

CSC 209 Assignment 2, Fall 2011: C programming

Due by the end of Monday October 31, 2011; no late assignments without written explanation.

Part 1: fgrep

Write a standard unix tool which is a simplified *grep*: It will search for fixed strings (using `strstr()`), not regular expressions. So we will call it “myfgrep”.

This tool takes file names on the command line or it reads the standard input, in the standard unix tool manner, and produces its output on the standard output.

It takes two command-line options. ‘-q’ (“quiet”) supresses all non-error output (so the result is only in the exit status, e.g. for use in an *sh* ‘if’ statement); it takes no argument. ‘-m’ takes an argument which is the maximum number of lines to output (total between all files) (thus when the output limit is reached, it becomes like the ‘-q’ situation).

The `-q` and `-m` flags must work in either order and in any standard combination. Use `getopt()` to process the command-line options correctly.

The exit status is 0 if at least one line was matched; 1 if no lines were matched; and 2 for usage or file-opening errors.

Part 2: ls -l

Write a program named “`ls1`” which does an “`ls -l`”, except that it will not sort. It takes no options on the command-line, only file names. If there are no arguments, it lists “.”.

The first character of each line will be ‘d’ for directories, ‘l’ for symbolic links, and ‘-’ for regular files. If a file is in none of these categories, print ‘?’.

To turn the numeric uid into the text logname, use `getpwuid()`. Similarly, for the gid use `getgrgid()`.

For the “`rwX`” stuff, you may need to review the use of bitwise (logical) operators.

The GNU *ls* has a fancy algorithm for choosing output field widths, but your program will use fixed field widths. Also, do not worry about the decision as to whether to print the year or the time—always print the time.

Part 3: Manual page for myfgrep

For this section of the assignment you will produce a file named `myfgrep.1` which contains a manual page for the `myfgrep` program, ready for run-off with the command “`nroff -man myfgrep.1`”.

Use the standard section headings: NAME, SYNOPSIS, DESCRIPTION, FILES, SEE ALSO, DIAGNOSTICS, HISTORY, and BUGS. If a heading does not apply, or a section would provide only obvious information, do not include it; except that NAME, SYNOPSIS, and DESCRIPTION are mandatory.

A good introduction to man-page-writing is Henry Spencer’s document “How to write a manual page”, which you can find at

<http://www.dgp.toronto.edu/~ajr/209/manpage.pdf>

This document also describes proper use of the “`nroff`” formatting language in which man pages are written.

(over)

Your man page will be graded on stylistic matters as well as content, so remember the elements that make good technical writing (brevity, objectivity, etc). Keep in mind that man pages are meant to act as reference pages and not as instruction manuals for beginners. Other similar pointers can be found in the above-cited document, and also in

`http://www.dgp.toronto.edu/~ajr/209/manpage-ajr.html`

Note that many of the GNU man pages violate standard man page practice by using a boilerplate template in which every man page contains text about copyright, etc, which users ignore. Avoid any mechanical or useless text. (This is also a life recommendation, but that's beyond the scope of CSC 209.)

Other notes

Your C programs must be in standard C. They must compile on the CDF machines with “gcc -Wall” with no errors or warning messages, and may not use linux-specific or GNU-specific features.

Pay attention to process exit statuses. Your programs must return exit status 0 or 1 as appropriate (or 2 in the case of myfgrep).

Call your assignment files myfgrep.c, lsl.c, and myfgrep.1. Your files must have the correct names to be processed correctly by the grading software, and auxiliary files are not permitted.

Once you are satisfied with your files, you can submit them for grading with the command

```
submit -c csc209h -a a2 myfgrep.c lsl.c myfgrep.1
```

You may still change your files and resubmit them any time up to the due time. You can check that your assignment has been submitted with the command

```
submit -l -c csc209h -a a2
```

You can submit files individually instead of all at once, and you can resubmit a particular file by using the -f option, e.g.

```
submit -c csc209h -a a2 -f lsl.c
```

This is the only submission method; you do not turn in any paper.

Please see the assignment Q&A web page at

```
http://www.dgp.toronto.edu/~ajr/209/a2/qna.html
```

for other reminders, and answers to common questions.

Remember:

This assignment is due at the end of Monday, October 31, by midnight. Late assignments are not ordinarily accepted, and *always* require a written explanation. If you are not finished your assignment by the submission deadline, you should just submit what you have, for partial marks.

Despite the above, I'd like to be clear that if there *is* a legitimate reason for lateness, please do submit your assignment late and send me that written explanation.

And above all: Do not commit an academic offence. Your work must be your own. Do not look at other students' assignments, and do not show your assignment (complete or partial) to other students. Collaborate with other students only on material which is not being submitted for course credit.