How to compile C++ programs on Windows

We have available a few programs, written in C++, that are useful to students in our upper division physics lab classes. While most of these programs are available as executable programs (binaries), all of them are available as source code packages.

These source code packages, as their name implies, contain the source code for some program, and this code, which is often written on a Unix-type (e.g., Linux) machine, is designed to be available for compiling and executing (running) on a variety of machines, including, for example, your Windows 7 or your Mac OS X machine.

The notes that follow describe how to compile and run a typical source code package on a Windows 7 machine, including, for example, your laptop, if it happens to have a Windows 7 operating system (OS) installed on it.[1]

A typical source code package consists of a compressed *tar file* (often called a *tarball*), whose name will look something like `someprogram-L.M.N.tar.gz`, where "`someprogram`" is the name of the program, and "`L.M.N`" is the version number. The usual Windows 7 machine will not know what to do with a *tarball*, so our first steps will involve the installation of some free software to enable your machine to recognize and open a compressed *tar file*, and then to compile the source code therein so as to create an executable program.

This software will enable your Windows machine to behave in several respects like a Linux machine, with its accompanying set of Linux-style commands. The software we describe is called **MinGW** (for "Minimalist Gnu for Windows"). Its main website is http://www.mingw.org/.[2]

Along with **MinGW** is also **MSYS**, which supplies additional needed packages.

These notes also include detailed descriptions of a few tools, including the useful terminal emulator called **mintty** (supplied by MSYS) and an excellent text editor called **vim**. **mintty** smoothly facilitates linux-like features (some of which we describe) on your Windows system. **vim**, which is widely used and available not only for Linux but also for Windows and Mac operating systems, is much admired, and worth learning. Get it into your kinesthetic memory. It is much better than the usual "what you see is what you get" (WYSIWYG) text editors.

Step by step instructions

1. **Use Firefox:** If you are accustomed to using **Internet Explorer** as your browser, our advice is not to use it. Use **Firefox** instead, and specify it as your default browser. It has useful add-ons, is faster than Internet Explorer, and less loaded with garbage. Hence the first thing to do, if you have not done it already, is to install the Firefox browser on your Windows 7 machine. It is available from https://www.mozilla.org/en-US/firefox/new/. When you have it installed, here are useful things you can do:

   (a) Make sure you can view the **Menu Bar** as well as the **Navigation Toolbar**. If neither of these is visible, you can right-click on the Firefox tab at the upper left of the browser window, and check the boxes for both of these so you can see them.

[1] The steps described in these notes worked well in February, 2015. It is hoped that they will still work by the time you read this. The maintainers of MinGW and MSYS sometimes update these packages, which can cause them to behave in unexpected ways. If something does not work for you, remember that Google is your friend, so try Googling some phrase or error message that describes the problem. Although you may find bad advice, it is often surprising to discover that others have experienced similar problems and have solved them.

[2] The URLs shown in this document should be clickable if you are reading this with a good pdf reader.

(b) The most useful add-on extension is called **Tree Style Tab**. Click on Tools → Add-ons, and enter "Tree Style Tab" into the search box. If it's not in the initial list, click on "see all 2160 results" at the page bottom, and it will come up at the top of the list. Install it, and your tabs (after restarting Firefox) will be in a column at the left, instead of spread across the top. In the "Preferences" for this add-on, choose the "Metal" skin, and you will be set.

(c) Then in Edit → Preferences, choose, under "General → Startup", "Show my windows and tabs from last time", and under "Tabs" check the box that says "When I open a link in a new tab, switch to it immediately".

2. **Install MinGW:** Now download `mingw-get-setup.exe` from http://sourceforge.net/projects/mingw/files/Installer/ and save it to your Downloads folder.

3. From Windows Explorer → Downloads, right-click and Open `mingw-get-setup`, and run it. Be sure to run it as admin (for all users). This should install the **MinGW Installer** and cause the appearance of an executable shortcut icon on your Desktop, labeled "MinGW Installer".

4. Open the Installer by double-clicking on it (or right-clicking and choosing "Open"). A window should pop up.

5. Click on the top tab (**Basic Setup**) in the left panel. Several rows should appear in the right panel.

6. Choose `mingw32-base` and `mingw32-gcc-g++` by clicking on the box to the left of each, and click on **Mark for Installation** for each.

