

VirtualBox VM setup

If you have not used a VM before, the idea is that you will be running an entire operating system inside a program (VirtualBox) that looks to the virtual guest operating system like real hardware. We provide a downloadable image of the guest OS with everything installed, so that you only need to do minimal setup to get started.

Note that since a VM contains a whole second operating system, the system requirements are high. **The VM will take a bit over 512MB of ram (so you will probably want to have 2GB of total ram to keep Windows or Mac OS X running smoothly), plus a bit over 2GB of hard disk space.**

Getting VirtualBox

- First, download Oracle's VirtualBox, either from the VirtualBox website at <https://www.virtualbox.org/wiki/Downloads>, or from Oracle's download page at <http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html>. Choose the version for the operating system you are running on your computer.
- Once the download completes, run the installer to install VirtualBox. You will need administrator access on your computer to do so.

Getting our VM image

- Get our linux virtual machine image.
- Unzip the file into a convenient directory. The unzipped files are about 2GB, so make sure you have enough disk space available. If you are a Windows XP user and have trouble opening the zip file, try WinZip (shareware) or 7-Zip (free, open source) instead of the built-in zip support in Windows.
- Once you have unzipped the VM, double-click on the file "Compilers.vbox". This should open the VM in VirtualBox.

Using the VM

- **The provided account is "compilers" and the password is "cool".**
- To start the VM, click the green "Start" button. This should make the VM boot.
- To shut down the VM, click on the round "Bodhi" button at the bottom left and click System. Then choose Power Off.
- We have installed what you need to do the assignments, plus a few other programs, such as vim and emacs. If you want to install other packages, you can use the Aptitude graphical package manager (under the Bodhi menu->Applications->Preferences) or the apt-get command line tool. If you are not familiar with these, there are many tutorials online that you can find through a quick Google search.
- To get a terminal, click on the terminal icon at the bottom of the screen. This should get you to the point where you can start the assignments.

This VM is based on Bodhi Linux, which is itself based on the popular linux distribution Ubuntu. We used Bodhi Linux in order to keep the download size more manageable, since a full Ubuntu install is quite large. However, since it is based on Ubuntu, most Ubuntu software packages can be installed on Bodhi Linux as well.

Other Project Resources

A list of other resources that you might find helpful:

- **C++ and UNIX:**
 - Thomas Anderson's [Quick Introduction to C++](#) may be useful if you know C but are not familiar with all of the C++ subset we use in the programming projects.
 - A more thorough guide to using the version of Gnu C++ installed on our Linux machines is [here](#).
 - Quick guide to [UNIX development tools](#) (one of the many useful documents available in Nick Parlante's ever-growing [CS library](#))
- **Help with the gdb debugger:**
 - GNU's online [gdb users guide](#)
 - Printable quick reference: [gdbref.ps](#)
 - A GDB article that Julie Zelenski wrote a few years ago for a programming journal: [GDB breakpoint tricks](#)
- **References on lex & yacc:**

Lex is the original lexical scanner developed by Lesk and Schmidt; Paxson's improved version is flex. Similarly, yacc is Johnson and Sethi's original parser; bison is the GNU-equivalent written by Corbett and Stallman. Both are designed to be upward-compatible with the original while adding extensions and improvements.

 - Original documentation by the authors of the tools themselves. These papers are quite readable and serve as an excellent introduction for familiarizing yourself with the tools.
 - [Lesk and Schmidt on lex](#)
 - [Johnson on yacc](#)
 - Man pages are available from command line, e.g., `man lex`. We've also put up browsable versions of the Solaris man pages for [lex](#), [flex](#), [yacc](#), and [bison](#).
 - GNU's online documentation (full manuals, long, but very complete)
 - [flex](#)
 - [bison](#)
 - The [lex & yacc page](#) from Combo.org.
 - An [article](#) from the Linux Journal singing the praises of lex & yacc.
- **References on JLex and Java_cup:**
 - Manual for [JLex](#), [Java CUP](#).
 - [JLex](#).
 - [Java CUP](#).
 - [Cool Trees](#) (Start with class `TreeNode`).
- **References on MIPS & SPIM:**

- A PDF version of the [SPIM Manual](#) (appendix from Hennessy & Patterson's architecture book)
- The [SPIM home page](#) (downloadable versions, more docs).

Just for fun:

- A list of [funny error messages](#) from the old MPW C compiler.
- A translator for [Latin to Perl](#) (and you thought there was no practical use for what you learned in 143!)

Installing Directly on Linux

Steps:

- Install packages (If you only intend to use the C++ version, you don't need the jdk). For Ubuntu:

```
sudo apt-get install flex bison build-essential csh openjdk-6-jdk libxaw7-dev
```

- Make the /usr/class directory:

```
sudo mkdir /usr/class
```

- Make the directory owned by you:

```
sudo chown $USER /usr/class
```

- Go to /usr/class and download the tarball:

```
cd /usr/class
```

```
wget https://s3-us-west-1.amazonaws.com/prod-edx/Compilers/Misc/student-dist.tar.gz
```

- Untar:

```
tar -xf student-dist.tar.gz
```

If you want things exactly like the VM:

- Add a symlink to your home directory:

```
ln -s /usr/class/ ~/cool
```

- Add the bin directory to your \$PATH environment variable. If you are using bash, add to your .profile (or .bash_profile, etc. depending on your configuration; note that in Ubuntu have to log out and back in for this to take effect):

```
PATH=/usr/class/bin:$PATH
```