

# Opening Remarks

- We'll get through most of the text
- We'll skip chapters 6 & 7 - vi and emacs editors
- You learn an editor when you use it, not by reading a book
- I'll show you *nano*, a simple editor
- *nano* is good enough for our purposes
- For work in this course you will need to use a text editor

# Opening Remarks

- You can use any text editor **that runs on Unix/Linux -- not** Notepad or MS Word
- I won't cover chapter 9 - TC shell
- We'll use the bash shell in this course
- bash has more features than the TC shell, and it is the default shell in most Linux installations
- Modern installers are very good - almost idiot proof

# Opening Remarks

- The Boston Linux User Group has periodic Installfests at MIT
- To learn more about this user group, go to [blu.org](http://blu.org)
- Work for the course will be done on a Linux machine running Ubuntu
- We will only use the Windows machines in the Lab to connect to the Linux network
- Please feel free to bring a laptop to this class, which you can use instead of the Windows machine for class exercises

# Connecting to the Linux Machine from Home

- You can connect to the Linux machine we'll use in this class from home
- To do this, you'll need an Internet connection and an [ssh client](#)
- An SSH client uses the [SSH](#) protocol to connect to a remote Unix/Linux machine
- An SSH client allows you to use one machine to give you a command line on another machine that runs Unix/Linux
- If you have a Mac at home, open Terminal and run  

```
ssh UNIX_USERNAME@it244a.cs.umb.edu
```

# Connecting to the Linux Machine from Home

- The SSH client I recommend for Windows is PuTTY which you can get here:

`http://www.chiark.greenend.org.uk/`

`~sgtatham/putty/download.html`

- Download the file `putty.exe`
- When using PuTTY, use the parameters you will find in today's class exercise

# Connecting to the Linux Machine from Home

- Contact me if you have trouble connecting
- If you have a Mac, read the instructions here:

**`http://www.cs.umb.edu/~ghoffman/linux/  
remote_access_mac.html`**

- If you have a Windows machine, read the instructions here:

**`http://www.cs.umb.edu/~ghoffman/linux/  
remote_access_mac.html`**

# Operating Systems

- The operating system is the software that allows you to use the computer
- The textbook defines an operating system as  
**A control program for a computer that allocates resources, schedules tasks, and provides the user with a way to access resources**
- The operating system provides both resources and protection
- It's like a city government
- The city government provides resources - water, sewage, snow plowing
- It also provides protection - police and fire departments

# Operating Systems

- The operating system provides resources, such as like access to memory, files and devices
- It also provides protection to keep one program from interfering with another
- Every computer has an operating system of some sort
- "Invisible" computers, like those in cars and cell phones, have simpler operating systems than those on a desktop machine
- They use embedded operating systems

# Unix

- Unix is an operating system developed as a research tool at Bell Labs
- Bell Labs was created as a research facility for the Bell telephone system
- Some of the most important advances of the 20th century came from Bell Labs, where researchers won many awards
- Unix has a number of powerful features
- For this reason, it is often used in academic, engineering and research environments

# Unix

- Unix was developed before Graphical User Interfaces (GUIs) become common
- The Unix approach is to create many small programs that each do *one thing* well
- Unix allows simple tools to be strung together to perform a complex task; this allows people to write commands that do complicated things on the fly
- Unix was given away free to universities, and many Computer Science students learned to like it
- Talented programmers made many contributions to Unix, and they continue to contribute to Linux

# Different Unix Releases

- There is no single collection of software that can be called Unix or Linux
- Unix distributions consist of
  - The kernel
  - Utilities
  - Applications
- The kernel is always present in the computer's memory and provides key services, like access to the CPU and to files
- You can only access the kernel indirectly by using other programs

# Different Unix Releases

- The utilities help you use computer resources
- Applications are programs that are used to get work done -- such as a word processor or an email client
- Down through the years a couple of different Unix distributions have emerged
- You can think of a "distribution" as a specific operating system – the Unix/Linux kernel plus particular utilities and applications.

# Different Unix Releases

- One popular Unix distribution is BSD, which stands for Berkeley Software Distribution
- It was developed by the Computer Systems Research Group at the University of California at Berkeley
- Berkeley charges nothing for this Unix
- The BSD license is the least restricted open source license
- Check <http://www.opensource.org/licenses/bsd-license.php> for more information
- The Unix used in the Mac OS comes mainly from BSD

# Different Unix Releases

- Another popular Unix distribution is Unix System V
- It is a commercial release of Unix by AT&T the corporate sponsor of Bell Labs
- There were four major releases of System V
- Release 4, abbreviated SVR4, enjoyed the greatest commercial success

# Different Unix Releases

- The Unix used at UMB was, until recently, Solaris
- Solaris was developed by Sun Microsystems as a commercial product
- Subsequently, Sun released most of the codebase as an open source project called Open Solaris
- Sun was acquired by Oracle in January 2010
- Solaris has a reputation for being a somewhat non-standard Unix

# GNU and the Free Software Foundation

- Richard Stallman is a legendary figure in the Unix world
- He received a MacArthur "genius" grant
- He worked for many years at the MIT Artificial Intelligence Laboratory
- Stallman's early programming experience took place in an environment of free exchange of software code and ideas

# GNU and the Free Software Foundation

- In these early days, companies were not too concerned about trade secrets
- Stallman and his fellow programmers had access to the source code for the programs they used
- When a bug was discovered, they could look in the source code and fix the problem *directly*
- As the software industry became more commercialized companies refused to release their source code

