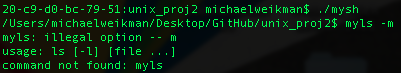
**Section III (Task II)**

**Myls:** The myls function accepts one option ‘-l’. When executing myls without option it will list the files/folders of the current directory in row column format similar to ls. The files/folders are listed in alphabetical order in a cyclic orientation based on position in the array structure. I used FTSENT functions to parse provided or current (if not provided) directory information and store the content information in the returned structure. My program is dependent on two .c files print.c and util.c and one header file myls.h. The two .c files contain functions related to the printing and formatting of the list of files while the header files contains the relevant prototypes and extern variables that all files are dependent on. FTSENT structures contain all information required to perform the operations of this command, including file name, file name length, stat information, and error information. I created a primary struct in the header file FINFO to hold length (character space) of particular attributes of the file to format the long format listing to reflect that of the original ls function. The util.c contains functions needed to perform actions needed to gather length of numbers (character space) and translating st\_mode to readable permission format (i.e \_r\_w\_x).

Essentially, how the command works with respect to the code, is the command is executed with or without an option, the option of which determines the printing format. If no option is provided the default print is row column format of just file names and if the option –l is provided it prints single column format with file attribute information similar to the normal ls command. If a single file is provided as a parameter to the command it will print just that file information based on the option selected. If a directory is provided it will traverse the directory tree and print files in that directory. If not directory or file is provided it defaults to the current directory and prints all files and folders in the current directory in the format selected by the provided or non-provided option. In the case of directories if the permissions of the directory don’t allow traversal or read access of the directory the command outputs error message explaining permission related error. Parent directory ( . . ) is ignored to prevent traversing parent directories. Hidden files are also ignored due to the project not requiring the displaying of them.

Two issues that need to be handled when handled when implementing this function are permissions of directory folders that may be traversed for listing and the use of an option that is not of the list of options available by the command. To handle the permissions issues, I kept track of return errors from functions used to traverse directory trees and used functions in the err.h library to print the errors and corresponding error message. For the option handling, I created a usage function that is called when any option other than –l is used. The usage function prints out to stderr the reason for the error and the usable option information for the myls command.

Invalid option error



No traversal permission for directory

