CSC 4410 Operating Systems | Spring 2018

Programming Assignment 1

Florida Southern College | *Due Date: February 2*

UNIX Local Interprocess Communication

This assignment will require some thought. Don't procrastinate. I will give some hints about the kinds of library and system calls you should use. Consult the UNIX man pages for more information.

For this assignment you should do the following:

1. Write a main program that takes as the only command line argument the name of an input file (some generic text file).

2. Set up an unnamed pipe [**pipe(2)**], create a child process [**fork(2)**], set up stdin of the child to read from the pipe and the stdout of the parent to write to the pipe [**close(2)/dup(2)**].

3. In the child, exec the program **/usr/bin/more** [**execl(2)**]. In the parent open the input file named on the command line [**open(2)**] and read the input file BUFSIZ characters at a time. Write the data to the input end of the pipe.

4. When the parent is finished sending the file through the pipe to the child it should wait for the child to exit [**wait(2)**]. The child will exit when 'EOF' is sent down the pipe. After the child has exited the parent should exit [**exit(2)**].

Note that this program is harder than it appears. In particular, be very careful to close **all** I/O descriptors when they are no longer useful. Your program will not function properly if you fail to do this. Most of the errors you will encounter will be the result of "still open" file descriptors. Seriously, I've used this assignment many times over the years and probably 90% of the problems students have are due to open file descriptors. They swear they've closed them all but I always find one they forgot. File descirptors: close them! It's not just a good idea, it's the law.

What to submit: By 11:59 PM on the due date you should email a zipped directory containing the following:

- commented source code
- a makefile or README file giving detailed instructions on how to compile and run your program