

Anthony Kim

2/22/2019

CSC345-02

## Project 1 Report

I implemented a UNIX shell that takes a user's varying commands. Users can pipe up to two commands, run programs in the background, use files as input or output for commands, change directories using the "cd" command and execute the user's last command with "!!".

Piping was implemented by finding the '|' character using the strchr command then using pointers to create two separate strings, the first command and the second command. Then these strings are used as input for a function that fills an array of pointers to characters for each string. Then these arrays are used as input for a function that creates two forks and uses dup2() to direct the output of the first command to the input of the second command. The parent process waits for both processes to finish.

Running programs in the background was implemented by finding '&' in the command then using setpgid() in the child process. The parent process when this is running does not wait for the child process to finish. To test this I ran lab02\_ex3 then used the ps command to look at the processor state.

File input/output was implemented by finding '<' or '>', then creating a string that copies the filename. The filename is used to open() or creat() and dup2() to direct input or output to the file. The nums.txt file can be used as an input for "sort" and any name can be used to create a file for output of a command.

Changing directories was implemented by using getcwd() to get the current directory and strcat() to add the desired destination to the current directory. After, chdir() is used to change directories to the desired destination. If the path name contains '/' in the beginning, the program uses chdir() to change directory without concatenation.

Executing user command with "!!" is implemented by saving the previous command that was not "!!", and loading this saved command into the array of arguments. Once the array is constructed, it is used as input for the function that executes commands using child processes.