CS 481/ECE 437 Lab 5: My “ls”

Due Friday, May 6

- **This assignment is to be done individually.**
- You must complete this assignment using C.
- You may discuss this lab with others, but you may not share code.

Concepts: Reading directory and file entries; reading directory and file attributes

Synopsis

`myls` [OPTION] ... [FILE] ...

Description

List information about the specified FILEs or current directory if no FILE is specified.

- Print each entry one per line.
- Print contents in the order returned by `readdir()`.
- Always print *hidden* entries of the form ’.*’ including ’.’ and ’..’
- Print normal output to `STDOUT` and error output to `STDERR`.

Options:

- *-c, –classify*: append file type indicators: ’/’ for directories; ’@’ for symbolic links; ’*’ for executables.
- *-d, –disk-usage*: specify file size on disk, based on its number of allocated blocks. **This option takes the file system block size (in bytes) as a mandatory parameter.**
- *-l, –long-listing*: use detailed (long) listing format, printing in order:
  - `inode` number
  - `mode` (using the same “drwxrwxrwx” format of the standard `ls` command.
  - `owner` (you must convert the `uid` to the proper user name)
  - `group` (you must convert the `gid` to the proper group name)
  - `file size` (in bytes unless otherwise specified)
  - `file modification time`
    - if file has been modified in current year, use Month Date hh:mm format, e.g. “Apr 17 05:03”
    - otherwise, use Month Date Year format, e.g. “Apr 15 2013”.
  - `file name` (not including any leading directories)
- *-f, –follow-symlinks*: follow the targets of symbolic links

- *-h, –human-readable*: if long listing format is specified, print file sizes in a human readable format using B for bytes, KB for kilobytes, MB for megabytes and GB for gigabytes, for example, 234B, 234.2KB or 8.7MB.
- *-r, –recursive*: list subdirectories recursively using depth-first, pre-order traversal. first print the contents of the current directory then recursively traverse all sub-directories of the current directory.

Guides & Tips

For this final assignment, hints and tips are minimal:

- There are standard C routines that may prove helpful for command line parsing.
- Manual pages are your friends.

What to turn in

YOU MUST FOLLOW THESE INSTRUCTIONS PRECISELY.

You should turn in the following:

- **all** .c and .h files needed to build your executable files, including any you've downloaded. **Do not turn in any object files (.o) or binary executable files.**
- the single `makefile` that builds the `myls` program
- An optional `writeup.txt` file with any disclosures or other relevant information.

When you are ready to turn in your assignment:

- Place the requisite files in a directory named lastname_lab5 where lastname is your last name;
- Move to the parent directory that contains this lab5 directory;
- execute the command: `tar -czf <lab5_dir>.tgz <lab5_dir>`

  where <lab5_dir> is the name of the directory containing your lab5 files.
- This will create a new file <lab5_dir>.tgz containing the contents of your <lab5_dir>. You can verify the contents of this ”compressed tar file” with the following command: `tar -tzf <lab5_dir>.tgz`
- Submit the file `<lab5_dir>.tgz` via UNM Blackboard.