# Lecture 11: Everyday Tasks in Linux

CS2042 - UNIX Tools

October 24, 2008

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# Everyday Linux

Most of the time you spend on your computer isn't consumed by writing shell scripts. If your computer at home didn't have a mouse, how often would you use it? Would your parents even look at it?

The point is, most of the everyday tasks you perform on your computer are GUI-based. Fortunately, there is a great deal of GUI support for UNIX-based systems as well.

In fact, many of the programs we'll discuss today are available on Windows, too.

# The All-Important WWW

Web browsing is a great place to start. You can always use a text-based web browser like **links**, but the multimedia experience depends on a GUI browser.

One potential setback for Linux browsing is the lack of plugin support, such as streaming video from many commercial sources (YouTube works fine!).

Firefox is still King of the Ring when it comes to open-source web browsers. Also available on Windows!

---

Not a fan of Firefox? Most desktop environments have their own web browsers which get the job done.

# Consolidating E-Mail

If you have multiple e-mail accounts to keep track of, using webmail gets tedious after a while. Fortunately for those of us who like one-stop-shopping, Mozilla's Thunderbird is open source and strongly supported under Linux/Solaris/OSX (as well as Windows).

Evolution is another mail client alternative for Linux users, while Apple has given OSX a lovely proprietary client.

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# Open Office

Microsoft Office is one of the strongest arguments for use of Windows. OpenOffice.org, or OOo for short, has put together the strongest Office replacement suite available for free. OpenOffice.org can create and edit the following types of documents:

- Word (.doc)
- Excel (.xls)
- PowerPoint (.ppt)
- Plus a thousand others that MS Office won't handle

## Alternatives

OpenOffice.org is the closest thing there is to a Microsoft Office suite on Linux. However, there are always single-serving substitutions for the individual components of Office:

- AbiWord is a handy word processor (Word)
- Gnumeric makes an effective spreadsheet app (Excel)
- Slideshow presentations (like this one!) can be made using LaTeX (PowerPoint)

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# Instant Messaging

Pidgin is a wonderful instant messaging client. It operates much like Thunderbird/Outlook, except that instead of your e-mail accounts, it is consolidating your IM accounts. Pidgin can handle these protocols:

- AIM
- MSN
- Yahoo
- GoogleTalk
- MySpaceIM
- and more that you've never heard of!

Also available on Windows. OSX users: try Adium instead.

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# Playing Audio on Linux

One of the biggest (legitimate) complaints about Linux is the somewhat sketchy sound support. While getting audio working may take some effort, there are lots of free options for playback once you're up and running.

- **beep-media-player** (BMP) is a nice approximation of WinAmp Lite for easy playlisting of your MP3 collection
- For those who like internet radio and streaming podcasts, try BMPx, Amarok, or the new AudioSource

There are tons of choices out there, so feel free to explore until you find something you like.

# Editing Audio on Linux

Need something to put your mixtapes together or produce your own album? It may be tough finding something quite as powerful as the proprietary apps you're used to using on Windows, but give these a shot:

- Audacity - an intuitive interface on a fairly fully-featured editor, good for casual users
- Ardour - perhaps the closest you can get to professional-grade audio mixing on Linux
- SND - the geekiest option - powerful, but with a suitably steep learning curve

# Lecture Outline

1 Your Parents' Linux
- Web Browsing and E-Mail
- Office Replacements
- Communications

2 Multimedia
- Audio
- Video
- Images

3 Development
- Text Editing
- Compiling Code
- Installing Programs from Source

# Playing Video on Linux

Much like audio playback, there are endless players to choose from when it comes to video. Here are some old standbys:

- **mplayer** - Powerful, simple, with an iffy GUI
- **Xine** - Better GUI than mplayer, but less reliable codec support
- **VLC** - familiar to Windows users, plays everything under the sun

# Editing Video on Linux

Apple and Adobe really have a corner market when it comes to video editing software. If you insist on using Linux because you hate Macs, some options still exist for you.

- Cinelerra - as close as you'll get to commercial-grade editing software, too complicated for home movie editing
- Kino - designed for DV editing, easy to grab from your DV camcorder and start cutting
- Jahshaka - Aims for ease-of-use with an intuitive GUI

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# The GIMP

If you are a big fan of Photoshop, it may take some tears before you're used to using The GIMP (Gnu Image Manipulation Program). However, if you normally do your photo editing in MS Paint, The GIMP is a big step up.

Not quite as fully-featured as Photoshop, GIMP still packs plenty of punch for most image editing tasks.

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# VI and Emacs

For those who don't like text editing in the shell, there are two strong options. **gvim** provides a nice GUI frontend for the **vim** editor...

...and **xemacs** provides the same for **emacs**. While these are the two most popular cross-platform editors, there are other strong system-specific editors like **gedit** available as well.

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# GCC

Since the Linux kernel is written in C, it stands to reason that it'd be easy to compile C on a Linux system, right? Anyway, GCC stands for "Gnu C Compiler", but it is actually capable of compiling a ton of languages:

- C, C++
- Objective C, Objective C++
- Fortran
- Ada
- and a few others nobody writes in!

To get some idea of how flexible **gcc** can be, open the manpage and hold PGDN for a good 10 seconds.

# Other Languages

C-based languages aside, there are many popular languages that require a compiler or interpreter. Odds are, if you are writing code in a more obscure language, you know what to do to run it. Here's a short list of easily supported languages:

- Java
- Python
- Ruby
- Perl
- Lisp/Scheme
- OCaml

# GDB

## Gnu DeBugger

**gdb [options] program**

- Debugging tool for C, C++, and Modula-2 (?)
- Used to closely examine program execution to find/fix bugs

We could easily spend a lecture talking about **gdb**, but debugging is an art form, and if you have a project large and complex enough for **gdb**, you probably know what you're doing already.

# Lecture Outline

1. Your Parents' Linux
   - Web Browsing and E-Mail
   - Office Replacements
   - Communications
2. Multimedia
   - Audio
   - Video
   - Images
3. Development
   - Text Editing
   - Compiling Code
   - Installing Programs from Source

# Configure

90% of the time you will be able to use your distribution's package manager to install programs easily. Sometimes, however, you have to build stuff from source code. Fortunately, the process here is almost always the same.

## Step 1: Extract the Source Code

- **tar -zxvf** <**archive**> - extracts a .tar.gz archive
- **tar -jxvf** <**archive**> - extracts a .tar.bz2 archive

## Step 2: Configure

**cd** into the **tar**-created directory

**./configure**

- The **configure** script checks your system so that the code can be compiled and installed optimally.

# Make

## Step 3: Compiling the Code

Assuming **configure** ran cleanly, you're ready to compile the code. Just run:

**make**

- Compiles the code according to a pre-specified Makefile
- This step can take a while!

## Step 4: Installing Your Program(s)

Once **make** is finished, you're ready to install your new software.

**make install && make clean**

- **make install** moves all the binaries to their proper places
- **make clean** removes leftover files used when compiling

# Systems That Aren't Yours

**Note:**

If you don't have root permissions, you won't be able to **make install** files into the usual */usr/bin* or */usr/local/bin*.

If this is the case, you can install programs locally in your home directory with a little bit of finesse.

- **./configure –help** will print a long list of options for destination directories, etc.

Use **configure** to control where all of the files end up and you should get a working installation without having to worry about permissions.