CSE410: Operating System (Spring 2018)
Project #1: Implementation of "Find" Function

Deadline

Thursday, Feb 1 2018 at 11:59pm (Handin) https://handin.cse.msu.edu/

Assignment Goals

Get familiar with some system functions that Unix-based systems use for file handling.

Assignment Overview

In this project, you are required to design and implement a C++ program that allows the user to find the specific file based on different file attributes (name, permissions, last modified time, etc.) by looking at the designated search path.

Assignment Specifications

In this project you are required to implement the following tasks (total 100 points).

- **Basic "find" function implementation (80 points)**

  The "find" function works as follows:

  ```
  myFind <search path> <option> <target>
  ```

  search path – the absolute path for the program to look for the target

  options – the type of the search target (as shown in the table)

Category	Sub-Category (10 points each)	Option	Format	Example
File Attributes	File Name	-name	string (char *)	myNote*
	File Size	-size	long int	64000
Ownership	User ID	-uid	integer	178
	Group ID	-gid	integer	230
Time Attributes	Last Access Time	-atime	"%T-%D"	"12:22:16-01/01/12"
	Last Modify Time	-mtime	"%T-%D"	"12:22:16-01/01/12"
	Last Status Change Time	-ctime	"%T-%D"	"12:22:16-01/01/12"
File Permissions	Permission Number	-perm	integer	755

  (Note: T is in the format of HH:MM:SS and D is in the format of MM/DD/YY)

  target – the search target

- **Multiple options search (10 points)**

  The user can search for a file based on multiple options and targets.

  ```
  myFind <search path> <option> <target> <option> <target> …
  ```

- **Basic programming requirements (10 points)**

  Your program is required to have a proper error handling, and appropriate output. The code should be commented thoroughly and each function should be declared clearly.

- **Recursive search (Extra Credits, 10 points)**

  Extra bonus (10 points) is considered for recursive search in which your program recursively search in all subfolders of the search path. This option should be presented by "-R" option.

Assignment Deliverables

A zip file named according to your NetID (e.g. john9999.zip where "john9999" is the NetID) containing the following files.

Project1.cpp – source code of your solution

SearchTask.cpp – the source file we provided

SearchTask.h – the source file we provided

Makefile – make file for generating an executable "myFind"

Readme – if you have implemented the task for extra credits or anything you want us to know when grading the project, please mention about it in this file.

The code should work on "cse410.cse.msu.edu". Any compilation error as well as segmentation fault problem would not be compromised. So please make sure your program is clear from any compilation and runtime errors before submission. Note that your project file should be submitted with specified file name via the "Handin" system.

Assignment Notes

1. You can find all the required source files by downloading the "Project1.zip" file. In this archive file, you can find the following items.
   - ***Makefile***
   - ***Project1.cpp*** (you have to implement the Find() function here)
   - ***SearchTask.cpp***
   - ***SearchTask.h***
   - ***myFind*** (this is for demonstration and you can play with it by executing it on "cse410.cse.msu.edu" for further understanding about the required tasks of this project) **NOTE: do NOT hand in this file please.**
   - ***lab01* files***
   - ***lab02* files***
   - ***lab15* files***
2. You may refer to Lab01, 02, and 15 files for some introductory information.
3. The following manual pages may be of interest for this assignment
   - man –s 3 opendir
   - man –s 3 readdir
   - man –s 2 stat
   - man –s 3 localtime
4. If you are not familiar with the "make" utility, you may wish to take a look at the following content.
   `~cse410/General/intro.make`
5. Also note that permissions in this project are represented by permission number which contains three octal digits where first digit is dedicated to the file owner, the second one for the group members and the third one is assigned to the other users. Each digit is calculated based on the following table.

Octal digit	Text equivalent	Binary value	Meaning
0	---	000	No access granted
1	--x	001	Execute access allowed only
2	-w-	010	Write access allowed only
3	-wx	011	Write & execute allowed
4	r--	100	Read access allowed only
5	r-x	101	Read & execute allowed
6	rw-	110	Read & write allowed
7	rwx	111	Read, write, and execute allowed

For instance, number 752 represents the following permission policy:
Owner: read, write, and execute permissions.
Group: read and execute permissions.
Other: only write permission

6. Some useful links

- Basic Unix Commands http://www.cse.msu.edu/Facility/Howto/BasicUnixCommands
- Unix File Permissions http://www.cse.msu.edu/Resources/Facilities/unixpermissions
- X2Go http://www.cse.msu.edu/Facility/Howto/X2Go