Project 1

Due October 11, 2018 at 11:59 PM

You will be working with a partner for this project, you specify your partner in your readme file. This specification is subject to change at anytime for additional clarification. For this project, you and a partner will be implementing a fairly simple shell called **ashell**. The shell must be able to execute applications, be able to setup pipes between executed applications, redirect standard input/output from/to files for the executed applications, and perform simple commands internally. You must provide a make file (or cmake file) to compile your shell. Your program must be written in C or C++. You may not use scanf, fscanf, printf, fprintf, cin, cout, or their derivatives; you must use read and write system calls for all I/O. You may use the C++ containers. If you have any concerns if a standard library function is allowed, ask before using it. Submit only one copy of your project per pair on Canvas as tgz file.

The shell must handle commands cd, ls, pwd, ff, and exit internally. Below are descriptions of the commands:

`cd [directory]` – Changes the current working directory to specified directory. If the optional directory is not specified, then the current working directory is changed to the directory specified by the HOME environmental variable.

`ls [directory]` – Lists the files/directories in the directory specified, if no directory is specified, it lists the contents of the current working directory. The order of files listed does not matter, but the type and permissions must precede the file/directory name with one entry per line. Below is an example listing

```
drwxr-xr-x .
drwxr-xr-x ..
-rwxr-xr-x ashell
-rwxr-xr-x main.cpp
```

`pwd` – Prints the current working directory name.

`ff filename [directory]` – Prints the path to all files matching the filename parameter in the directory (or subdirectories) specified by the directory parameter. If no directory is specified the current working directory is used. The order of the files listed does not matter. Below is an example of a listing of `ff main.cpp ECS150`:

```
ECS150/Project1/main.cpp
ECS150/Project2/main.cpp
ECS150/main.cpp
```

`exit` – Exits the shell.

The shell needs to be able to handle up and down arrows to select through the history of commands. Only the most recent 10 commands need to be stored in the history. The up arrow

must toggle to the previous command, down arrow to the next command. If the user attempts to navigate beyond the end of the list the audible bell character '\a' (ASCII 0x07) must be output. In addition, if a user enters a backspace or delete and no characters exist on the command to delete, the audible bell must be output.

The shell prompt must have the current working directory followed by the percent symbol. If the full current working directory path is longer than 16 characters, then "/…" should be prepended to the last level directory name. Below are examples:

```
/home/cjnitta%
/.../Project01%
```

A working example can be found on the CSIF at /home/cjnitta/ecs150/ashell.

You should avoid using existing source code as a primer that is currently available on the Internet. You **must** specify in your readme file any sources of code that you or your partner have viewed to help you complete this project. All class projects will be submitted to MOSS to determine if pairs of students have excessively collaborated with other pairs. Excessive collaboration, or failure to list external code sources will result in the matter being transferred to Student Judicial Affairs.