Part 1

This program you will implement a command-line interpreter (a.k.a. a shell).

The shell operates in this basic way: when you type in a command (in response to the prompt), the shell creates a child process that executes the command you entered,

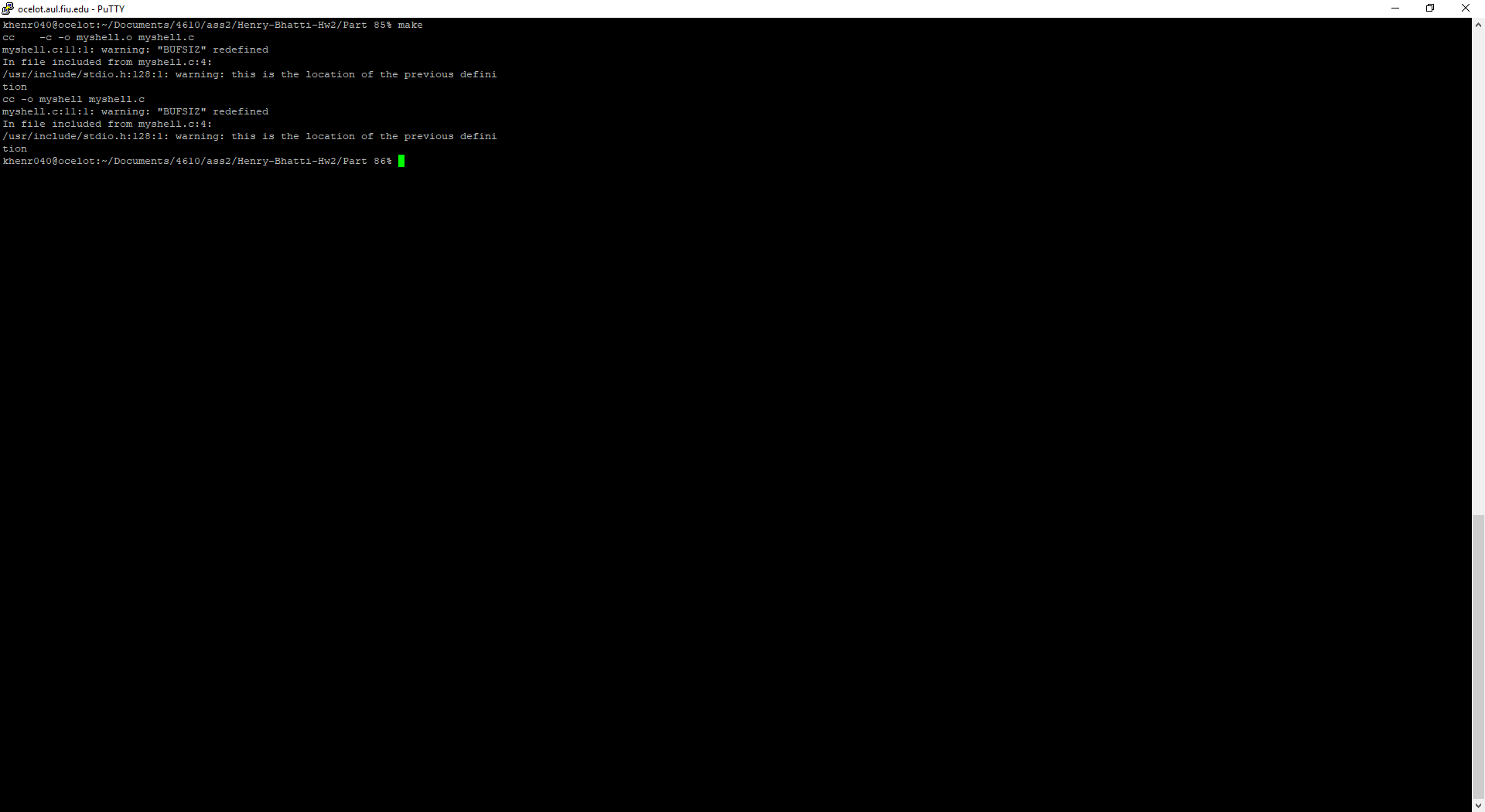
displays the output of the command, and then prompts for more user input when it has finished.

