Wayne State University

CSC 4421 - Computer Operating Systems Lab
Winter 2107
Lab 5 - Process Control II

Instructor 001: David Warnke
Instructor 003: Rui Chen

Points Possible: 100
Due: 2/27/17

Implement a program that will communicate the value of an integer x, between processes. You can do this using a file to store the value of x, or you can use shared memory with shmget or mmap. *Shared memory is out of the scope of this class, so you will have to teach yourself how to use it, however if you put in the effort to learn, it results in simpler more elegant code.*

x will start with the value 19530, and you must print that value. Then you will loop 5 times where in each loop: the child goes first and subtracts 5 from x, the parent then divides x by 5, and both processes print who they are and their result of x after the operation. If you use a file and not shared memory, then during each iteration of the loop both processes **must** get the value of x from the file, and store the resulting value of x to the file. wait() is a required function call, so you will need to create a new process, and terminate it each time through the loop.

The following system calls are **required** for this lab:

- `fork()`
- `wait()`

The following functions may be useful for this lab:

- `open(), read(), write()`
- `sleep():` make the process wait for a number of seconds
- `atoi():` converts string to integer
- `sprintf():` can be used to convert integer to string
- `fflush():` forces a file buffer to flush. This is useful to flush stdout before creating a child so you don't have two processes printing the same buffer to the screen. This can sometimes fix strange printing behavior.

The following is an example of what you will output to the screen.

```
x = 19530

ITERATION 1
Child: x = 19525
Parent: x = 3905

ITERATION 2
Child: x = 3900
Parent: x = 780

ITERATION 3
Child: x = 775
Parent: x = 155

ITERATION 4
Child: x = 150
Parent: x = 30

ITERATION 5
Child: x = 25
Parent: x = 5
```

Listing 1: Screen Output