Assignment 0: Using VirtualBox

Due September 7, 2015, at 10:00pm EST

1 Background
We will be offering a standardized development system for CIS455/555 that is based on a virtual machine image. This image contains a small Linux installation, the Eclipse IDE, and all the tools you will need for the homework assignments. Having such a standardized system has several advantages:

You do not have to worry about incompatibilities between the software we will be using and other software that may be installed on your machine; if something goes wrong, you have a 'safe' configuration to go back to - just reinstall the virtual machine image; the virtual machine image will be the 'gold standard' for grading – thus, if your solution works in the VM image, you can be sure that it will also work on the graders' machines.

The virtual machine image works with VirtualBox (www.virtualbox.org).

Of course, you are free to use a different development system (e.g., a different operating system, your favorite editor or IDE, ...). However, we will only grade solutions based on their performance in the virtual machine image, and we can only offer limited support if you decide to use your own setup.

2 Installing the VM image
Your first task is to install the VM image for this assignment. The following step-by-step guide assumes that you are using a Mac OS X, Windows (7+), or Linux (recent Ubuntu) machine.

1. Download VirtualBox 5.0.2 from https://www.virtualbox.org/wiki/Downloads. Note that if you are using Windows 10 you may need to download the test build for 5.0.3 linked on the site.
2. Download the VM image from the link provided on the Web at www.cis.upenn.edu/~cis455/assignments.html. Please follow the instructions there carefully.
3. Double-click on the CIS455.ova file you downloaded. This should trigger VirtualBox. Leave the settings as they are and choose Import.
4. Choose CIS455 and click on the button in the toolbar.
5. The Linux in the VM image should boot now and automatically log select the user “cis455.”
6. Whenever a password is requested, “cis455” should work.
You should now have a running Virtual Machine with Ubuntu 14.04.

**Setting up a Shared Folder**
The next step is to ensure that you can share files between your “base” operating system and your virtual machine.

1. Go into the **Devices** menu and select **Shared Folders Settings**. Click on the little folder with a plus icon:

![Add Share](image)

and select your home or Documents directory as the **Folder Path**. Then put “shared” as the **Folder Name** and check **Auto-mount** and **Make Permanent**.

2. Your folder will now be found at the path `/media/sf_shared`.

You can create additional shared folders as you like. If you make them Auto-mount they will show up under `/media/sf_{whatever}`.

**Setting up Network Port Forwarding**
You can actually set VirtualBox to automatically forward requests from a port on your host machine, to a port on your virtual machine.

1. Go to **Devices** | **Network** | **Port Forwarding**.
2. You will likely see an existing rule from Host Port 8080 to Guest Port 8080. If not, hit the **Ins** key (or click on the insert icon on the side of the dialog box) and fill in:
Care and Feeding of Your Virtual Machine

A few tips regarding the virtual machine:

- **Always shut down the VM properly** before closing VirtualBox (by clicking on the power switch icon in the upper right corner and selecting 'Shut down...').
  
  **If you do not do this, the VM image can become corrupted, just like your operating system can become corrupted if you switch off your computer before shutting it down properly.**

- If the screen size of your VM is tiny, try resizing it or putting it into full-screen mode by hitting `right-Control-key F`. This is a toggle; `right-Control-key F` will make it windowed again, e.g., if you want to see the VirtualBox menus.

- The data you store inside the VM is persistent (i.e., will survive reboots), and we will be using a version control system. Nevertheless, we **strongly recommend that you make occasional backups**, e.g., by copying your data files to a place in your `/media/sf_shared` folder.

- You can edit text files with Eclipse, gedit, or nano. To install another editor, you must use `apt-get` to install it. For instance, to install `vim`, run `sudo apt-get install vim` in a terminal window.

If the VM runs slowly on your machine, try increasing its memory size to 1.5GB or 2GB (under Virtual Machine / Virtual Machine Settings..., or under "Edit virtual machine settings").

Developing Outside the VM, Testing within the VM

We advise you to test your code **within** the virtual machine as much as possible. This will result in a more secure (i.e., less hackable, less crash prone) setup for your development. For advanced users: If you would like to do development outside the VM, you may install Eclipse and ant. You should set the Eclipse Workspace to a directory that is shared between the VM and your host OS.

3 The Hello Web application [10 points]

Your third task in this assignment is to write a simple server program.

Create ssh Credentials

First we need to create a key pair for logging into your resources from the VM.

1. Open the Terminal, then `cd ~/.ssh`
2. Type `ssh-keygen` and hit `[Enter]` to accept the default filename.
3. Type in `cis455` as the passphrase (it’s never a good idea to leave passphrases empty)
4. Now launch Eclipse. Go to Window|Preferences. Type “ssh” into the area where it says “type filter text”. Select General|Network Connection|SSH2. Make sure Private keys says “id_dsa,id_rsa,checkout_id”.
5. From the terminal, type “`cat ~/.ssh/id_rsa.pub`”, then select and copy what it shows (starting with “ssh-rsa” and ending with “cis455@adminuser-VirtualBox”). You'll need this for Bitbucket setup.
Create Your Own Git Repository on Bitbucket

For your own code, you’ll want to make sure you have version control set up. A version control system lets you “check out” code and “commit” different versions. It saves the changes in each successive revision. You can “revert” to old versions (important if you go down a bad path!), compute diffs across revisions, and share code. Today, the most popular version control systems are git and subversion (svn), with Mercurial another fairly popular choice. We will use git on Bitbucket.

- Go to https://bitbucket.org/account/signup/ and sign up with an account.
- Follow the Tutorial to create a sample repository so you’ll have a nice program to pull (fetch from the server), commit (checkpoint your code locally) and push (propagate your updates to the server).
- Go to Eclipse and choose Import | Projects from Git. Choose “Clone URI” and make the URI: git@bitbucket.org:upenn-cis455/testserver.git
- Set the Protocol to ssh. Then hit Next. Say yes if there is a question about whether you want to continue connecting. If there is a question about passphrase, use “cis455.” Hit Next. Change the Destination Directory from /home/cis455/git/testserver to /home/cis455/workspace/testserver. Hit Next a couple of times, then Finish.

You now have a copy of the TestServer for your own work. Now as you make changes to your code, remember to periodically (ideally, at stable points) right-click on the project, and choose Team | Commit. You’ll need to enter a description (for future reference) of what you are committing. This will commit a “checkpoint” that you can always go back to. Note that by default Git commits go only to the local repository sitting in the same file system. You might want to add a remote repository in Bitbucket and “push” your commits to that. (For obvious reasons, you should not be able to push commits to the upenn-cis455 repo.)

The Actual Work

The first assignment is a VERY SIMPLE Web server that makes use of some built-in Sun libraries. (You’ll be replacing these yourself in HW1.) This assignment is basically designed to just validate that your machine configuration is OK, so it should be super-easy (by CIS 455/555 standards, at least).

- Open edu.upenn.cis.cis455/TestSimpleServlet.java and change the variable response (in the handle method of MyHandler) to the string corresponding to your PennKey / userid.
- Now go to the toolbar at the top of the Eclipse window, choose Run As, and choose Java Application.
- Next, open your Web browser on your main operating system, not your VM (!) and go to localhost:8080/test. If all goes well you should see a string there. Capture this somewhere because you’ll need it for Submitting your Results (below).
If it failed, go back to the instructions on Network Port Forwarding.

**Submitting Your Results**
Go to the following link:
http://goo.gl/forms/anfUFLhBHW

and enter your SEAS user ID and information about the output. Answer the brief questions in the form. That’s all until HW1!