CS5323 – Operating Systems II
Programming Assignment 0
Due: February 6, 2023, 11:59 p.m.
Submission: via Canvas

Your first assignment is to solve the Collatz Conjecture using forks with some changes.

What is Collatz Conjecture?

The Collatz conjecture concerns what happens when we take any positive integer n and apply the following algorithm:

$$n = \begin{cases} {}^{n}/_{2}, & if\ n\ is\ even \\ 3\ x\ n\ +\ 1, & if\ n\ is\ odd \end{cases}$$

The conjecture states that when this algorithm is continually applied, all positive integers will eventually reach 1. For example, if n = 35, the sequence is

35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Write a C program using the fork() system call that generates this sequence in the child process. The starting number will be provided from the command line. For example, if 8 is passed as a parameter on the command line, the child process will output 8, 4, 2, 1. Because the parent and child processes have their own copies of the data, it will be necessary for the child to output the sequence. Have the parent invoke the wait() call to wait for the child process to complete before exiting the program. Perform necessary error checking to ensure that a positive integer is passed on the command line.

You will fork two processes to print their respective sequence for the Collatz conjecture. The first will produce the sequence which is indicated by the number on the command line and the second process the sequence from the command line number plus 3. Please print the child (1 or 2) with each number output and be sure the forked processes can run concurrently.

Expected output:

```
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,
From Child 2 init n=14, From Child 2 n=7, From Child 2 n=22, From
Child 2 n=11, From Child 2 n=34, From Child 2 n=17, From Child 2
n=52, From Child 2 n=26, From Child 2 n=13, 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,
Children Complete
```

Things to remember:

1. The number entered on the command line must be greater than zero and less than 40.
2. Please put the function code in your file.
3. 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.
4. 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.
5. Submit your code as a ZIP file. Include a README with instructions on how to run and expected output.
6. **If you lock up another machine trying this assignment out, it is a 0 for this assignment!**