7. Now in the left panel, click on the **Installation** tab, and then on **Apply Changes**. A window should pop up, summarizing your proposed installations, so click **Apply**, which should cause the installation of the packages you checked, along with the additional packages on which they depend. This may take a while. When it completes, close the installer (Installation → Quit).

8. **Install MSYS:** Next, to enable smooth functioning of your MinGW (and MSYS) installation, click on this link: http://downloads.sourceforge.net/mingw/MSYS-1.0.11.exe to download `MSYS-1.0.11.exe`. Save it in your Downloads folder.

9. Next (again from your Downloads folder), run `MSYS-1.0.11.exe` (again as admin), following all the default settings, with **TWO EXCEPTIONS**: First, change the installation folder from `C:\msys\1.0` to `C:\MinGW\msys\1.0`. Second, set the Start Menu folder to `MinGW\msys\1.0`. (If you don't make these changes your system may not work reliably.) A "postinstall" script should subsequently come up, asking two questions (answer "y" to both, since you have already installed MinGW), followed by a third question asking for the install location of MinGW, which should be `c:/MinGW`. When it finishes, a new shortcut icon, labeled "MSYS", should appear on your Desktop. It is a shortcut to `msys.bat`, whose full pathname is `C:\MinGW\msys\1.0\msys.bat`.

10. Now, right-click on this **MSYS** icon, and choose "Open". A terminal emulator window should pop up, with a prompt showing your username, the name of your machine, and on a new line, a "$" sign, which is the end of the prompt.

11. **Install mintty:** At this prompt, give the command `mingw-get install mintty`. **mintty** is a much better terminal emulator than the default that popped up in the previous step. After

installing **mintty**, try giving the command `mintty`. A mintty terminal emulator window should pop up, which you can close for now. It would also be a good idea to install `msys-man`, with `mingw-get install msys-man` so you can read "man pages" (documentation).

12. To incorporate **mintty**, we'll make a shortcut icon for it that will be available on your desktop or in the Windows task bar, and useful for starting an **MSYS** session. Start by right-clicking on the Desktop, choosing "New" and then "Shortcut". A new window will appear (along with a fresh shortcut icon) that asks "What item would you like to create a shortcut for?". So enter this phrase: `C:\MinGW\msys\1.0\bin\mintty.exe`. Use the default "mintty" for the name, and click "Finish". This will create your new shortcut icon, labeled **mintty**.

13. Now, to be able to use this new icon to start your **MSYS** session, right-click on it, and choose "Properties". In the pop-up window, add `/bin/bash -l` (that's an "ell", not a "one") to what appears in the Target space, then click "Apply" and "OK". Now you should be able to right-click on the mintty icon, choose "Open", and a fresh mintty terminal emulator should appear, logging you in to your HOME directory. If you give the command `pwd` (present working directory) you should see `/home/<your_username>`. Note that the `mintty` window may be enlarged by dragging its edges with your mouse, or by double-clicking on its top bar. The next steps will customize your setup.

14. **Install vim:** The first step is to install **vim**, which is an excellent text editor. In your **mintty** terminal, give the command `mingw-get install msys-vim`. This should install **vim** on your system. Try giving the command `vim --version`, and to get you started, try `man vim`. (This won't work if you don't have `msys-man` installed, in which case `mingw-get install msys-man` will enable `man vim` to work.)

15. You can now use **vim** to create three useful text files:

    (a) From your home directory, give the command `vim .vimrc`, which means you're going to edit the file `.vimrc`, and in the "insert" mode (hit the "i" key to get into the insert mode), type the line `set number`. Then hit the "ESC" key to get back into the command mode, and type `:x` to save the file and exit the editor. This will cause, for the next time you invoke **vim**, line numbers to be displayed in the editor, which turns out to be very convenient, especially for larger files.