To compile the program, type the make command in the terminal. To execute the program use "./myshell" into the terminal.

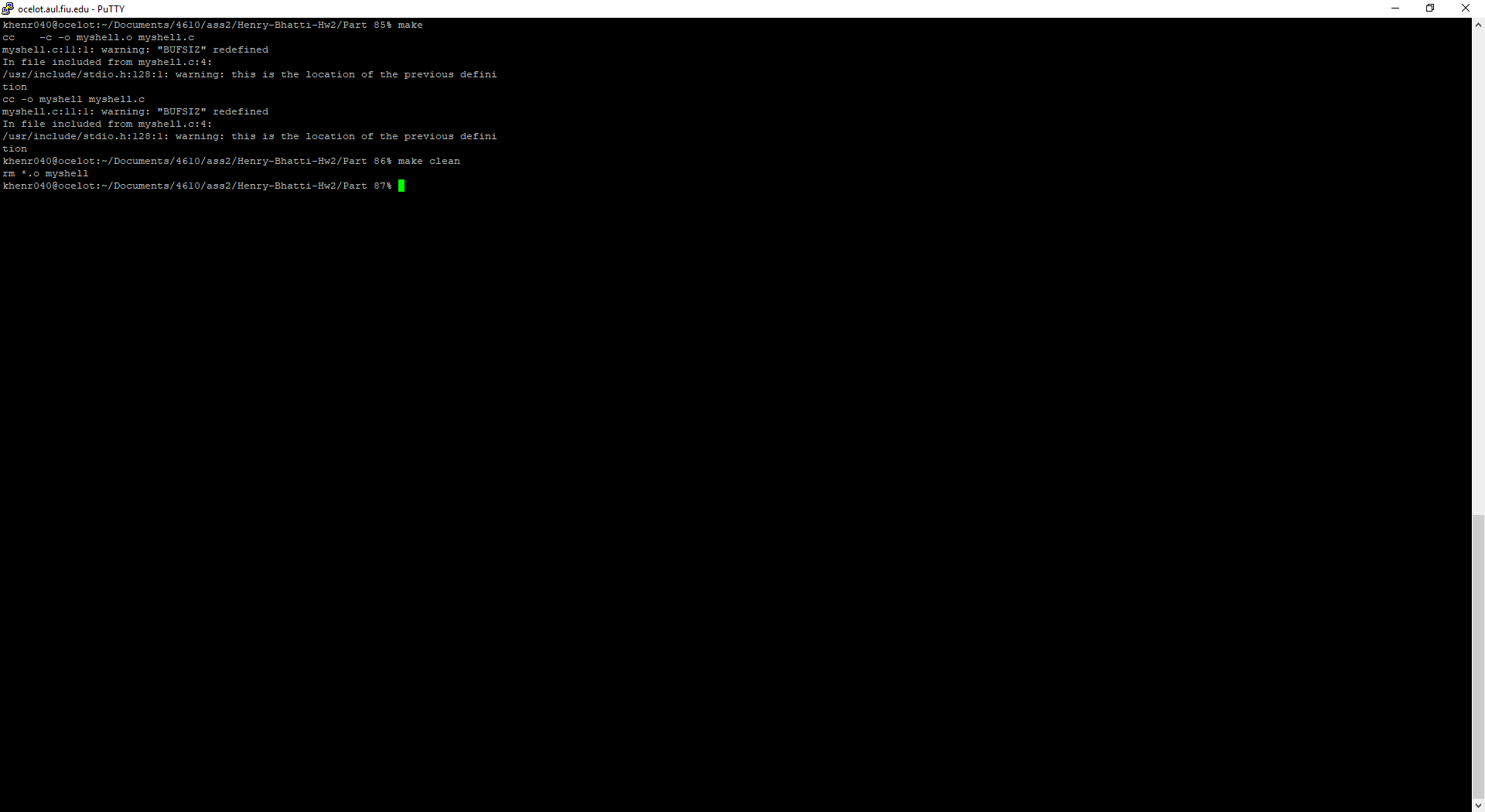
This will start the program and the shell. Enter your commands after the prompt. Use an ampersand (&) after a command to have it run in the background. The process ID is printed.

