

COP 6611 Programming Homework 1

Due: Jan 23, 2017 at 9:00a.m.

Your first programming assignment is to do problem 3.21 in the book in C with the following changes. You will fork two processes to print their respective sequence for the Collatz conjecture. The first will produce the the sequence which is indicated by the number on the command line and the second process the sequence from the command line number plus 4. Please print the child (1 or 2) with each number output and be sure the forked processes can run concurrently. Example output is:

```
[lohall@netcluster cop6611]$ collatz 11
```

From Child 2 init n=15, From Child 2 n=46, From Child 2 n=23, From Child 2 n=70, From Child 2 n=35, From Child 2 n=106, From Child 2 n=53, From Child 2 n=160, From Child 2 n=80, From Child 2 n=40, From Child 2 n=20, From Child 2 n=10, From Child 2 n=5, From Child 2 n=16, From Child 2 n=8, From Child 2 n=4, From Child 2 n=2, From Child 2 n=1,

One done!

From Child 1 init n=11, From Child 1 n=34, From Child 1 n=17, From Child 1 n=52, From Child 1 n=26, From Child 1 n=13, From Child 1 n=40, From Child 1 n=20, From Child 1 n=10, From Child 1 n=5, From Child 1 n=16, From Child 1 n=8, From Child 1 n=4, From Child 1 n=2, From Child 1 n=1,

Children Complete

The number entered on the **command line** must be greater than zero and less than 40.

Please put the function code in your file.

You will need to use `stdlib.h` if you want to use `atoi` to translate a character string into an integer. Use `sprintf` to put values into strings. You will need to do `wait` twice so that the main program finishes after the children (no cascading termination). You will need to use `argc` and `argv` to get command line arguments. Also, observe whether the processes always finish in the order in which they are forked.

Be extremely careful that a child process does not itself fork a process or you can fill the process table and lock up the machine. Testing of this work **must** only be done on `netcluster.cse.usf.edu`

If you lock up another machine trying this assignment out, it is a 0 for this assignment!

To hand in this program must be logged in to `netcluster.cse.usf.edu` and then type:
`/usr/local/os/turn_in1`

Then follow the prompts. Make sure your name is in the code file in comments!

If you are not familiar with Linux here is some useful information. `pwd` tells you the directory you are in, `cd` - changes directories, `mkdir` creates a new directory, `emacs` and `nano` (and `vi`) are available editors with `nano` easiest to use. The compiler is `gcc`.