COSC 3750
Linux Programming
Homework 5, Write "wyls"

1   Intro

This is another "simple" utility. This program MUST compile and run on the department's Linux machines. Again, read the **man** pages and ask me questions. Do not forget to refer to the lecture notes. They may not answer all the questions but should point you in the right direction.

2   Your job

Write a C program. This program will be called "wyls". It will be a version of the standard system utility "ls". The first step in this process is to create a pseudocode file, **hw5_pcode.txt**. Remember pseudocode is a plan for the program, NOT a bunch of C code. It is plain text. Yes, you can use words like *if* and *while*, but it is really plain text. Really plan, it will make things a lot easier.

Then create the C source code file (named exactly **wyls.c**), and you will create a makefile, named exactly **Makefile**. It will have a target, wyls, that is dependent on wyls.c and generates the executable, wyls.

The compiled program, "wyls" will

1. Process, <u>without</u> modifying them, its arguments via **argv** and **argc**.

2. Arguments can be pathnames (filenames, link names, or directory names), and/or options. If some argument cannot be accessed (if it is not an option), a suitable error message (just use perror()) will be printed and the program will **continue**. If some given option is not supported, a simple error message will be printed and the program will **end**.

   If one of the non-option arguments is a directory list the contents of that directory.

3. Options MUST appear before filenames. That means that

   ```
   $> wyls -n dir1 -h
   ```

   is acceptable with the caveat that "-h" is treated like a pathname. Of course you probably will not be able to open the file so you will be (using perror()) printing a message like "-h: file not found."

1

The user can "lump" the options together. This

```
$> wyls -nh dir1
```

IS entirely acceptable. Thus an combination of "-n -h," "-h -n,", "-nh," or "-hn" is good. If the user gives the same option twice, the repeat is <u>quietly</u> ignored.

4. If, after any options, there are no arguments on the command line, wyls will list the contents of the ***current directory*** except for the "." and ".." directories. See the function *getcwd()*. Always ignore "." and ".." when listing ANY directory.

5. The output will be similar to the output of "ls -l", that is a long listing. A typical output of "ls -l" is

```
total 72
-rw-------  1 kbuckner faculty  156 Sep 14 10:25 class_list.txt
drwx------  4 kbuckner faculty 4096 Sep 14 20:46 hw01/
drwx------ 15 kbuckner faculty 4096 Sep 21 21:19 hw02/
drwx------ 14 kbuckner faculty 4096 Sep 28 08:44 hw03/
drwx------  3 kbuckner faculty 4096 Sep 30 11:27 hw04/
drwx------  2 kbuckner faculty 4096 Sep 30 11:28 hw05/
drwx------  2 kbuckner faculty 4096 Aug 28  2014 module01/
drwx------  2 kbuckner faculty 4096 Sep 18 13:24 module2/
-rw-------  1 kbuckner faculty  854 Sep 18 13:29 notes
drwx------  3 kbuckner faculty 4096 Jun  2 08:48 old_hw03/
drwx------  4 kbuckner faculty 4096 Jun  2 08:50 old_hw04/
-rw-r--r--. 1 root root 4953680 May  2  2014 /usr/share/dict/linux.words
```

This output WILL BE NEATLY FORMATTED!!! There will be deductions for formatting. Do not use tab characters to format the output, too much space. Make sure that you align the columns reasonably. Do not worry about the file name being too long, that just happens. But the rest really should be aligned neatly. Read the printf(3) man page about this kind of thing.

6. Your version will NOT print the <u>first line</u>, that is the total number of blocks on disk used by all files in the directories listed. And we do not care.

Nor will it print the **second column**, the number of hard links for the individual entry.

7. You will print the permissions in the "rwx" format. The first character is a **d** if the listing is a directory, a **l** if a link, or a **-** otherwise.

8. The size will be in bytes unless the **-h** option is given. Then it will be "human readable", that is a floating point number with at most 1 digit to the right of the decimal (0's to the right are not WANTED) and undecorated for bytes or followed by a **K** for kilobytes, **M** for megabytes, or **G** for gigabytes. For example the first entry

in the sample would be 156, the second would be 4K (not 4.0K), and the last 4.8M. Remember, powers of 2, not powers of 10. Do not print anything else to indicate units of measure.

9. The time will the **mtime**, the time last modified. You have do some work to get this to print out like the "ls" utility would. You will need to use the *localtime()* and the *strftime()* functions. And, make provisions for printing the year. If the file was modified within the last 180 days, do NOT print the year, but if it was modified <u>before</u> that, print the year along with the Month and Day, but omit the time. You can easily test this by using *touch* and setting the modification date to last year, or 1944, or ...

10. If wyls is called with the **-n** option, the program will print the <u>uid</u> and <u>gid</u> **instead of** the username and group name. The default is the "real" username and group name. You need to check out the *getpwnam()* and *getgrnam()* functions.

3   What to turn in

You will upload a tar archive containing three files, **wyls.c**, **hw5_pcode.txt**, and **Makefile**. If you think you need to create some header and additional files that have your own functions in them, that is fine, just name them appropriately. Do NOT tar the entire directory, only those files you are submitting. Do not use zip, bzip2, rar, or anything besides plain vanilla tar. If you compress the archive (not necessary) make sure that you use gzip. The only two extensions that will be allowed are **.tar** and **.tgz**.

After unpacking the archive in some directory, I will be able to execute *make* in that directory and *make* will:

1. generate only the executable **wyls** and

2. **not** generate **any** warnings or errors.

3. The **Makefile** will have a "clean" target that will remove the executable. There should not be any other targets beside the one for the wyls program. There will be variables used for CFLAGS and CC. The compiler WILL be "gcc". The flags will at a minimum use "-ggdb" and "-Wall". Remember to compile and test in the deparment machines. Do not slavishly copy things. If you do not NEED something, leave it out of the Makefile. You will use the special ".PHONY" target that will at least have clean in its list. "wyls" is NOT phony.