Use semicolon (;) to have several commands run concurrently. The process IDs are printed.

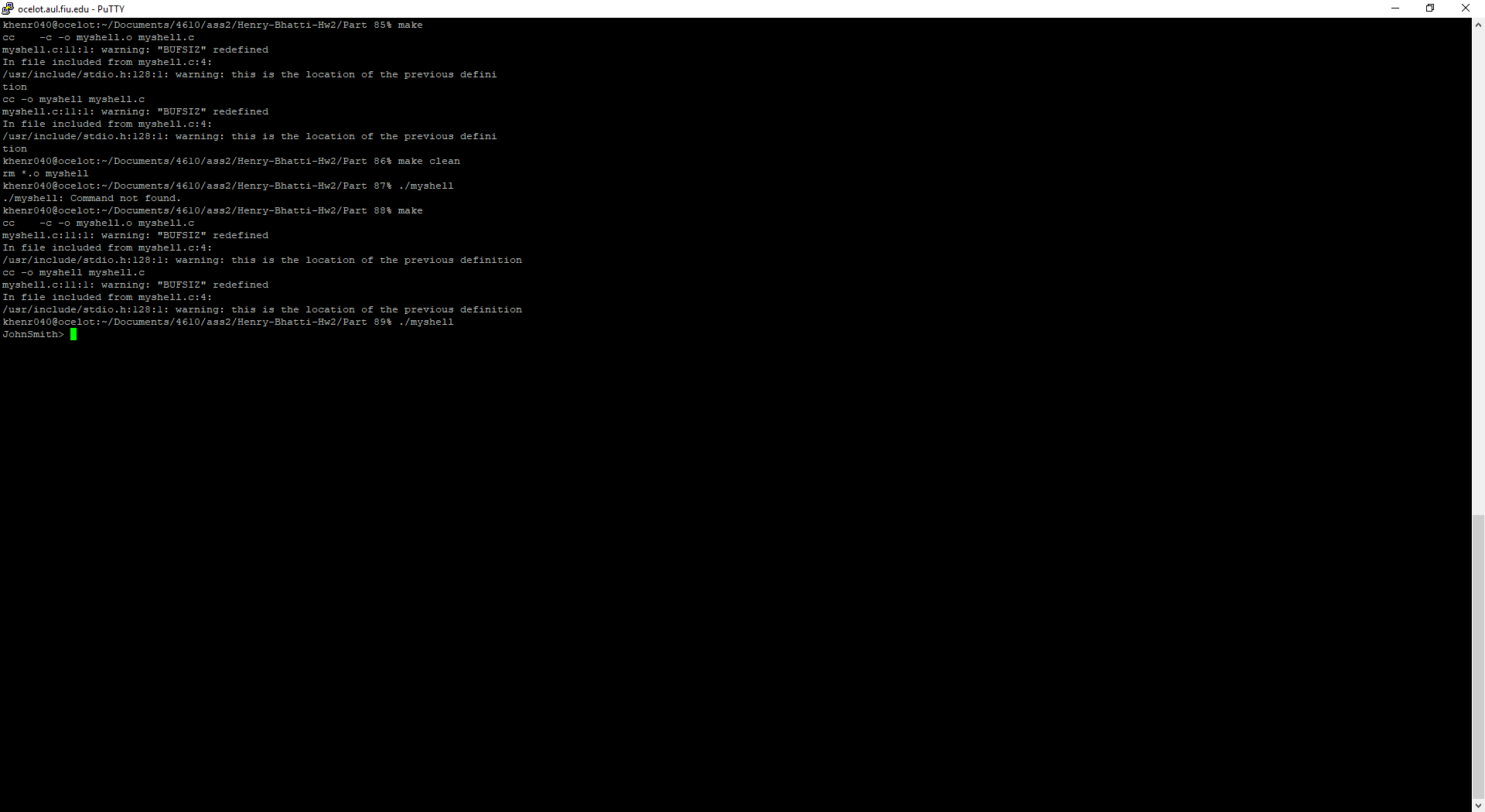
Command: make. Used to compile the program.



