we continue by identifying node B since it is the only remaining leaf that continues to fork into D which is the last node. This one after another method to our design was implemented using a lot of if-else statements so that we were able to check the nodes one at a time.

What We Have Learned

- We had quite a bit of trial and error while attempting this problem. What we learned from this problem specifically is how forking impacts the parents and the children. We also learned how to code advanced projects in C since most of us have more experience in Java programming. All in all, this was a great learning experience for all of us, especially with how to do a project with multiple group members and how to coordinate responsibilities etc.

Questions

1. If root process A is terminated prematurely, then the termination message for Process A is printed and the code will hang.
2. If we display the process tree with root `pstree(getpid())` instead of `pstree(pid())`, then we will display the PID of the calling process instead of the PID of the child process from the parent process. The other processes that appear in the tree will be the calling PID and the return PID of the calling PID.
3. The maximum random tree one can generate is of degree 4. This is because the process tree has 2 “branches” ($a \rightarrow b \rightarrow d$ and $a \rightarrow c$) and the longer branch has three nodes. If each process were able to fork successfully, then there would be a tree of height 4 with one more child at the end of each branch.