University of Washington
Computer Science & Engineering

CSE 303: Concepts and Tools for Software Development, Winter 2008

CSE 303 Homework 7A
Due: Saturday, 2/24/08, 11:59PM
Turnin: Described Below

FAQ

1 How do I print this assignment?
Print it like any other web page, but selecting landscape mode. An example is here.
1 HW7A Q&A Wiki Page

Overview

This first step in HW7 does two things:

1. Provides some experience with the typical install procedure for (Unix-y) open source projects.
2. Installs the software we need for HW7B, since it isn't on any lab machines, nor in typical installs of home machines.

The first point is the more important of the two. There is a lot of open source software available. It's typically either easy to install or nearly impossible - which becomes clear with only minimal effort. Having done it once it should be easy to try out more whatever you feel like in the future.

You should do this part of the assignment alone. The larger portion (HW7B) can be done alone or in pairs.

The Seven Steps of HW7A

Everyone should go through this procedure using a department Linux box (i.e., `attu` or one of the Linux workstations). Completing the install *is* turnin -- we'll try to use your installed library at the due date.

This procedure can be done by remote connection to `attu`. (In fact, the entire assignment can be done by remote connection. The downside, which is quite large, is that you have to have the sound files on your local machine to listen to them. All code you'll use or write can work running remotely on `attu`. If you want to hear the output, though, you'll have to `scp` the sound files to your local machine.

1. **Find your cse303 group**
The easy way to figure this out is to invoke the `groups` command on `attu`. Look for a group `cse303XX`, where the XX is one or two lower case letters. That's your group.

That procedure can fail, if you're a member of too many groups. If a dozen or so groups are printed but there's no cse303XX group, go to this page and do

a Query By Username. There should be a cse303XX group shown. To make use of it, follow the man page links on this page of documentation.

2. **Fetch the SOX distribution**
   Navigate to the Sox home page, http://sox.sourceforge.net/. Look down the page for Release Information. Fetch the "latest source code," which is called `sox-14.0.1.tar.gz` as I'm writing this page.

   Get the tar.gz file into directory `attu:/projects/instr/08wi/cse303/hw7/cse303XX`, where cse303XX means your group. Depending on how you're doing things, you might be able to instruct the browser to drop it there, or you can have the browser save it anywhere and then copy or move it there using the usual methods.

   This step tests that you have access to your group's (i.e., your) directory in the shared file space.

3. **Run configure**
   Untar the Sox distribution. It will create directory `sox-14.0.1`.
   `cd` into that directory. Issue this command:
   ```
   ./configure --prefix=/projects/instr/08wi/cse303/hw7/cse303XX/usr
   ```
   This should run for a while. It's looking at what software is loaded on the machine you're using, so that it can configure Sox itself to be built on your system. When done, it prints a summary with a lot of 'yes' and 'no's about the features it did/didn't find, and a line about doing make and make install..

4. **Build Sox**
   ```
   $ make -j 2
   ```
   The "-j 2" means "do two compiles at once." I'm guessing 2 is appropriate for `attu`, which is a shared machine with 4 processors. If you're not sharing a machine, the typical value is 1.5 to 2.0 times the number of processors (or cores).

   This takes a while.

5. **Install**
   Execute **`make install`**. This should create a `usr` directory in your `/projects/instr/.../cse303XX` directory. In `usr` will be `bin` (Sox executables), `lib` (Sox libraries), `include` (Sox .h files), and `share` (Sox man pages).

6. **Set up environment**
   Sox provides one executable we'll want, plus a dynamically loaded library, plus a few man pages. To use them, you need to:
   - add `/projects/instr/08wi/cse303/hw7/cse303XX/usr/bin` to your PATH environment variable
   - add `/projects/instr/08wi/cse303/hw7/cse303XX/usr/lib` to your LD_LIBRARY_PATH environment variable
   - add `/projects/instr/08wi/cse303/hw7/cse303XX/usr/share/man` to your MANPATH environment variable.

   You can/should do this by editing your `~/.bashrc` file to add the commands that append to those environment variables. For example:
   ```
   export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/projects/instr/08wi/cse303/hw7/cse303XX/lib"
   ```

7. **Test**
   If your install worked, and you set up your PATH correctly, this should work:

   ```
   $ cd
   $ sox /cse/courses/cse303/08wi/hw7/dataFiles/20080216.a.wav test.wav
   ```

   (i.e., it should create test.wav in your home directory). If your MANPATH is correct, this should work:

   ```
   $ man libsox
   ```

   Other man pages that might be of use are `sox` and `soxformat`. (Look

in `/projects/instr/08wi/cse303/hw7/cse303XX/usr/share/man` for a complete list, then use man to read any you want to see.)

Setting Up Other Linux-y Machines

You should be able to set up any Linux-y machine using the procedures above, modified to choose an appropriate install path, and adjusting for the fact that you won't be able to directly access `attu:/...` files. Also, you must execute `make install` as root.

