Due: Sun Oct 25, 2020 11:59pm

**Unix Shell**

**Objective**

A shell is an interface between a user and the operating system. It lets us give commands to the system and start other programs. Your task is to program a simple [shellLinks to an external site.](https://en.wikipedia.org/wiki/Shell_(computing)) similar to for example [BashLinks to an external site.](https://en.wikipedia.org/wiki/Bash_(Unix_shell)), which probably is the command shell you normally use when you use a Unix/Linux system.

**Preparations**

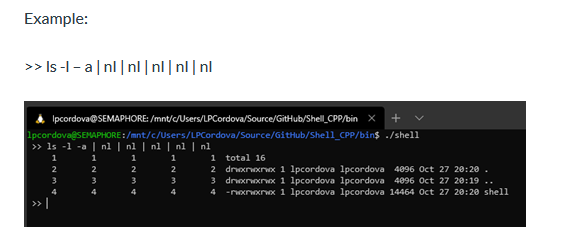
When programming a shell, several of the POSIX system calls you studied already will be useful. Before you continue, make sure you have a basic understanding of at least the following system calls: fork(), execvp(), getpid(), getppid(), wait(), and pipe().

Requirements

Your shell must be able to handle a single command and pipelines with two to sixteen commands.

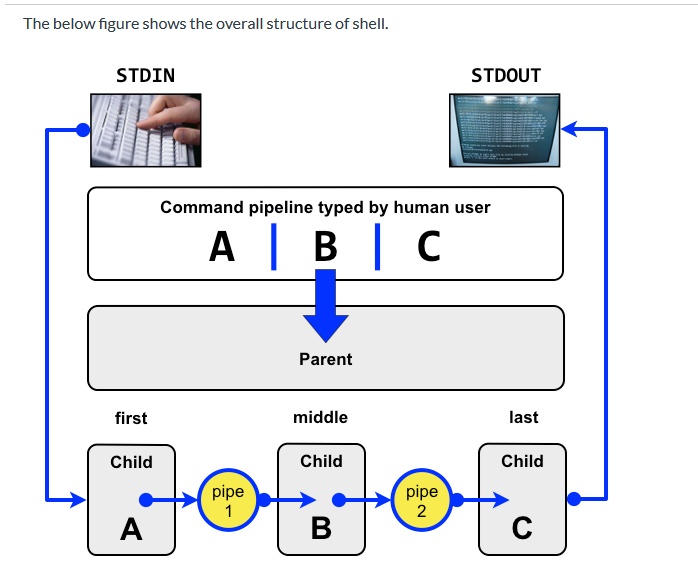
Example:

>> ls -l – a | nl | nl | nl | nl | nl



**Program Design**

The below figure shows the overall structure of shell.



When a user types a command line on the form **A | B | C** the parent parses the user input and creates one child process for each of the commands in the pipeline. The child processes communicate using pipes. Child A redirects stdout to the write end of pipe 1. Child B redirects stdin to the read end of pipe 1 and stdout to the write end of pipe 2. Child C redirects stdin to the read end of pipe 2.

**Stipulations**

1. Use C libraries only for strings (strings.h) and file input and output (e.g. fopen, fclose, strlen, strcpy), and structs (no classes).
2. You must support 2 - 16 commands.

**Submission**

1. Use the class provided Github repository.
2. Place your solution in the Lab-02 directory.
3. Provide a screen shot demonstrating the required functionality (similar to the example). Store in the Lab-02 directory.
4. Create a tag/release on Github.com and submit the tag/release URL to Canvas by the date specified in Canvas.