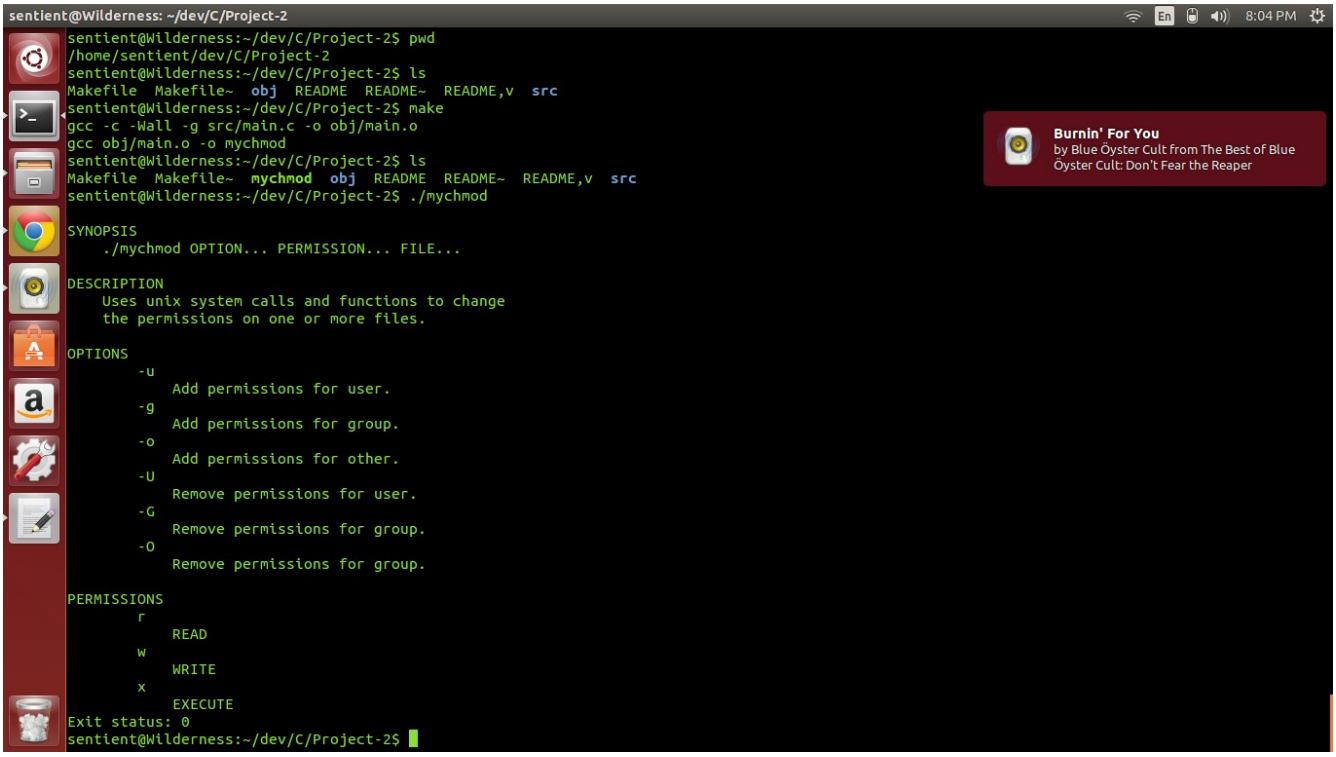
Dan Wilder
CS 2750
14 March, 2015

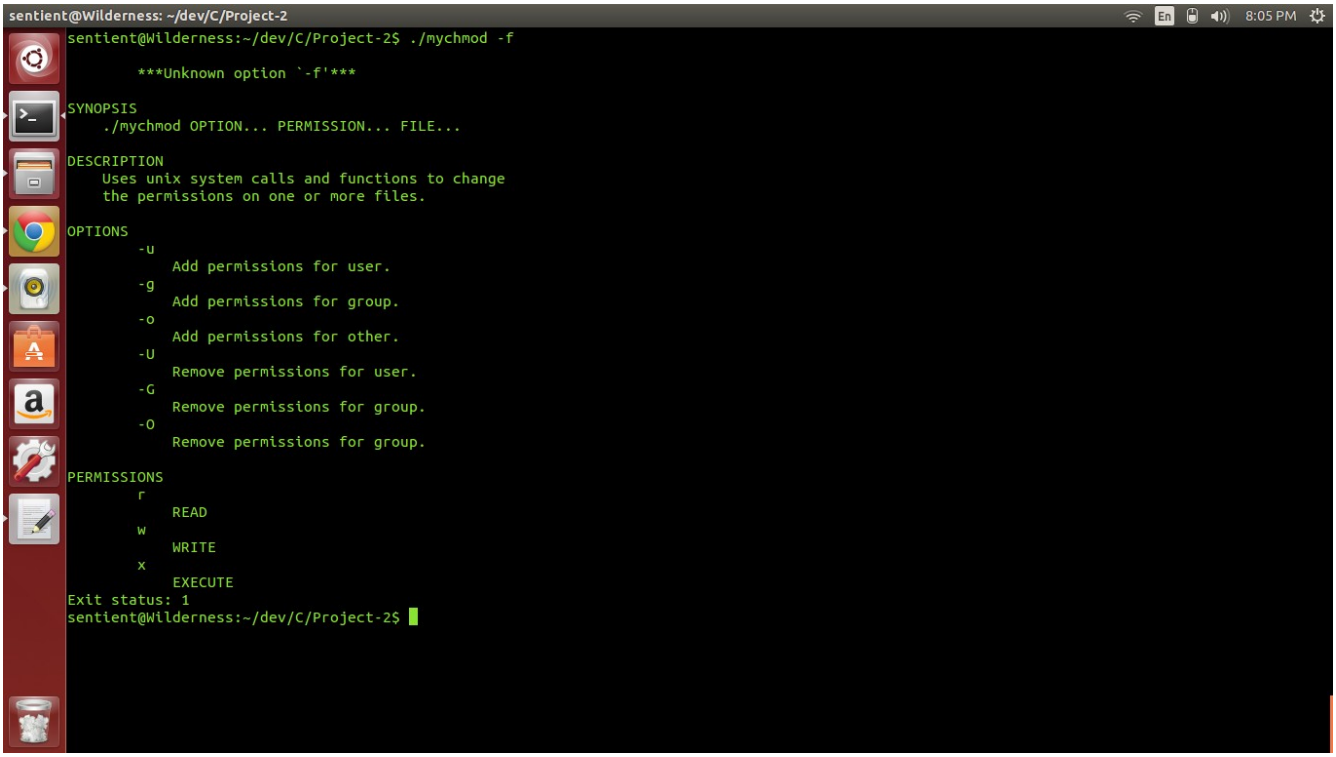
Report: Project 2

This project involved using Unix system calls and functions to change the permissions on one or more files. The first challenge was flexible parsing of command line options. This was accomplished by the use of the C library function getopt. Valid options included: u,g,o,U,G,O. For each of these options that were present, the program would check each character of optarg which corresponded to file permissions. Two variables of type mode_t called addPermissions and removePermissions were created and initialized to 0. If a valid a permission character was present in optarg, a bit in in either addPermissions or removePermissions would be set to 1. The specific bit to be set depended on which option and which permission character (r, w, x). U,G,O corresponded to removePermissions and u,g,o to addPermissions.

