F.A.Q. for CS162 at Oregon State University

Kevin Bergman

August 1, 2014

Contents

1 Intro

It should be noted that your instructor may make some of these requirements and failing to use them could mean point deduction. For instance, if your code is unreadable without any comments, or if you have an excessive amount of files without a makefile. Just dont make grading hard for us.

2 Web Portals

2.1 TEACH

This is a portal to a server on campus where we collect your submissions. Each time you submit to TEACH, a new folder is created with your username in a corresponding parent folder. If you submit to TEACH more than once, the folders on our end get renamed as so:

username/	Most recent submission.
username.old.1/	First submission.
username.old.2/	Second submission, etc.

Keep in mind: Its a general principle to always include all of your work every time you submit. Dont split up the files in the submissions. For isntance if your professor asks you to submit a writeup before the actual assignment, you'll want to first submit the writeup, and then include it again once you submit the code. Unless otherwise specified, theres a chance we will look over your .old folders.

2.2 Piazza

Discussion board, easy to use and collaborate on.

2.3 Blackboard

eCampus interface. It has your grades, holds a lot of course material, and you take tests and quizzes here. Theres also a discussion forum that is sometimes used.

2.4 Canvas

3 Submission

3.1 Source Files

3.1.1 .cpp

These are the C++ source files you will be working with. Each program submission will (likely) have at least one of these, if not many.

3.1.2 .h / .hpp

These are header files, which is a supporting file to the source files. You don't nessesarily need these in your assignments (unless the instructor asks for them), but they can be benificial. For isntance they can make code and project more readbale, and are essential in good program design. The biggest difference between .h and .hpp files is that the .hpp file extension should indicate to the user that the file is not interchangeable with a C program, whereas .h could be.

3.2 Writeup

This is a PDF document you will include with all (most) submissions. These are the generally required sections, but will vary depending on your instructor or the particular assignment:

Requirements:	Talk about the assignment requirements.
Assumptions:	Is there something thats not clear?
Design:	How do you envision your program?
Testing:	Test valid and invalid values. What is observed?
Reflection:	Discuss your project in retrospect.

3.2.1 Design

These are usually two of the most important sections of the writeup. The design is suppost to reflect your conception of the program.

3.2.2 Testing

Testing is support to reflect the idea that you've used your program and understand what happens when you use it with interesting values. Generally this section is just a table with at minimum three columns: Input, Expected Output, Actual Output. Then the rows consist of various input you've given your program.

You'll want to test obvious cases (1, 5, 8), boundary cases (0, 10, 100), and extreme cases (999999, negative numbers). The previous numbers are arbitrary and depend on your program, but hopefully you get the idea. You then record this data under the testing section of the writeup.

3.3 Makefiles

So you know how to compile a program, awesome. But what happens if the file list and compilation options grow long and confusing? Must be a pain to write that compiler string each time. Even with tab completion its a pain. So lets talk about makefiles. You can think of them as an engine to start your engine. The files have a special language with semantics meant for build control. Please include a makefile.

There should be two makefiles included somewhere with this document, a simple one that's easy to udnerstand, and a more robust one that's more reusable.

3.4 Archive

Please use the tar compression tool on your items before you submit. This offers timestamping and an archive of your work. It also familiarizes you with a popular tool. Theres an option for doing this automatically in the robust makefile mentioned above.

Compress tar cvjf cs162_assignment2_username.tar.bz2 file1.cpp file2.cpp file3.cpp

Decompress tar xvjf cs162_assignment2_username.tar.bz2

cvjf are options: c / x: (create / xtract) archive v: verbose, j: bzip2 compression, f: use the following file.

cs162_assignment2_username.tar.bz2 is the name of the compressed file were going to create. file1.cpp, file2.cpp, file3.cpp are the files that are going to be archived.

Also acceptable: - Zip

Not acceptable: - submitting files as is (no compression) - compressing with .rar.

4 Style Guide

Code should look nice. It should have logical and consistent Indentation. There should be descriptive yet concise comments describing the functions. See style guide document.

5 Basic Shell Commands

pwd	Prints the current working directory.
ls	Lists files in current directory.

ls -a	Includes self and parent folders, and hidden files
ls -l	Shows permissions, timestamps, groups, etc.
cd /folder1/folder2/folder3/	change directory to nested folder3.
	(absolute path, / is your home folder)
cd ../../folder2/folder3/	change directory to nested folder3
	(relative path, ../ is a link to the parent folder)
rm file1 file2.c file3.cpp	removes these files.
rm -r folder1/	removes folder and everything in it.
mv file1.cpp ../folder2/file1.cpp	moves a file to another directory.
mv file1.cpp file2.cpp	keep files in same directory but renames it.
cp file1.cpp file2.cpp	makes a copy of file1 and names it file2.
man man	Gives the manual page for the man command.
man cd	Manual page for cd command.
man ascii	Man page (and tables) for the ASCII character set.
top	Lists working processes in resource-heavy order.
exit	An explicit exit from a terminal (or ssh session).

