

Week 7 – Forking Cool!

CSE 503 – Summer 2024

Richard Douglas

Assignment Description

Project Details:

Please do the following problems:

1. Construct an array of 1500 random integers within range [0, 120] An input file input.txt is provided. Each line of input.txt is a query integer that you need to check how many of that number is in your random integer array. For each query integer, fork a new child process to do the counting. The output is for each input query, output the count and child process id. For example:

```
$> query: 17 count: 5 pid: 13342
```

```
$> query: 29 count: 3 pid: 13357
```

Logic Employed

Overview

Gosh... Processes are interesting. It's kind of tricky working with them. I actually had some fun and also made a for bomb... It seems like the simplest malware ever but still crashes my machine.

When approaching this problem, I first tried actually writing functions to do the work of this code – one to create the array (I used a vector first... but then implemented a standard C style array), one to print output, one to count the occurrences of the queried integer.... but the counting function would only return the correct count for the first query, then it would return a large number for each subsequent query... I'm not entirely sure why that happened, but I suspect it had to do with memory addressing.... and I didn't exactly look into it. Maybe you'll have some insights in the feedback.

At any rate, the same logic worked when I placed it all inside the main scope.

I think one of the trickiest parts of this assignment was iterating through all of the queries and creating a new child process for the query, and only one process for each query. I simply kept track of the parent query and if it matched the current process, a return statement was executed in that process.

The one other thing I struggled with was dealing with my username and path being displayed once more because a child process was run. It's a little funky but the program does what the assignment asked...

Screenshots

The following is the output run in my terminal.

```
richard@richard-IdeaPad:~/Desktop/uofl/2024_summer/cse503/homework/w7$ ./fork.exe
query: 9          count:14          PID:23463
richard@richard-IdeaPad:~/Desktop/uofl/2024_summer/cse503/homework/w7$ query: 27 count:21          PID:23464
query: 44          count:33          PID:23465
query: 103         count:40          PID:23466
query: 82          count:47          PID:23467
query: 115         count:53          PID:23468
query: 90          count:67          PID:23469
query: 117         count:78          PID:23470
query: 10          count:93          PID:23471
query: 31          count:100         PID:23472
query: 68          count:113         PID:23473
query: 2           count:123         PID:23474
query: 60          count:130         PID:23475
query: 59          count:147         PID:23476
query: 71          count:153         PID:23477
query: 69          count:167         PID:23478
query: 53          count:176         PID:23479
query: 88          count:184         PID:23480
query: 57          count:199         PID:23481
query: 35          count:212         PID:23482
query: 81          count:219         PID:23483
query: 74          count:234         PID:23484
query: 11          count:244         PID:23485
richard@richard-IdeaPad:~/Desktop/uofl/2024_summer/cse503/homework/w7$
```

Conclusions

This exercise was tricky. Operating systems is a challenging topic... I'm going to be reading a lot about this. It's as exciting and interesting as it is challenging.