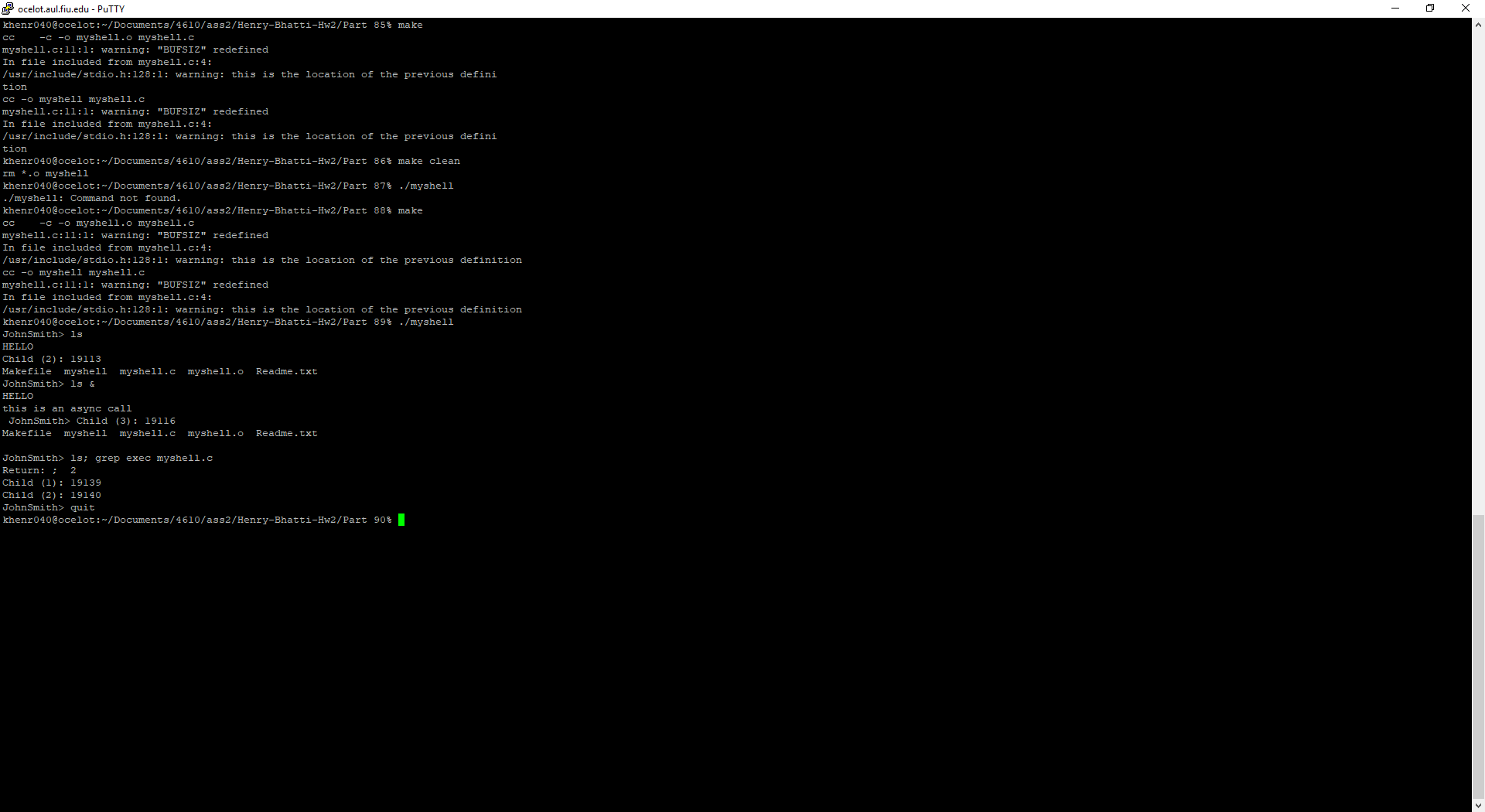
Command: make clean. Used to remove temporary files.



