**Authors**

* Sikai Ni
* Xuan Fei

**Description**

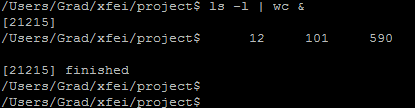
* This Shell supports the built-in commands and external commands by spawning child processes.
* The project includes a Makefile which produces an executable called shell when you run command make.

**Shell Features**

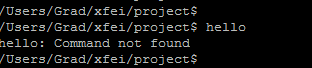
* The ‘$’ character is shown before any commands input.
* The shell has two mode: foreground mode and background mode.
* In foreground mode, the process runs immediately and the input function is closed until the process finished.
* When a ‘&’ character is detected, the command line runs in background where the Pid of this process will be printed immediately and the other command will be proceed simultaneously.
* When a sub-command is not found, the shell reports the error and accepts new command.
* When the input redirection operator ‘<’ is detected, the shell redirects into the first file followed by ‘<’. When the file cannot be found, the error massage is sent by system.
* When the output redirection operator ‘>’ is detected, the shell redirects into the last file followed by ‘>’. When the file path is invalid, the error massage is sent by system. Or the file can be accessed by function cat.
* In our shell, system can accept 5 sub-commands at the most. 4 pipes connect these sub-commands so that they read and write data through pipes. We create pipe in parent and close all the unused file descriptor after setting the pipe’s sending-end and receiving-end.
* The path can be shown before the prompt and command cd can change the directory.
* When the command runs in background, a message can be printed after the process finished and the user presses Enter.
* The shell can quit using command exit.

**Execution Demo**

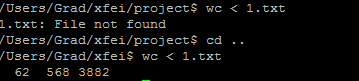
* The figure shows a command with two sub-commands runs in the background mode. We can see the pipe works properly between the sub-commands and the Pid shows on the screen. Besides, when the process finished, a message shows after presses Enter.



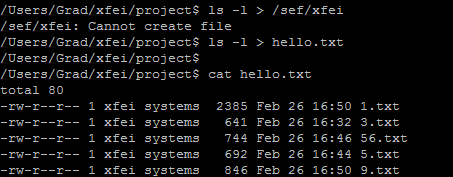
* When an invalid command inputs, an error message is printed.

****

* When the system access invalid text, an error message print. The command cd changes working directory.

****

* The output redirection operator ‘>’ can redirect to the right file when the path is correct. The error message can print when path is invalid.

****

* The system can exit using command exit.

****