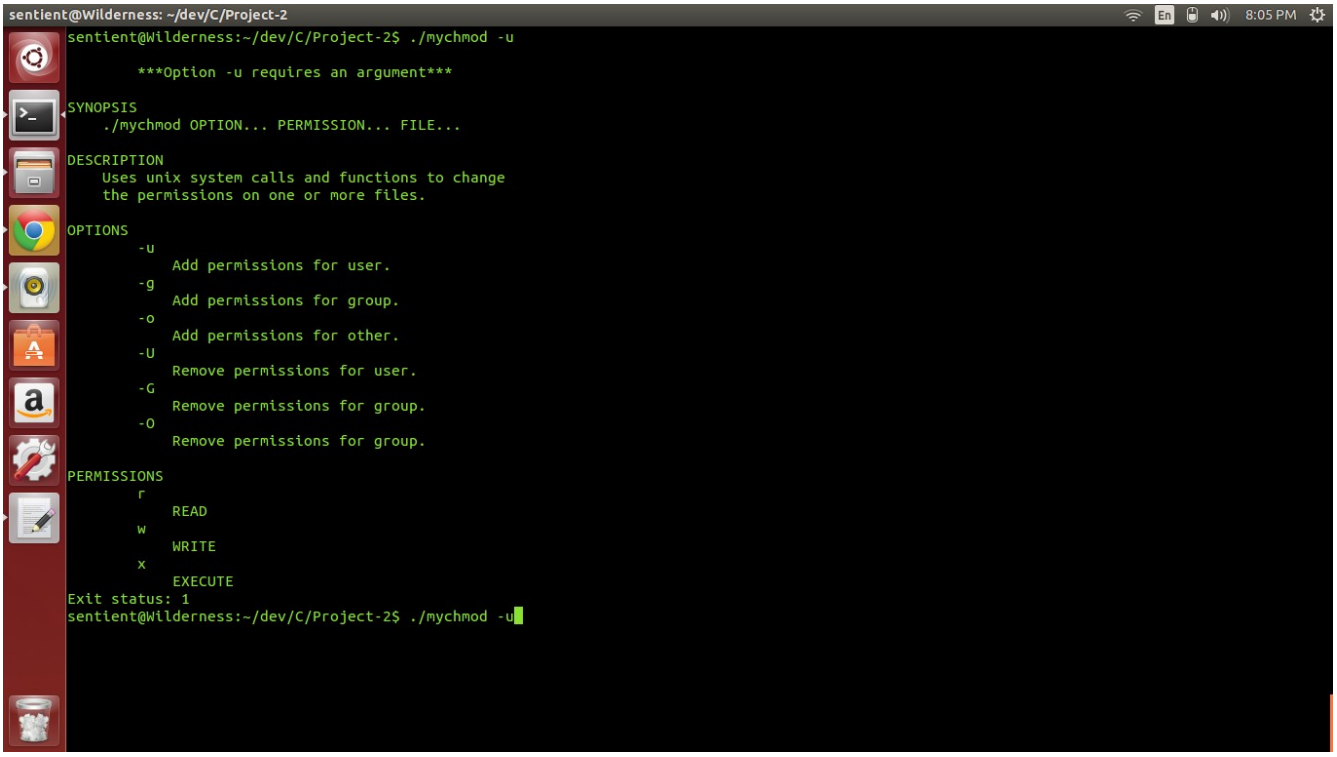
The function getopt also made it easy to check remaining command line arguments which in the case of this program meant files. The function stat was used to check if a file existed, and if so stored information in structure variable 'fileStat'. This variable was used to retrieve original permissions of a file. Using bitwise operators with the original permissions, addPermissions, and removePermissions allowed me to compute the final mode that a file should be in. This, along with the file name was passed to a system call to chmod.



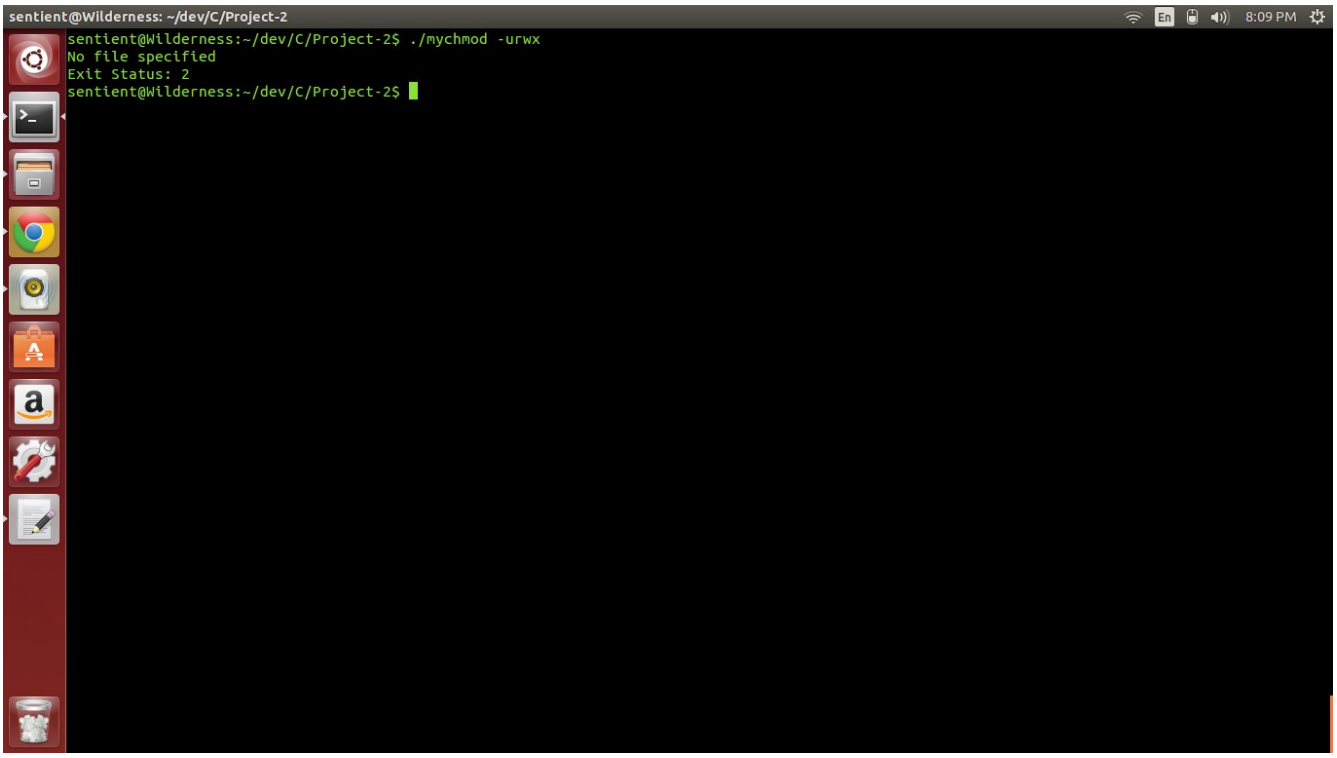
Initial compilation and running without arguments



