

Computer Project #4

Assignment Overview

This assignment focuses on system-level programming in a Linux environment. You will design and implement the C++ program which displays information about selected files, as described below.

It is worth 40 points (4% of course grade) and must be completed no later than 11:59 PM on Thursday, 2/11.

Assignment Deliverables

The deliverables for this assignment are the following files:

`proj04.makefile` – the makefile which produces `proj04`
`proj04.student.cpp` – the source code file for your solution

Be sure to use the specified file names and submit them for grading via the CSE Handin system before the project deadline.

Assignment Specifications

1. The purpose of the program is to allow the user to search for and display information about selected files. For each file name which is supplied by the user, the program will examine each directory in a designated set of directories for a matching file name and display information about that file.

2. The user will interact with the program using command line arguments, which will be processed from left to right.

The valid program options are "-L" and "-S" (defined below). If an argument does not begin with the character '-', it will be processed as a file name (which may be an absolute pathname or a relative pathname).

3. Options "-L" and "-S" control the appearance of the program output. If "-L" is selected, the program will display the following information about each file name listed by the user:

- a) absolute pathname of the file
- b) permissions for the file
- c) size of the file (in bytes)
- d) numeric ID of the file's owner
- e) numeric ID of the file's group
- f) time of last read from the file
- g) time of last write to the file
- h) time of last change to the file's status

If "-S" is selected, only items (a) through (c) will be displayed.

Options "-L" and "-S" may be selected multiple times on the command line and will be in effect from the point at which they are selected until the next selection of one of those options. By default, option "-S" is active.

The permissions for a given file will be displayed in a format similar to that used by "ls" with the "-l" option: three characters (from the set "rwx-") for each of the three levels of access (owner, group and world). For example, "rwxr-x--x".

The three time stamps for a given file will be displayed in a format similar to that used by "ls" with the "-l" option. For example, "Sun Jan 31 12:27:29 2021".

The information for a particular file does not need to be displayed on a single line (it may appear on multiple lines).

4. The set of directories which will be searched for relative pathnames depends on the "DIRLIST" environment variable, which is a list of directories, separated by colons. For example, a list of four directories:

```
/user/cse325/General:/user/cse325/Projects:./usr/include
```

If the "DIRLIST" environment variable exists, the program will decompose it into individual directories and search each of those directories for the requested relative pathnames. If that environment variable does not exist, the program will use the current directory as the only directory to be searched.

No search is required for absolute pathnames, since an absolute pathname is already the complete name of the file.

5. The program will use the following function to access information about the files:

```
int stat( const char *pathname, struct stat *buf );
```

6. The program will include appropriate logic to handle exceptional cases and errors.

Assignment Notes

1. As stated above, your source code file will be named "proj04.student.cpp" and you must use "g++" to translate your source code file in the CSE Linux environment.

2. The "DIRLIST" environment variable is not standard, so you will need to create it at the command line before accessing it in your program. For example, the following shell command could be used to create a list of two directories:

```
setenv DIRLIST /user/cse325/Labs:/user/cse325/Projects
```

Functions such as "string::find()" and "string::substr()" might be useful to decompose the list of directories.

A suggestion: make sure that you do not alter the program's environment variables while processing them (if necessary, make a copy of the character string derived from "DIRLIST" and alter the copy).

3. The following is a valid execution of the program:

```
proj04 file1 -L file2 file3 -S file4 -L /user/cse325/file5
```

Assuming that the "DIRLIST" environment variable exists when the program executes, the program will search each of the directories in "DIRLIST" for the file name "file1", then it will search each of those directories for the file name "file2", and so on. Based on the pattern of "-S" and "-L" options, all matches on "file1" and "file4" will have a short display, and all matches on the other file names will have a long display.

Note that a particular file name which is a relative pathname may appear in more than one directory.

4. Information about the system call you will use for the project is available in section 2 of the "man" utility:

```
man 2 stat
```

Additional functions which you might consider using:

```
man 3 getenv  
man 3 ctime
```