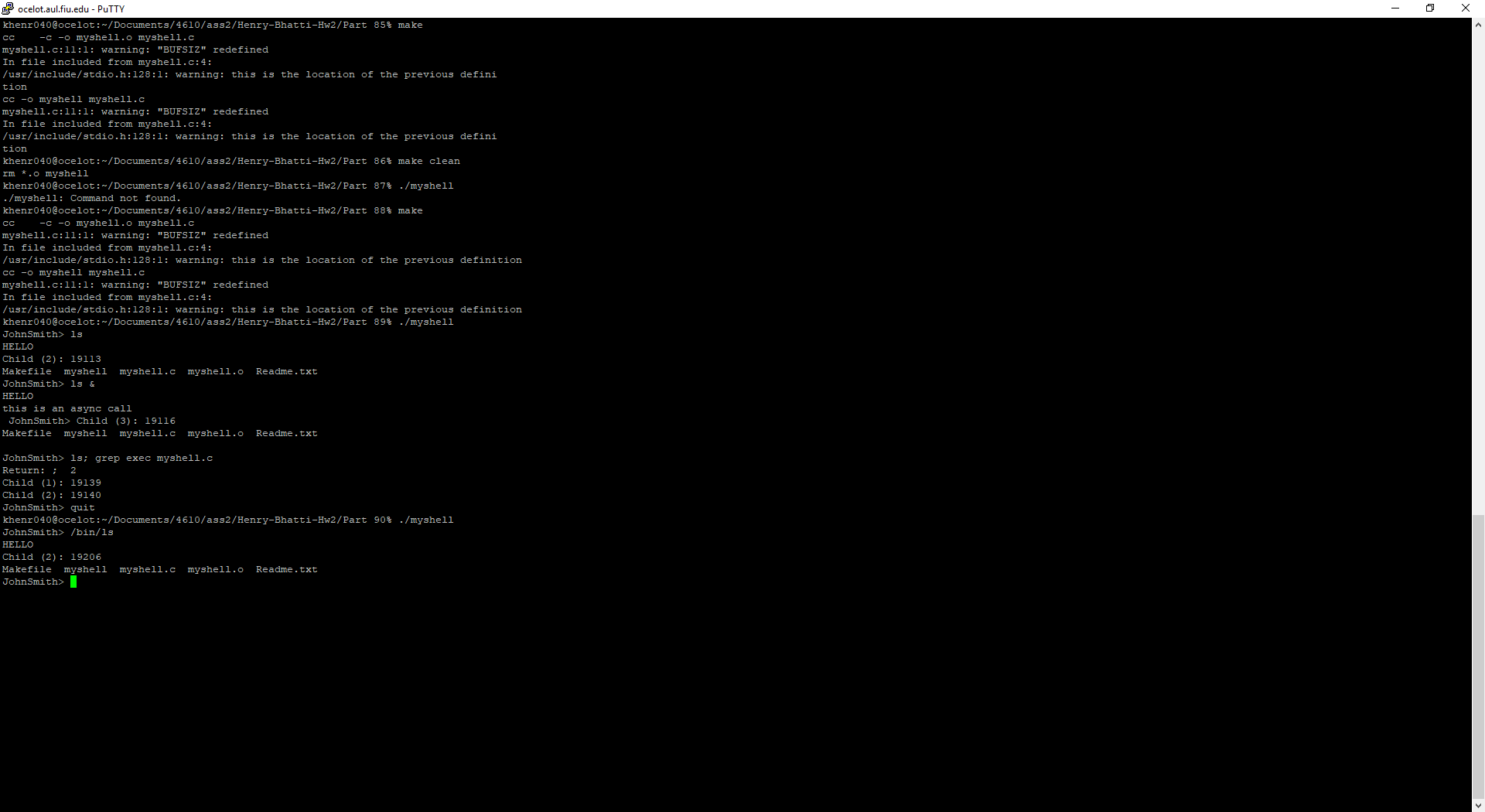
Command: ./myshell. To Begin the shell.



