CSC 345 Project 1 Report
Vivek Bharadwaj
Section 01

Requirements Implemented:

1. Child processes are created and execute the commands. If the child process is created successfully, it will execute the actions of the child process if (result == 0). However, if (result == -1), it will fork unsuccessfully and show an error. Furthermore, if (result > 0), then it will be in the parent process and wait for the child process to complete.
2. Command history using !! is implemented. This happens when you have executed a command and executed the previous command entered when typing !!. A history array was implemented to hold the previous commands entered by the user. However, I was not able to figure out a way to use arrow keys to support command history.
3. Input and output redirection is supported within the program. I was able to test the output direction by creating an empty file1.txt file and using the command (ls > file1.txt) which wrote all the contents of the ls command to the file. For input direction, I tested it by using cat < file1.txt which returns all the contents of file1.txt.
4. Allow the parent and child processes to communicate via a pipe is implemented within the program. I was able to test the pipe by using the ls -l | less command which returns the details of the contents within the current directory including date, time, and owner permissions.
5. Working directory information is kept and implemented. This can be tested by using the cd command to change into another directory. Once you have executed that command, use the ls command to see the details of the current directory that you have changed to.
6. An up-to-date display of the current directory in the prompt is implemented. This was implemented by finding the last '/' in the path name and printing the characters that follow after (osh>). This can be tested by using the cd command to change into a different directory and the command prompt will include the current directory name.