ICS 312, Spring 2014: Getting NASM to run

A lot of the assignments in the course require that you use NASM on a Linux platform. This "homework assignment" guides you through the steps you need to follow to set up your environment correctly.

Disclaimer: It seems it is possible to run all software on Windows or MaxOSX for this course, but I haven't tried. If you can do it, more power to you (but you're on your own). In everything below we assume a Linux environment. *However, note that gaining some experiment with Linux is invaluable for your future career.*

I. Getting access to a Linux machine

For the programming assignments you will need to have access to a Linux machine on which the NASM assembler is installed. If you already use Linux, either as your OS or as a guest OS via virtualization, then you can skip this and the next section and go directly to the Installing NASM section. Note, however, that in the course I'll assume the **32-bit Ubuntu desktop distribution**, which you can get from the UH mirror. Download the file named **ubuntu-13.10-desktop-i386.iso**. If you don't have a Linux installation then you have two options:

- **Option #1 (recommended):** You install a Linux Virtual Machine on your computer. I recommend using the VirtualBox VM manager, which is both easy to used and free (other options include VMWare and VirtualPC). I have created an Installing Ubuntu on VirtualBox page.
- **Option #2 (more difficult):** You can install a Linux partition on your machine so that it becomes dual-boot (e.g., Windows/Linux). Tons of people do this all the time, so there are many step-by-step guides on-line. A decent one for setting up a dual-boot Windows/Linux machine is on the Ubuntu Web site. Ubuntu also provides a Windows installer, which I've never tried myself but has been used successfully by many people. See this page for MacOSX/Ubuntu dual boot.

II. Using Linux (for the first time?)

Having some experience with Linux is a great asset for your professional life. Almost every (interesting) job will have you work with more than on O/S and most include some Linux development. So if you're not familiar with Linux, don't think of this as just something you have to do for this (and other) courses, but as a valuable experience.

If you need help on using Linux, there are **tons** of useful on-line tutorials (e.g., this set of tutorials.

I have put together a few very random slide on the UNIX/Linux command-line environment in case this is useful for some of you ([PDF]). I am always available for

Linux questions.

III. Installing NASM

Now that you have access to an Ubuntu installation, you need to install NASM. Ubuntu has a package manager tool with a GUI, but I typically use only the command-line (you'll find that's a theme throughout the semester) to install packages.

Open a Terminal window and then simply type the following command:

```
sudo apt-get install nasm
```

This command will prompt you for your password, and install NASM provided you're connected to the network. That's it. The `sudo` command runs commands as the super user, and when you installed Ubuntu you were placed by default in the list of users who can use `sudo`. The `apt-get` command is used to deal with everything related to package management, and in the invocation above, to install a package.

IV. Checking that NASM works

I've built a little test case, which is in fact the basic framework we'll be using throughout the semester, so that you can check that everything's working. Download the nasm_check.tar archive to your account on your Linux box, e.g., in your home directory. This can be done in many ways, for instance from a Web browser that you're running in your Ubuntu installation. The easiest way is to type in a terminal the command (on one line):

```
wget
http://navet.ics.hawaii.edu/~casanova/courses/ics312_spring14/hw0/nasm_check.ta
```

(If you get a "`wget: command not found`" error, simply install `wget` by typing the command "`sudo apt-get install wget`")

Once the archive is in your working directory, you can then type the command:

```
tar -xvf nasm_check.tar
```

which will create a directory named `nasm_check`. Go to that directory ("`cd nasm_check`") and type the command:

```
make
```

You should get some output, and the last line should be: "It works!". If not, then you should send me an e-mail with the whole output. If you get the correct output, you now have the whole desired setup for all NASM assignments.

If you get an error, it may be because you're running 64-bit Ubuntu. In this case, just install multilib:

```
sudo apt-get install gcc-multilib
```

---

henric@hawaii.edu