

CSE 399-004 Unix/Linux Skills

Spring 2006

Administrative

- Instructor
 - Marc Corliss (mcorliss@cis.upenn.edu)
- Date and time
 - Monday, Wednesday, Friday 1-2pm
 - Classes from 1/9 to 2/24 (only first half of spring semester)
- Room
 - Monday, Wednesday: Lecture in Towne 309
 - Friday: Lab in Moore 100B
- Textbook
 - None, we'll use the web and man pages as resources
- Webpage
 - <http://www.cis.upenn.edu/~mcorliss/teaching/cse399>

Unix/Linux Skills - Marc Corliss

Enrolling

- Course is not on the roster (since recently added)
- Need to register for Python course (CSE 399-004)
 - Unix/Linux skills is in first half of the semester (1/9 to 2/24)
 - Python course is in the second half of the semester
- Can attend either this course, Python course, or both
 - However, in all cases, only half credit will be granted
- Talk to me if you have questions

Unix/Linux Skills - Marc Corliss

Grading

- Weekly homeworks (60%)
 - 6 of them, worth 10% each
 - Should take < 2 hours a piece
 - Usually handed out and collected on Mondays
 - Except first week due to MLK day, handed out and collected on a Wednesday
 - May talk with classmates, but should do the work individually
 - See Penn's Code of Academic Integrity
- Final exam (30%)
 - Take-home exam
 - Should take < 5 hours
 - Should work on the exam individually
 - Tests all aspects of the course
 - Due on 2/27, the Monday after class ends
- Participation (10%)

Unix/Linux Skills - Marc Corliss

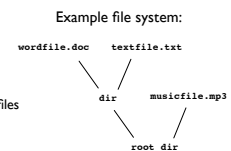
What is Unix and Linux?

- Unix and Linux are operating systems
 - Like Windows and Mac OS X
- Ok, so what is an operating system (OS)?
 - Control software
 - Manages machine resources (e.g., processor, disk)
 - Multiplexes machine for each user
 - Appears to each user that they have sole control of the machine
 - Most important software on any machine
 - If computer system were human being then OS would be the brain
 - Gross oversimplification
 - For more precise description take cse380

Unix/Linux Skills - Marc Corliss

Important OS Abstractions

- Files
 - File = a collection of data
 - e.g., word document (filename.doc)
 - Directory = special file that contains other files
 - Filesystem is a tree
- Processes
 - Process = a program in execution
 - Processes work with files
 - Word reads file to display document to user
 - Word writes file when saving document for user



Unix/Linux Skills - Marc Corliss

(Very Brief) Unix History

- Developed in the late 60's and early 70's at Bell Labs
 - Main contributors were Ken Thompson and Dennis Ritchie
 - Developed from Multics, a failed OS out of MIT, Bell Labs, and GE
 - Multics = MULTiplexed Information and Computing System
 - Many novelties including multi-user (timesharing) system, enhanced security
 - Unix = stripped-down version of Multics
 - Written in C programming language

Unix/Linux Skills - Marc Corliss

(Very Brief) Linux History

- Developed in early 90's by Linus Torvalds
- Modeled after Unix (hence the name: linux+unix=linux)
- Targeted the personal computer
- Unlike Unix (and most other OSs), Linux is **open-source**
 - Source code is available for free!
 - How does anyone make any money off of Linux?
 - Mainly by providing support
- Unix != Linux
 - However, this course uses them interchangeably

Unix/Linux Skills - Marc Corliss

Why Learn Unix/Linux?

Windows is dominant OS, so why learn Unix/Linux?

- Dominates some domains
 - Servers, research labs, academia, and some parts of industry
- Not graphical-oriented (although this is changing)
 - Easier to understand what's happening at a low level
 - Separates the boys and girls from the men and women
- Can keep you from looking like a bonehead in a job interview
- Necessary for some Penn courses
 - cse240 (intro. to architecture), cse331 (networks and security), cse380/381 (operating systems), cse341 (compilers), cse371/372 (architecture)

Unix/Linux Skills - Marc Corliss

Why Learn Unix/Linux Continued

Windows is dominant OS, so why learn Unix/Linux?