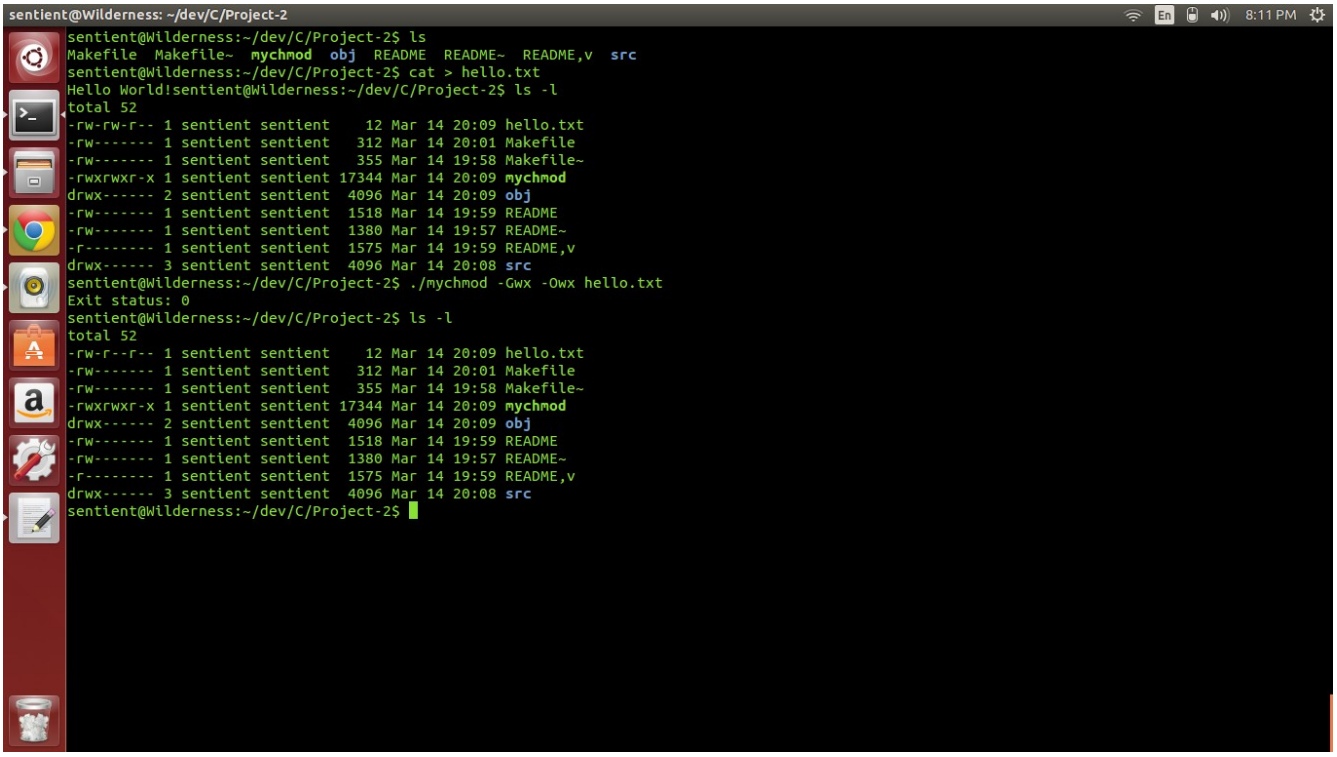
Running with invalid option



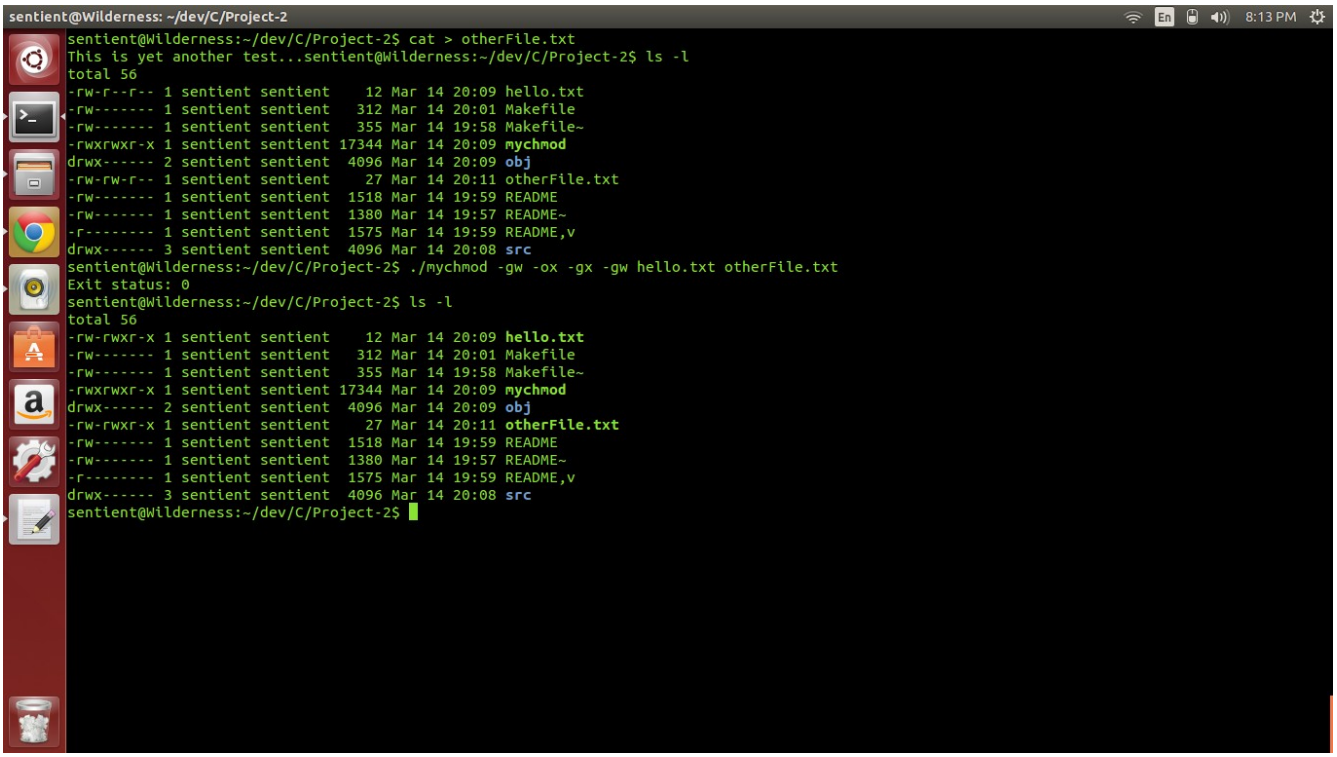
Running valid option without argument (permissions) following



Running with valid option and permissions, but no file specified



Proper running with single file; ls -l to show results



Running with options split up and multiple files specified

Conclusion

This was a fun program to create as it involved experimenting with parsing command line options and using system calls. I especially found command line parsing useful, and I will incorporate into my future programs. I learned a lot about various libraries, and a greater understanding of how more complex C programs are created.