Washington State University Vancouver

Systems Programming - CS 360

Assignment 4 - Due: 11:59AM Feburary 26

Instructor:
Ben McCamish

February 19, 2020

Overall Assignment - 100 points

Write a program (in C) called `assignment4.c` targeted at the Linux platform that performs like a shell pipeline. **Example:**

```
# ./assignment4 ls : sort = # ls | sort
```

Where the colon breaks argv into a left (`ls`) and right (`sort`) portion. Implementation will fork/exec and setup a pipe such that:

- parent: left portion, runs with stdout = pipe write end
- child: right portion, runs with stdin = pipe read end
- Consider swapping parent and child functions, why?

Program Interface (Required)

```
./assignment4 <arg1> : <arg2>

Where: <arg1> and <arg2> are optional parameters that specify the programs
to be run. If <arg1> is specified but <arg2> is not, then <arg1> should be
run as though there was not a colon. Same for if <arg2> is specified but
<arg1> is not.
```

Specifications and Restrictions

- (60 points - Autolab) Must pass tests on various inputs.
- (20 points - Autolab/TA) Must be robust, including error catching. You must catch errors and print out (to stdout) only message produced by that error using `strerr()`. This means you will need to `errno.h` and `string.h`, libraries at least.
- (Required) Design one source file `assignment4.c`
- Helpful functions: `fork`, `exec`, `dup` (or `dup2`), `open`/ `close`, `exit`, `wait`, etc. (consult man pages as needed)
- Do not use `popen` or `system`.
- **Note:** The specification for this program is intentionally incomplete. Consider various situations and exception conditions that may occur. Determine a reasonable interpretation of the arguments, then design and implement a robust program
- You will need a main. Unlike previous assignments, you are writing and independant program that I will be executing and comparing output. As such, you might remove all print statements before submitting to autolab.

What to turn in (to Autolab):

- assignment4.c (no header files)