- Up-and-coming OS (Linux only)
 - In the future, Linux may have bigger piece of the pie
- Source code available (Linux only)
 - Can make development easier in a Linux environment
- Fun!!!

Unix/Linux Skills - Marc Corliss

Course Goals

- Make the material accessible
 - Compare with Windows (and sometimes Mac OS X)
- Make you "dangerous"
 - Teach the basic concepts
 - Then, show how to use resources (man and web) to do more advanced stuff
 - Many resources are linked off of the course webpage
- Have fun!!!

Unix/Linux Skills - Marc Corliss

Course Schedule

- Part I
 - Shell
- Part II
 - Emacs editor
- Part III
 - Scripting
- Part IV
 - Applications
- Part V
 - User-level package installation

Unix/Linux Skills - Marc Corliss

Part I: Shell

- Similar to Windows explorer (note: not internet explorer)
 - Both can manipulate files, directories
 - Two big differences:
 - Explorer graphical, shell textual
 - In explorer, use mouse to move things around (some things require keyboard as well: renaming a file)
 - In shell, enter "commands" at a prompt to do similar operations
 - Shell has extensibility
 - Users can define their own commands using basic commands
 - Consequently, shell much more powerful than explorer
 - Actually, shell is more equivalent to DOS on Windows
 - But no one uses DOS
 - Example: move (rename) file file_foo to file_goo

```
prompt$ mv file_foo file_goo
```

Unix/Linux Skills - Marc Corliss

Power of the Shell

- Problem: Need to remove all pirated music on my machine
 - Assumption 1: all my music is scattered all over the filesystem
 - Assumption 2: all my pirated music is by the Beatles or Rolling Stones
 - Assumption 3: music files are within directories that contain the artist name
- Solution:

```
prompt$ cd / \
find . \( -type d -a \
\ \( -name \*Beatles\* -o \
\ -name \*Rolling\ Stones\* \) \) \
| while read dir \
do find "$dir" \( -name \*.mp3 -o -name \*.Mp3 -o \
\ -name \*.m4p -o -name \*.m4a \) \
-exec rm {} ';' \
done
```

How would you do this with explorer?

Unix/Linux Skills - Marc Corliss

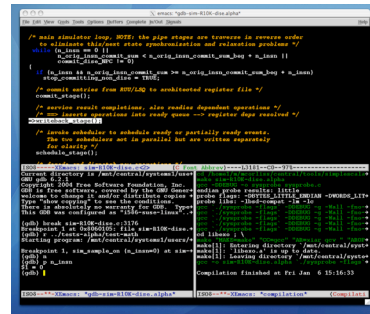
Part II: Emacs

To edit a file (i.e., modify its contents) in Unix/Linux, often use an editor

- On Windows, closest application is TextEdit
 - Both can edit ascii (simple text) files
 - Both are mainly text-based
- There are some big differences:
 - Emacs designed for programming
 - Emacs supports more keyboard commands
 - Emacs is extensible:
 - Can add new commands
 - Can add new features (since source code available)

Unix/Linux Skills - Marc Corliss

Emacs Screenshot



Unix/Linux Skills - Marc Corliss

Part III: Scripting

- Shell is powerful, but interactive
 - Everytime you want to execute a command must re-enter it
 - Annoying for long, complex commands
- Solution: wrap commands up into a shell script
 - Type string of commands into an ascii file (i.e., using emacs)
 - Then, can run commands by running shell script
 - Can even make scripts of scripts, etc.
- Other types of scripting besides shell scripting
 - Perl (more on this later)

Unix/Linux Skills - Marc Corliss

Scripting Example

- Similar example as before: want to find music (by artist) in filesystem
 - This time not removing files
 - Command:

```
prompt$ cd /; find . \( -type d -a -name \*Beatles\* \) | \
while read dir; do find "$dir" -name \*.mp3 -o -name \*.Mp3 \
-o -name \*.m4p -o -name \*.m4a; done
```