    (b) Next, give the command `vim .bash_profile`, and put the lines below in it. (Lines beginning with a sharp (#) are comments.)

```
# First, get the aliases:
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
# Next, set a couple of environment variables:
# This sets the prompt to something useful:
export PS1='{\W \!} '
# This makes the "less" command better:
export LESS='Cem'
# Prepend $HOME/bin to your PATH, so any executables you compile and
# install in $HOME/bin will be found:
export PATH=$HOME/bin:$PATH
```

(c) Finally, give the command `vim .bashrc`, and insert these lines:

```
# This allows the command "cl" to clear the screen:
alias cl='clsb'
# This allows "v" to edit a file:
alias v='vim'
# These help to prevent clobbering a file when moving or
# copying a file:
alias cp='cp -i'
alias mv='mv -i'
alias rm='rm -i'
# This is a useful abbreviation for "less":
alias m='less'
```

These last two ((b) and (c)) are shell scripts (bash scripts) that will be executed whenever you start a session from the `mintty` shortcut icon.

16. To get some practice with vim, there are some fine tutorials available. You might try starting with this one: http://www.openvim.com/tutorial.html. After you get some practice, try giving the command `vim testfile` (or `v testfile` — does the `alias v='vim'` work?) and play with it a bit. There are other (more conventional) tutorials. One of the best is available from within **vim** itself. From the command mode, type `:! vimtutor`. For more rabbit-holes to explore, just Google "vim tutorials". Many are available.

17. **Now, try compiling a program:** After you have completed the above steps, try compiling and running a C++ program. We'll illustrate the process witn a program called **meanvar**. The source code is contained in the compressed tar file called `meanvar-2.0.1.tar.gz`, where `2.0.1` is the version number as of this writing. The `.gz` extension means that the tar file is compressed using `gzip`, a common Unix compression tool. We'll assume that you have this file on the Desktop of your computer. It might have been on a flash drive, or downloaded from another machine or website, or emailed to you as an attachment. It's a small file, perhaps 100 kilobytes.

18. Bring up the `mintty` terminal from the mintty icon on your Desktop. You should be in your home directory. The command `pwd` should show `/home/<your_username>`.

19. Now give the commands `mkdir src`, then `cd src`, hitting Enter after each. This will create the `src` directory (folder) in your home directory and switch you to that `src` directory. (`cd src` stands for "change directory to `src`".) Then, at the prompt, type `cp`, then drag the `meanvar-2.0.1.tar.gz` from your Desktop to the mintty window, then type a space and a period and hit Enter. The command `ls` (for "list") should show the presence of `meanvar-2.0.1.tar.gz` in your `src` directory.

20. Next give the command `tar xzvf meanvar-2.0.1.tar.gz`. This should create the directory `meanvar-2.0.1`, and put the contents of the tar file in that directory. So now give the command `cd meanvar-2.0.1`. The command `ls` should list several items. You have extracted the contents of the tar file, ready for compiling.

21. Next give the command `./configure --prefix=$HOME`. This may take a while to complete. Setting `--prefix=$HOME` is a good idea, since it will direct the compiler (eventually) to install the executable `meanvar.exe` program in your home directory tree, *i.e.*, in `/home/<your_username>/bin`, which is a good safe place to have it end up. The `./configure` command may take a minute or two to complete, so be patient.

22. Next give the command `make`, which may take a minute or two to complete, depending on the speed of your computer.

23. Next give the command `make install`, which will install the executable `meanvar.exe` as `/home/<your_username>/bin/meanvar.exe`. To see if the `meanvar` works for you, give the command `meanvar --help`. A page of text should appear, explaining how to use the program, followed by a few lines that briefly describe its usage. The commands `meanvar test_boson`, `meanvar test_no_wts`, `meanvar -w test_wts` and `meanvar test_sim` should all work. Should you want to uninstall **meanvar**, the command `make uninstall` will uninstall the executable `meanvar.exe` from your system.

24. If you should ever want to remove `MinGW` and `MSYS` from your computer, the easiest way is simply to delete the folder `C:\MinGW`. This may be done from Windows Explorer under Computer → Local Disk(C:).

25. Here's one final bit of advice:

> **Avoid the use of spaces in names—names of files, names of folders or directories, names of accounts and names of machines. Spaces often confuse the Linux-type commands used in the MinGW software. To separate words, try using underscores or hyphens.**

There are many steps in all of the above. If any of them don't work for you—they are likely to become out-of-date, or may be confusing, or there may be steps we've missed—send me an email. We would like to keep it all working.

– Peter Scott, February, 2015
`drip@ucsc.edu`