Operating Systems Programming Assignment #6

A File Find Utility

Prof. Li-Pin Chang, NCTU

Introduction

- Implement a subset of the “find” command
  - Finding files by name, inode #, and file size
  - Recursively descending into sub-directories to find all matches

Command usage

- my_find [pathname] [options]
    - pathname
        - Any path name, including those containing . and ..
    - options:
        - -inode <number>
        - -name <filename>
        - -size_min <size in megabytes>
        - -size_max <size in megabytes>
        - Will be used in combination
        - The order of options can be arbitrary

Examples

- my_find . -inode 100
  - Find the file whose inode number is 100
- my_find ./sub1 -name test.txt
  - Find the file whose file name is “test.txt”, starting from the sub directory “sub1” of the current directory
- my_find ../sub2 -size_min 10
  - Find all the files whose sizes are >= 10MB, starting from the sibling directory “sub2” of the current directory
- my_find . -name foo -size_min 1 -size_max 10
  - Find all the files whose names is “foo” and sizes are between the range [1MB, 10MB]

Output Format

- Print the following entry for each match
  - [full path-file name][inode#][size in MB]
- Examples
  - ./sub1/foo.txt 233 12.2 MB
  - ./sub1/sub2/bar.txt 222 0.2 MB

- Note: 1 MB stands for 2^20 bytes, not 10^6 bytes
  - (MiB)

Related APIs

- <sys/dirent.h>
  - opendir(), readdir()
  - access directory entries
- <sys/stat.h>
  - stat()
  - access file metadata

Grading Policies

- Upload file name:
  - $(Student_number)_find.c/cpp
  - A wrong file name causes a 10pts penalty
- Do not plagiarize
- No example directory trees will be given, test your program with your own directory tree

Remarks

- Search is always recursive
- Files and directory are both targets for searching
  - But exclude . and ..
- The size of a directory is the size of the directory iteslef, not the total size of files under the directory
- Output relative path, not absolute path
- No wildcards (? or *) in file names

Testing OS Environment

- Ubuntu 16.04, Ubuntu 14.04 or CS linux work station
  - gcc my_find.c -o my_find
  - Your code should compile successfully in one of the above environments