- Put command in new file (using emacs) findmusic.sh

- Modified slightly to take artist name as an argument
- findmusic.sh:

```
ARTIST=$1
cd /; find . \( -type d -a -name \*$ARTIST\* \) | while read
dir; do find "$dir" -name \*.mp3 -o -name \*.Mp3 -o -name
\*.m4p -o -name \*.m4a; done
```

Unix/Linux Skills - Marc Corliss

Scripting Example Continued

- Can run findmusic.sh as follows:

```
prompt$ findmusic.sh Beatles
/home/mcorliss/music/The Beatles/Abbey Road/01 Come Together.m4p
/var/music/The Beatles/A Hard Day's Night/01 A Hard Day's Night.Mp3
```

- Can even make scripts of scripts

```
removemusic.sh:
ARTIST=$1
findmusic.sh $ARTIST |
while read filename
do rm "$filename"
done
```

Unix/Linux Skills – Marc Corliss

Part IV: Applications

- With the shell, emacs, and scripting we can do a lot
 - Still, what if we want to do word processing or use the www?
- Look at a number of Unix/Linux applications
 - Open-office (equivalent to Microsoft Office)
 - Word processing, spreadsheet, presentation
 - Firefox (equivalent to Internet Explorer)
 - Browsing the web
 - Thunderbird (equivalent to Outlook Express)
 - Email, newsgroups

Unix/Linux Skills – Marc Corliss

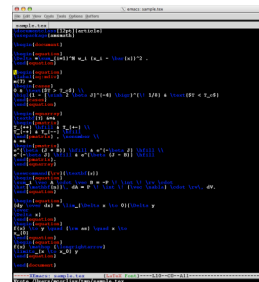
More Applications -- Latex

- For writing technical papers and presentations
 - No good windows equivalent, closest is Microsoft Word
 - Text-based unlike Word (What You See Is What You Get)
- Has many virtues
 - Symbols can be entered by hand instead of by mouse (useful in math papers)
 - Can define macros (extensible)
 - Formatting can be done globally
 - For Windows programs, such as Microsoft Word, often need to make changes throughout entire document
 - Especially important for complex documents such as scientific papers
- Has some disadvantages too
 - Text-based
 - Must compile before viewing
 - Not good for short, simple documents

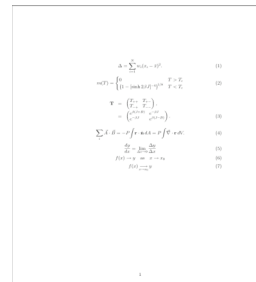
Unix/Linux Skills – Marc Corliss

Example Latex Document

Source file:



Output file (pdf):



Taken in part from: <http://iso.clarku.edu/tutorials/TeX/intro.pdf>

Unix/Linux Skills – Marc Corliss

More Applications -- CVS

- Development problems:
 - What if you want to develop concurrently on the same program?
 - What if you want version tracking when programming?
- Solution: CVS (or Concurrent Versions System)
 - Can edit program files at the same time
 - So long as not simultaneously editing the same piece of code
 - Can keep up-to-date when working with a team
 - Can archive older versions of the software
 - Even works for other domains besides programming
 - For instance, writing a paper in latex
 - Each author can simultaneously work on a different section
- Who uses CVS?
 - Developers of Linux, NetBSD, FreeBSD, XFree86, Mozilla, Apache, etc.
 - i.e., lots of important development teams

Unix/Linux Skills – Marc Corliss

Part V: User-Level Installation

- Discuss how to install new software
- Two general ways of doing this:
 - As the privileged administrator (called `root`)
 - As an unprivileged user (e.g., `mcorliss`)
- Only going to look at user-level installation
 - Administering a unix/linux machine interesting, but could be its own course

Unix/Linux Skills – Marc Corliss

Next Time

- Lecture: Using the shell
- First homework handed out, due the following Wednesday (Jan. 18th)
- Schedule is online, see the webpage

Unix/Linux Skills - Marc Corliss