6 Useful CLI Tools

ssh. This tools lets you connect your host computer to a remote terminal.

example: ssh username@flip.engr.oregonstate.edu

sftp. Its like using SSH but with easy file transferring commands. Adding l in front of the command applies the command to the host computer. example: sftp username@flip.engr.oregonstate.edu connects to the flip server. ls Lists files on remote server. lls Lists files on host server cd dir/ Changes directories on remote server. lcd dir/ Changes directories on host server. put file.cpp Transfers file from host to remote server. get file.cpp Transfers file from remote to host server.

alias. This tool lets you make custom command line commands. example: alias Lists all current aliases. alias l=ls Binds l to ls. alias flip=ssh username@flip.engr.oregonstate.edu Binds flip to ssh to flip server.

grep. Lets say you want to look for a particular bit of text, but you have several files to search through. Wouldnt be efficient to open each of them and search, would it? Grep comes in handy by scanning as many files as you like for text. example: grep text to search *

find. Like grep but used for file names.

vim/emacs. Command line text editor that comes stock on all (most) Linux distributions. I encourage you to learn one and use this throughout your courses. Benefits include increased code dexterity, more exposure to the CLI, and theyre stock on nearly every unix environment, so you can edit files on an ssh connection.

A short guide to Vim:

If you use your keyboards Delete command, it saves the deleted character into a register.

There is a file saved in your home folder that is used for your vim configurations. The file (.vimrc) can be used to customize your vim experience. Here is an example .vimrc file: ¡link to rc¿

Useful Vim commands:

:w: save file :q: quit editor :q!: quit without saving :wq: Save and quit

Movement: (same as arrow keys) h: left j: down k: up l: right 0: Move to first character of the line. Same as ¡Home¿. $: Move to the last char of the line. Same as ¡End¿. #¡direction¿: Move # units in ¡direction¿.

i (when in command mode): change to insert mode. esc (when in insert mode): change to command mode. (the following will assume we are in command mode) yy: yank (copy) the current line. dd: delete (and copy) the current line. y¡arrow¿ (where ¡arrow¿ is an up / down directional key): yank the current line and 1 line in that direction. d¡arrow¿ (where ¡arrow¿ is an up / down directional key): delete (and copy) the current line and 1 line in that direction. y#¡arrow¿ (where ¡arrow¿ is an up / down directional key and ¡#¿ is a number greater than 0): yank the current line and # lines in that direction. d#¡arrow¿ (where ¡arrow¿ is an up / down directional key and ¡#¿ is a number greater than 0): delete (and copy) the current line and # lines in that direction. p: Paste the most recent item in the save register. u: Undo. ¡ctrl¿R: Redo. v: visual mode, similar to highlighting with a mouse. Can be used with other commands like y and d. x: Deletes the character in front of the cursor. Same as using the ¡delete¿ button. ZZ: Save and quit, like :wq but can be done easily with one hand. ZQ: Quit and exit, like :q!. gg=G: This will use Vims auto-indent feature. It comes in handy if you have a particularly messy coding style. /¡string¿ (where ¡string¿ is a string of your choice): This will search your program for and highlight ¡string¿. n: Moves your cursor to the next instance of the term you searched for. N: Moves your cursor to the prev instance of the term you searched for. :%s/string1/string2/gc: This is a search and replace command where string1 is your string to replace and string2 is the string to replace it with. The c at the end of this command means to ask you permission before overwriting at each instance. Without the c it will not ask. :set number: Shows / hides line numbers. :¡number¿: Jumps to that line number. :set colorscheme ¡scheme¿: Changes text colors based on ¡scheme¿. *: Highlights and searches for whatever text your cursor is on.

7 Useful Software

FileZilla. A file transfer protocol (FTP) program. Used for transferring files to / from another computer.

PuTTy (alt: KiTTy). Secure Shell (SSH) software for Windows. It gives you a linux terminal from OSUs Flip sever. Dont forget to change keyboard settings to Alt-H so you can use backspace.

SVN / GIT Revision control software. This is beyond the scope of these classes, but theyre very useful to learn and you will need it in future classes / work.

8 Useful Websites

github.com pastebin.com stackoverflow.com cplusplus.com linuxquestions.com piazza.com blackboard.com

9 Other

man ascii