

Tutorial 1 Bash

CS 137

Fall 2014

By: Chantelle Gellert

Suggestions based on Operating Systems

Windows	Compilation	Editor	Ways to submit
Option A:	cmd / cygwin	Nano/pico/vim	Internet Browser
Option B:	gcc	Code Blocks, Dev C, visual C++	Internet Browser
Option C:	PuTTY / SSH	Working on linux.cs, See Linux	marmoset submit cmd

Mac	Compilation	Editor	Ways to submit
Option A:	Terminal	Nano/pico/vim	Internet Browser
Option B:	X11	Eclipse, Emacs, Xcode	Internet Browser
Option C:	SSH	Working on linux.cs, See Linux	marmoset submit cmd

Linux	Compilation	Editor	Ways to submit
Option A:	Terminal	Nano/pico/emacs/vim	Internet Browser
Option B:	SSH	Working on linux.cs	marmoset submit cmd
Option C:	gcc	QtCreator	Internet Browser

Note: linux.cs is the school server that you can ssh to using various terminals. The school server provides you with the gcc/g++ versions that marmoset uses. It also has various Text Editors already installed and ready for use.

Getting terminal set up

Mac, Cygwin or Linux: connecting to linux.cs (SSH)

1. Open the terminal
2. Enter the command: `ssh -Y your-username@linux.student.cs.uwaterloo.ca`
(Important: The -Y flag enables X11 forwarding and use of the X applications)
 (Your username is your quest username)
3. Enter your quest password If it does not work, follow this link to reset your password:
<http://www.student.cs.uwaterloo.ca/password>

PuTTY: connecting to linux.cs (SSH)

1. Open PuTTY
2. In the Host Name field enter "linux.student.cs.uwaterloo.ca"
3. In the sidebar under SSH, click X11
4. Click the box that says "Enable X11 forwarding" This will allow you to use X Applications
5. Press "Open"
6. Enter your Quest username and password. (It does not look like anything is happening when you type your password, but it is still retrieving the characters)

Optional: Install Xming

- This allows the use of X applications like xpdf which allows you to view pdfs
- To view a pdf, type the command "xpdf filename"

Basic Terminal Commands

Command	Followed by	Description
cd	directory name	changes current directory
cd	-c	returns you to the previous current directory
ls		views (non hidden) files within the current directory
ls	-a	view list of all in current directory (including hidden, ., and .. files)
ls	-l	view list of files in long form
ls	-al	view list of all non-hidden files in long form
mkdir	directory name	make a new folder (directory)
rm	filename	removes file
rm	-r	recursive (rm -r folder) used to remove folders
..	/filename	look at one directory back from current spot
cd	..	go back from current spot to one directory back
head	filename	outputs first lines of a given file
tail	filename	outputs last ten lines of a given file
head	-5	outputs the first 5 lines of a given file
tail	-4	outputs the last 4 lines of a given file
wc -l	filename	word count number of lines
wc -c	filename	word count number of characters
wc -w	filename	word count number of words

Text Editors

Three main options:

Pico/nano

- Very simple
- Cannot accomplish as much or as quickly as in vim or emacs

Emacs

- Very steep learning curve
- If you learn it well, can accomplish tasks very quickly

Vim

- Some learning curve
- Can accomplish tasks quickly
- Enter the command `vim file.txt` to create or start editing a file
- By default you are in command mode
- Keystrokes will activate a command
 - i - start inserting text (enter insertion mode)
 - Esc - escape back to normal mode (if in insertion mode)
 - :w file.txt - save the file to give filename(write) By default saves to the name specified when you open the file
 - :wq - write and quit
 - :q! - quit
 - h,j,k,l - navigate
 - x - delete the highlighted character
 - r followed by another character - replace the highlighted character with the one specified.
 - o - open a line below the current line
 - O - open a line above the current line
- Use vim tutor to learn everything yourself (enter the command `vimtutor` on the command line (not in vim))

Compiling

After ssh into the schools computer. See Lab0

```
vim filename.c //enter code into file (nano, emacs ect. is fine)
g++ filename.c -o spot //-o filename lets you change g++ outfile filename from a
    to anything you want like spot
./spot //runs the code in filename that has been changed to the name spot
./spot < input.txt //send input from a file to the program spot
./spot < input.txt > output.txt // sends input from a file and outputs to a file
```

Submitting to marmoset through terminal

Go to:

https://www.student.cs.uwaterloo.ca/~cs241/marmoset_submit
