

# CSC 369 Worksheet 2 Solution

August 16, 2020

## 1 Homework (Simulation)

1. I need to create process trees at each step when the command `./fork.py -s 10` is run.

1) Action: a forks b



2) Action: a forks c



3) Action: c EXITS



4) Action: a forks d



5) Action: a forks e



### Notes

- **fork()**
  - Is used to create a new process
  - **Creator** → parent process
  - **Newly Created** → child process
  - Child process is nearly identical to parent process
- **exec()**
  - Allows a child to break free from its similarity to its parent and execute an entirely new program.
- **wait()**
  - Is used to let parent code delay its execution until the child finishes executing.
  - Makes the output deterministic

2. I need to write what the resulting final process trees will look like as the fork-percentage changes. Here I ran command (`./fork.py -s 10 -a 10 -f 0.1` and `./fork.py -s 10 -a 10 -f 0.9`)

Notes

- `./fork.py -s 10 -a 10 -f 0.1`



- `./fork.py -s 10 -a 10 -f 0.9`



Based on the diagram above, I can deduce that the lower the fork percentage, the more likely that `exit()` is executed by the childmost process, and the final tree will either have a single node or none.

On the other hand, the higher the fork-percentage is, the more likely that `fork()` is executed by the childmost process, and the final tree will have nodes that are deeply nested.

3. I need to fill out blank entries created by the command `(./fork.py -t)`



4. I need to write what happens when a child exits; what happens to its children in the process tree.

When a child exists, all of its children will also exit.

I am not sure what happens when `-R` flag is used.

Correct Solution

I need to write what happens when a child exits; what happens to its children in the process tree.

When a child exists, its parentmost child, along with its children, will be attached to the parentmost node



When `-R` flag is used (i.e. `./fork.py -A a+b,b+c,c+d,c+e,c- -R`) and a child exists, its parentmost child, along with its children will be attached to the parent node of the child that exits



- I need to write down the final tree by looking at the series of actions generated (here, the command `./fork.py -F` is used).

Process Tree:

a

Action: a forks b

Action: b EXITS

Action: a forks c

Action: c forks d

Action: c forks e

Final Process Tree?

## 2 Homework (Code)

1.