Guide to installing programs on Linux (contact [ss93@iu.edu](mailto:ss93@iu.edu) for questions about this tutorial)

Modules

Sometimes you may not have to install the program you are looking for, because it already is installed on the cluster! Modules make certain programs available to you without having to find/install them. NCGAS maintains a long list of these programs, which can be found at https://ncgas.org/software-list.php.

View module list:

module av

Load a module (for example "bio" which contains useful bioinformatics programs like trinity):

module load <module name>

Unload a module (if you for some reason need to)

module unload <module name>

List the modules you have loaded

module list

Determine if a module needs another module loaded first (dependency) and to get more information on the module in general.

module display <module name>

If you look at the display, you will often see a line similar to:

append PATH <some location>

The next section covers what this PATH variable is, and why it is important when you are installing software yourself.

PATH and .bashrc

When you use the following command:

nano text.txt

You are using a program (nano), but you don’t have to tell the computer where the nano program is. This phenomenon is the result of a very important shell variable $PATH.

echo $PATH

As you can see, /bin is in there. All the folders are separated by a :. It is important to note that bash takes the FIRST instance of a program it comes across in the order of the folders in the PATH variable. If you have two copies of a program, it will run only the first it comes across.

Because of this order rigmarole, sometimes we want to know where a program is, to make sure it’s the correct one.

which less

This command tells you from where less is being executed.

If you want to run a program from anywhere without having to type in the full path to the program every time (gets very tedious), you can add the folder it lives in to my PATH variable, or more conveniently, put it in a folder that is in my path and holds all my programs. This is the /bin. However, you don’t have permission to add things to /bin (that requires root), so you can make your own.

cd $HOME

mkdir local

cd local

mkdir bin

You now have a local bin. ~/local/bin is a very common folder for installations, and some programs will look for it when installing. No let’s add it to the path, so Unix can find programs we install into this directory (which is generally everything).

declare PATH=$PATH:<NEW DIRECTORY>

By saying PATH=$PATH:newstuff, we’re appending the new stuff to what is already stored in PATH. Remember that : separates directories in your path. You can add several at a time if you’d like:

declare PATH=$PATH:<NEW DIRECTORY>:<NEW DIRECTORY>: (etc)

However, if you sign out, this variable is lost. Variable definitions are cleared every time you leave bash. How do you add these permanently? By using that .bashrc file I mentioned the above. It is an administrative file, with our profile information in it.

Modifying .bashrc

First, let’s get to the file:

cd ~

nano .bashrc

Find the line that says ‘export PATH=’ – if it doesn’t exist, add it! Export is a way to declare variables in the shell. Add a colon after the last directory listed, and insert the ABSOLUTE (i.e: full path from /) path to your bin directory. Save and exit.

You have to restart bash to affect change (it is read at log in). You can type:

source .bashrc

This command tells the shell to run the current .bashrc without having to log out. Now let’s check to see if it worked:

echo $PATH

You should now see your $HOME/local/bin in the PATH variable!

Installing Software

Installing software is kind of a rite of passage for bioinformatics. What is the best resource on how to install a program? The README and INSTALL files. ALWAYS READ THE README. It’s there for a reason, and it’s called Read Me for a reason. It’ll often tell you exactly how to install the software, but we’ll go through an example program, HMMER, which is a pretty standard installation. On occasion, there’s also a file called INSTALL which, not surprisingly, will tell you how to install the program.

Let's download it (google hmmer-3.1 to find):

wget <http://selab.janelia.org/software/hmmer3/3.1b1/hmmer-3.1b1-linux-intel-x86_64.tar.gz>

**NOTE: “**wget” is a program that gets files from the web. How do you tell which file? Our system is Redhat linux. No worries though, if you do it wrong, it will just tell you it won't work and you can remove the files (rm) and try again. Also, you generally want source files (these are converted into binary, which the computer then runs):



You can also download binary versions, but they are harder to customize (not human readable) and troubleshoot at times (cannot change directory locations, etc. as easily – more on that later).

Now let's decompress the file:

gzip -d hmmer-3.1b1-linux-intel-x86\_64.tar.gz

tar -xf hmmer-3.1b1-linux-intel-x86\_64.tar

Now we have a folder with a bunch of files. We are going to install this into the ~/local/bin. First let's look in the folder.

Oh look, a readme. Which tells us to go to the Install...

And it tells us the basic recipe:

./configure

make

make install # will not work! See below

What these sets do:

./configure sets up environmental variables, checks your system requirements, and well, configures things. Basically, checking your set up and program lists (in $PATH) to determine how to install things on YOUR SPECIFIC set up. It completes the Makefile to match your file locations, system settings, etc.

make follows the instructions of the Makefile and converts source code into binary for the computer to read.

make install installs the program by copying the binaries into the correct places as defined by ./configure and the Makefile.

But if you try this, you will get a very common error: you don't have permission to install things in the default location!! How many of you got this error? So what do you do?

./configure --prefix=<new location>

make

make install

The additional flag/option will tell the configure program to set up the Makefile to put everything where you would like it. Remember this command, you will need it often!!

What do you put for the location? $HOME/local. If you hit tab after the local, it will fill the full path in. Then when you make, and make install, the program will automatically end up in your ~/local/bin. You can put the program wherever you have access to, however, you would have to add each one of these directories to your PATH.

Why $HOME/local and not $HOME/local/bin? Because this is a PREFIX. It is the path that the installer adds to the beginning of all the folders it wants to create, one of which is “bin” (you may also see “share”, “lib”, “src”, etc. show up!). If you direct the program to $HOME/local/bin – it will create all these folders within your bin, meaning the program would be in $HOME/local/bin/bin. This is not only confusing to find, but it won’t be in your PATH automatically.

Installing Software from a Binary

Sometimes programs are pretty stand alone or very simple. These programs may not require customization that comes with using ./configure. Sometimes you will untar/unzip a program, and it will have the bin already there or the program already made. Lucky you! Copy the program to your $HOME/local/bin and you can use it without typing in the full path.

Others and tips

Sometimes other programs have weird installations. How do you know? README and INSTALL will tell you!

Occasionally, you will run into a program that requires “root”. You will know this when you run into a “permission denied” error. What do you do? It depends on the formula given to install the program. We discussed ./configure, make, make install above:

If ./configure, make, make install format gives you permission denied:

./configure –prefix=$HOME/local

If the program requests that you use cmake (a slightly different compiler) and gives you permission denied, you have to add a prefix flag to it as well:

cmake -DCMAKE\_INSTALL\_PREFIX=$HOME/local/

NOTE: there may be other flags listed in the README/INSTALL. Use them, but add the above option as well.

**When all else fails, contact help@ncgas.org!**