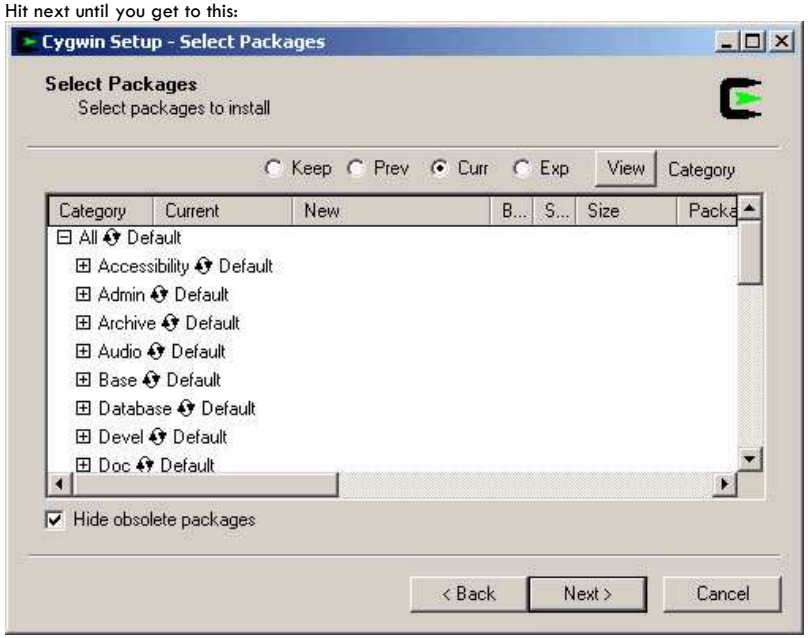
If you can act as an adminstrator (root), then a typical choice for the install path is `/usr/local`.

Special Help For Home Windows Machines

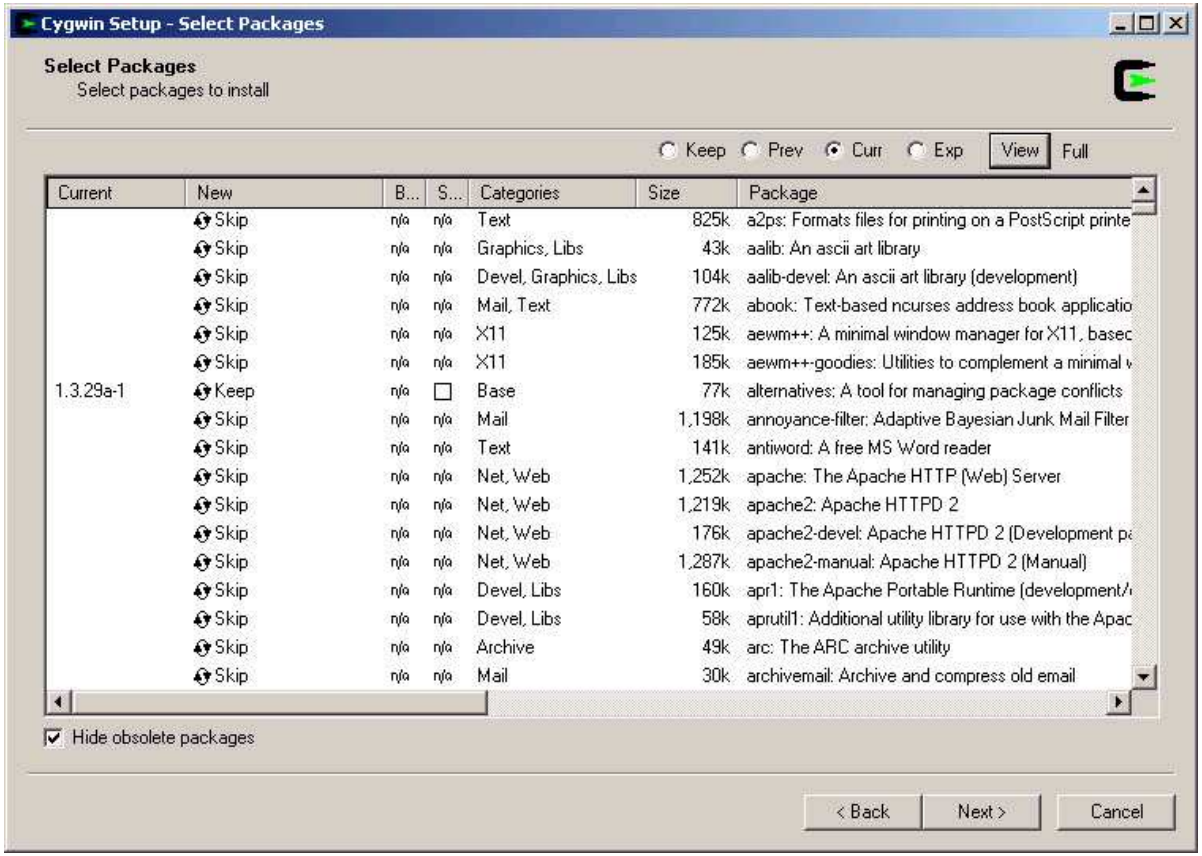
It is possible to do this entire assignment on a windows box. If that's what you have, and like to use, it can be handy to work this way, because you can listen to the sound files your code produces without having to fetch them from some remote machine. The downside is that, unless you install and run Reflection, you won't have an X server, so the interface to some programs might be a little cruder than you're accusomted to.

To do this, you'll need to use `cygwin`:

1. Go to www.cygwin.com. Click on "Install or Update Now." That will download a `setup.exe` file. You can save it anywhere, but my advice is to put it in `C:\\cygwin`.
1. Execute the setup file.
1. Hit next until you get to a screen that asks you pick a download site. You can pick any, but `ftp://mirrors.kernel.org` seems to be responsive.
1. Hit next until you get to this:



1. Make the window bigger, and click on the View button (upper right) to see this:



1. You'll need some packages not loaded by default. This list might be unreliable (as my setup is already set up), but you'll need `autoconf`, `automake`, `cvs`, `emacs`, `gcc`, `gcc-g++`, and `gdb`. To select something for installation, click on the Skip on its line in the list.
1. You should now be able to build Sox, just as above (for users of Linux-y machines). You are basically running as root, so can install to `/usr/local`. (You don't have to actually try to login as root; you are already.)
1. If you find you're missing something you need (because you get an error to that effect during configure or build), re-rerun `setup.exe` and look for whatever it is.

Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350
(206) 543-1695 voice, (206) 543-2969 FAX
[comments to zahorjan at cs.washington.edu]