Winston Shine

Operating Systems

Homework 1

4/9/2024

Chapter 4 Simulation

6. in the case that a child forks a new process it's easy to tell the order of actions, since child must have been forked first, then the nested child. but when 1 process makes multiple forks i don't think we can tell which of those forks occurred first

```
Final Process Tree:
                                a - b - c - f
                                |   |_ e
                                |_ d
```

b must have been forked before c, but we don't really know if b was forked before d.

Questions process-run.py

1. list 2 processes, each with 5 instructions and 100% chance of using the CPU (instead of issuing IO)
2. Runnning. Ready. Blocked.

   ```
   enum proc_state { UNUSED, EMBRYO, SLEEPING,
   RUNNABLE, RUNNING, ZOMBIE };
   ```

3. stdin, stdout, stderr
4. set of details about a specific process. stored in a process list that tracks all processes on the system

fork.py

1. specifies the seed used for random number generation
2. -F only shows the final state instead of the state after each action. -R has to do with how orphaned processes are handled. by default all orphaned processes become children of PID 1 (the inital process). with -R the result is different, thought i'm not entirely clear how it was decided which process to reparent to.