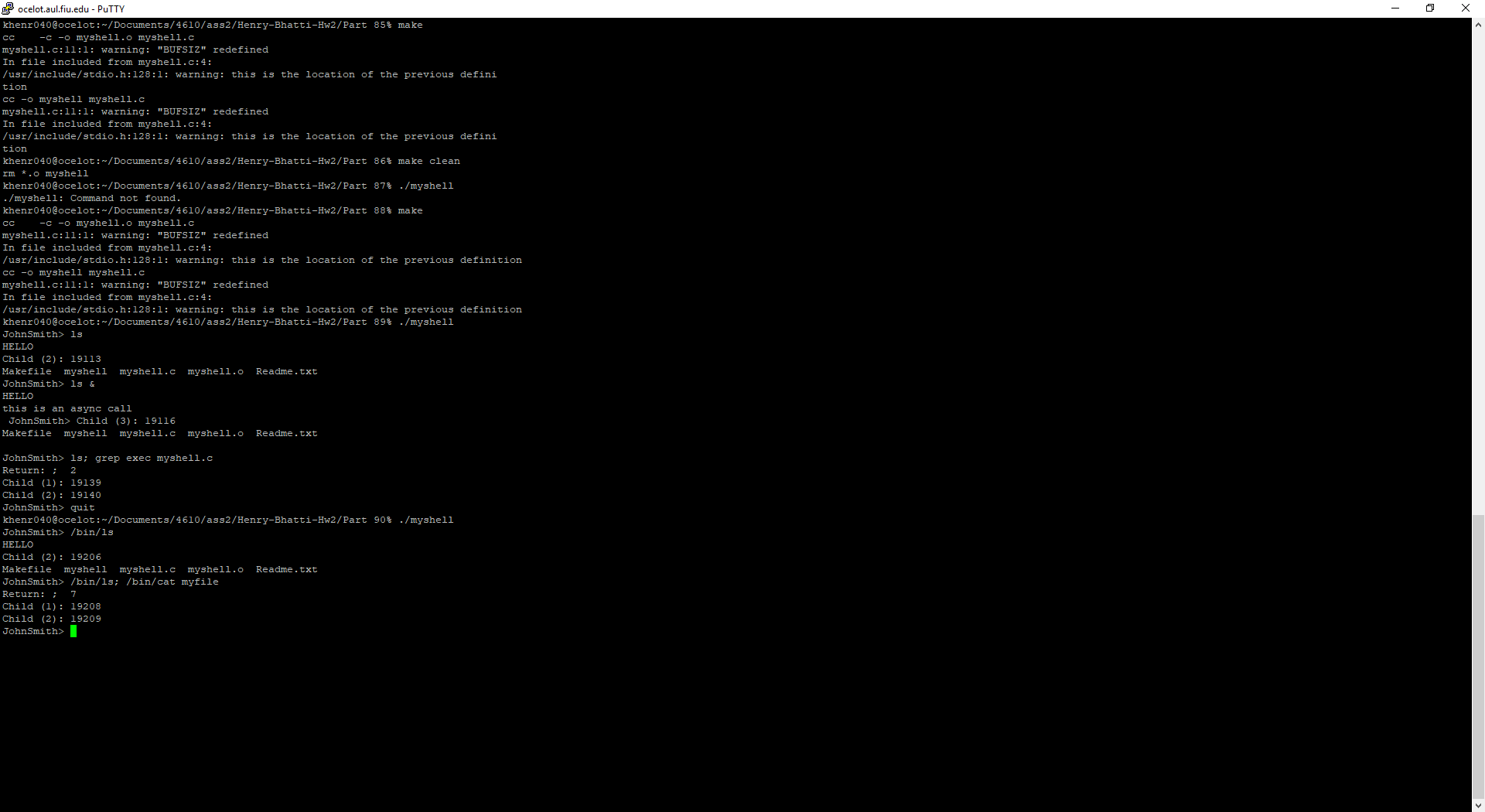
Command: quit. To exit shell.