# GNU and the Free Software Foundation

- This led Stallman to become disenchanted with proprietary software
- He could not read the software to fix the bugs he found
- Stallman found it very frustrating to have to report a bug and wait for a fix
- In 1983, Stallman announced plans for the GNU Operating System
- It would be distributed under the GNU License and would be compatible with Unix
- GNU is a recursive acronym that stands for "GNU is not Unix"

# GNU and the Free Software Foundation

- In 1985 Stallman founded the Free Software Foundation
- It's mission was to develop GNU software and to advance the cause of free software
- By "free" Stallman meant that every user has the right to read and modify the source code
- It did not mean that an organization could not charge for the software
- "Free as in speech, not beer"

# GNU and the Free Software Foundation

- The Free Software Foundation charged a modest amount for its software
- Stallman's definition of free software is very strict, and he distinguishes it from open source software
- Stallman has written two books on the subject of free software, which can be obtained from the Free Software Foundation, <http://shop.fsf.org/category/books/>

# GNU and the Free Software Foundation

- The GNU License is fairly elaborate
- If any person or company makes a change to software distributed under the GPL, they must distribute the improved source code under the same license
- This means a company cannot take software distributed under the GPL, make some improvements, and then sell it as proprietary software
- This prevents commercial developers from modifying the code and then selling the modified package without publishing the changes they have made

# GNU and the Free Software Foundation

- The GNU project first worked on developing tools that would be needed to complete the operating system
- Their most important contributions were the programmable text editor, *emacs*, and the compiler, *gcc*
- Most of Unix is written in C, so developing the GNU C compiler was an important first step
- Work on the kernel, the heart of the operating system, was saved for last, since it would be the most difficult
- The GNU kernel was to be named Hurd

# Linus Torvalds

- In the early 90's as Linus Torvalds was working on his Master's thesis
- At this time he began a personal project that eventually became a Unix-like kernel
- He released the source code for this kernel to the public
- This kernel was widely adopted
- Together with many of the GNU tools, it formed the core of what came to be called Linux
- Today Linus Torvalds is the chief architect of the Linux kernel

# Linux Distributions

- There are many Linux distributions, each released by a different group
- Each distribution consists of different choices, from available options for each component of the operating system
- The two most common distributions are Ubuntu and Red Hat
- CentOS is a noncommercial Linux distribution, which uses source code provided by Red Hat for their commercial Linux distribution

# Ubuntu

- The machine we'll be using in this class is [it244a](#)
- It is a virtual machine running Ubuntu 14.04.1 LTS
- Ubuntu is based on Debian Linux
- Development of Ubuntu is led by [Canonical, a company based on the Isle of Man in Great Britain and owned by South African entrepreneur Mark Shuttleworth](#)

# Ubuntu

- Ubuntu is named after the Southern African philosophy of "*ubuntu*" -- which is often translated as "humanity towards others"
- A new Ubuntu version is released every 6 months
- The Ubuntu version number is composed of the year and month of its release
- So version 14.04 was released in April of 2014

# Ubuntu

- LTS stands for "Long Term Support"
- An LTS version will be supported for 5 years after its release
- A new LTS version is released every 2 years
- Every Ubuntu release also has a name
- The name consists of two words:
  - An adjective
  - An animal whose name shares the same first letter

# Ubuntu

- The name for Ubuntu 12.04 LTS is "Precise Pangolin"
- A pangolin is a mammal that eats ants and is sometimes called a scaly anteater
- One of the reasons for Ubuntu's success is its package manager
- If you type in a command that is not installed, Ubuntu will suggest packages that will install the software

# Shells

- The shell is the program you interact with at the command line
- The shell accepts input from the user and executes the commands entered
- The shell is the program you use to talk to the kernel
- Modern shells provide a number of features that make them easier to use, such as command completion
- Down through the years there have been a number of different Unix shells
- Today, the two most common are the TC shell and Bash
- Bash has more features, and it is the shell we will be using in this course

# Unix Commands

- To run a Unix command, you type the command -- followed, perhaps, by options and arguments -- and then hit Enter (or Return on a Mac)
- Options change what the command does
- For example the *ls* command lists the contents of a directory

```
$ ls
```

```
foo.txt  it244  work
```

# Unix Commands

- When *ls* is used with the *-l* (for long) option, more information is presented

```
$ ls -l
total 3
-rw-r--r-- 1 it244gh libuuid  16 2012-06-26 16:19
foo.txt
lrwxrwxrwx 1 it244gh libuuid  34 2012-02-07 09:46
it244 -> /courses/it244/s12/ghoffmn/it244gh
drwxr-xr-x 2 it244gh libuuid 512 2012-06-27 11:08
work
```

# Unix Commands

- Arguments are the objects on which the command operates
- If *ls* is given the name of a directory as an argument, it will display the contents of that directory:

```
$ ls html
```

```
it244_f14_html  it341_f14_html
```

- Most Unix/Linux commands are very short (to save typing), and usually have *mnemonic* value
- For example, *ls* stands for "list"

# Today's Class Exercise

- Today you will use the Windows machines in the Lab to connect to the virtual machine [it244a](#)
- You will find the instructions on the Class Exercise
- We'll be using the Linux machine named [it244a](#) for all our work in this course
- In the exercise, you will enter your first Unix commands
- Type the commands specified, including any options, and then the arguments, if any

# Today's Class Exercise

- Options are preceded by a `-` or `--`
- You will also customize your Unix session
- You will create a file to forward Unix email to your regular email account
- If you do not create the `.forward` file correctly, you will not receive your scores by email