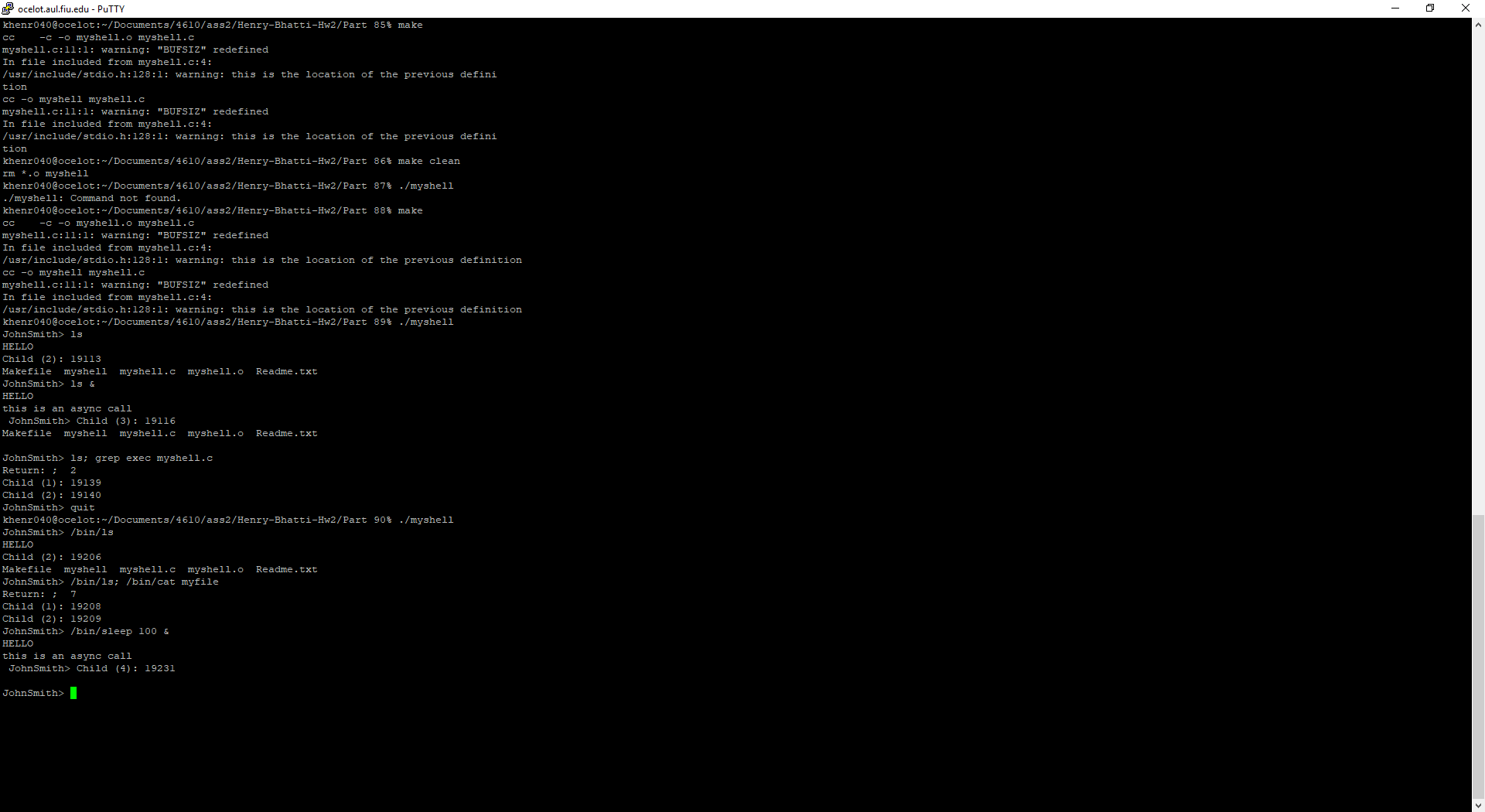
Command: /bin/ls List files in directory.



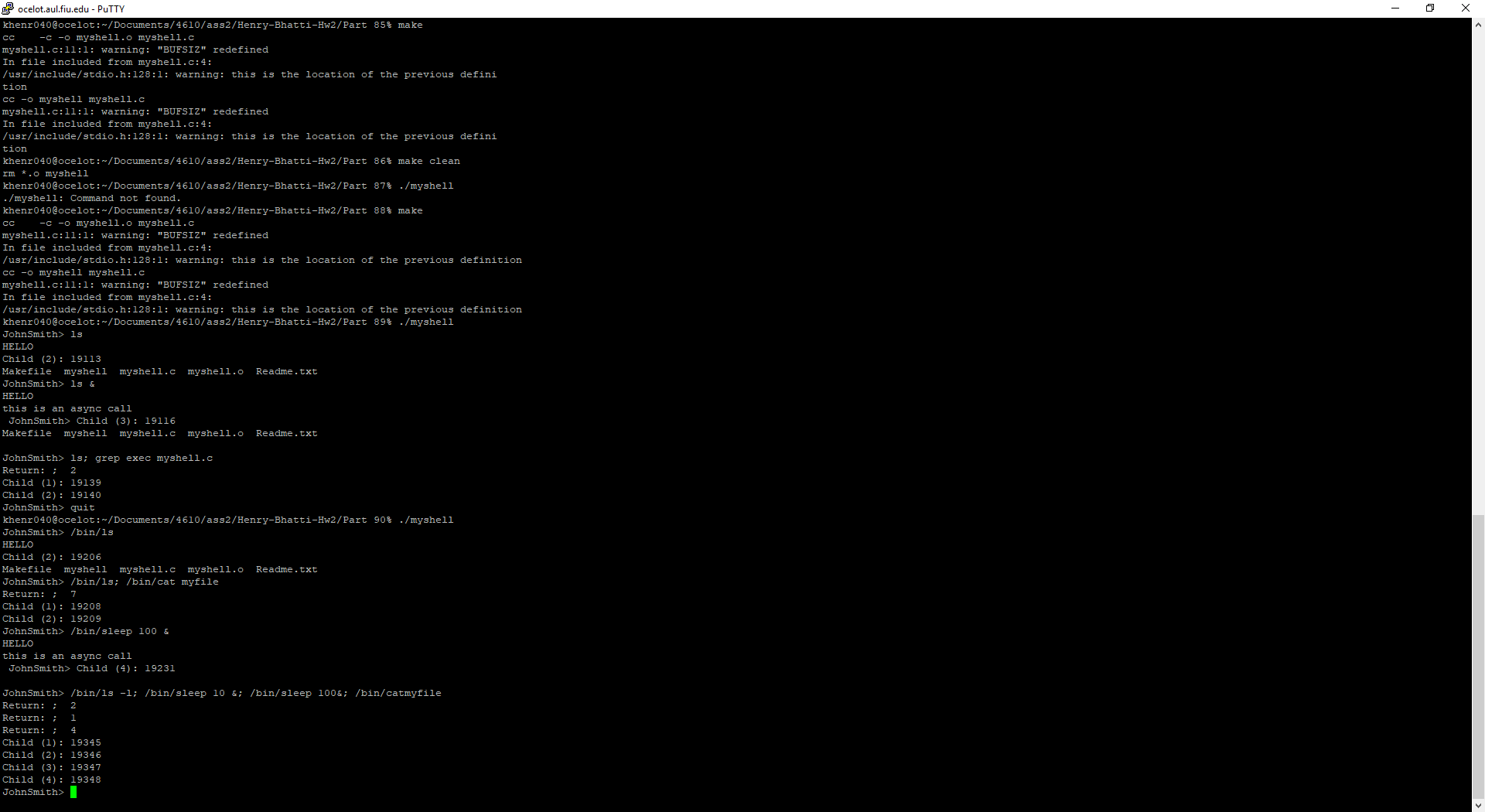
Command: /bin/ls; /bin/cat myfile Run ls and cat myfile concurrently. It is unable to show the results of running both commands but it does make the children processes.



Command: /bin/sleep 100 & Run command in the background. The background process is printed.



Command: /bin/ls -l; /bin/sleep 10 &; /bin/sleep 100&; /bin/cat Run background and other processes concurrently. It runs the background commands and makes the child processes.



KaiJana Henry worked on the Part 1 of the project. The different commands, the makefile, readme and report.