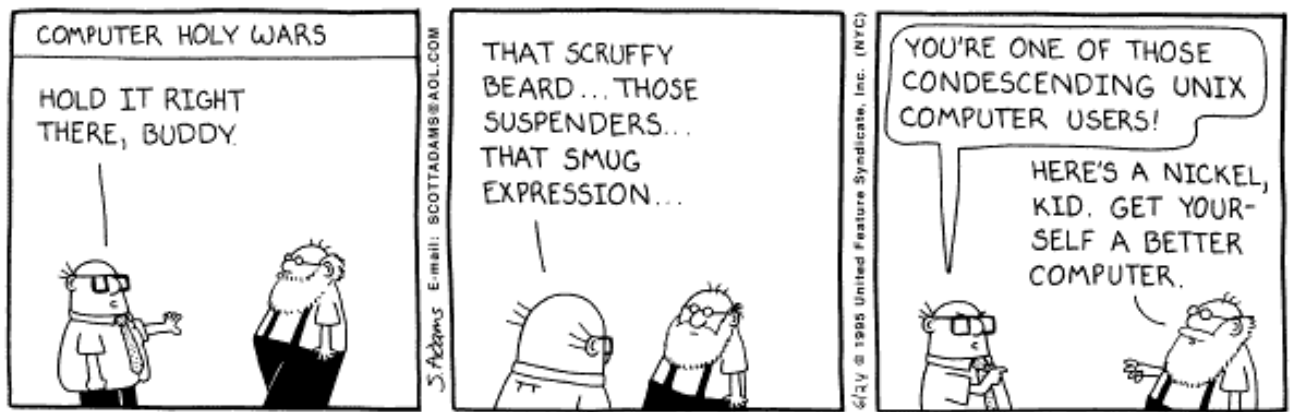


CS108 Software Systems: UNIX

Fall 2011



Course Info

`cs.utexas.edu/~edwardsj/teaching/2011fall/cs108`

Why Linux?

- Multi-user, multi-process operating system
- Open-source project with a rapidly expanding software base
- UNIX-like
 - The "UNIX Philosophy," as summarized by Doug McIlroy:
"Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface."
- A mature desktop environment
- Adaptable, Robust, Secure, Cost-effective (free)

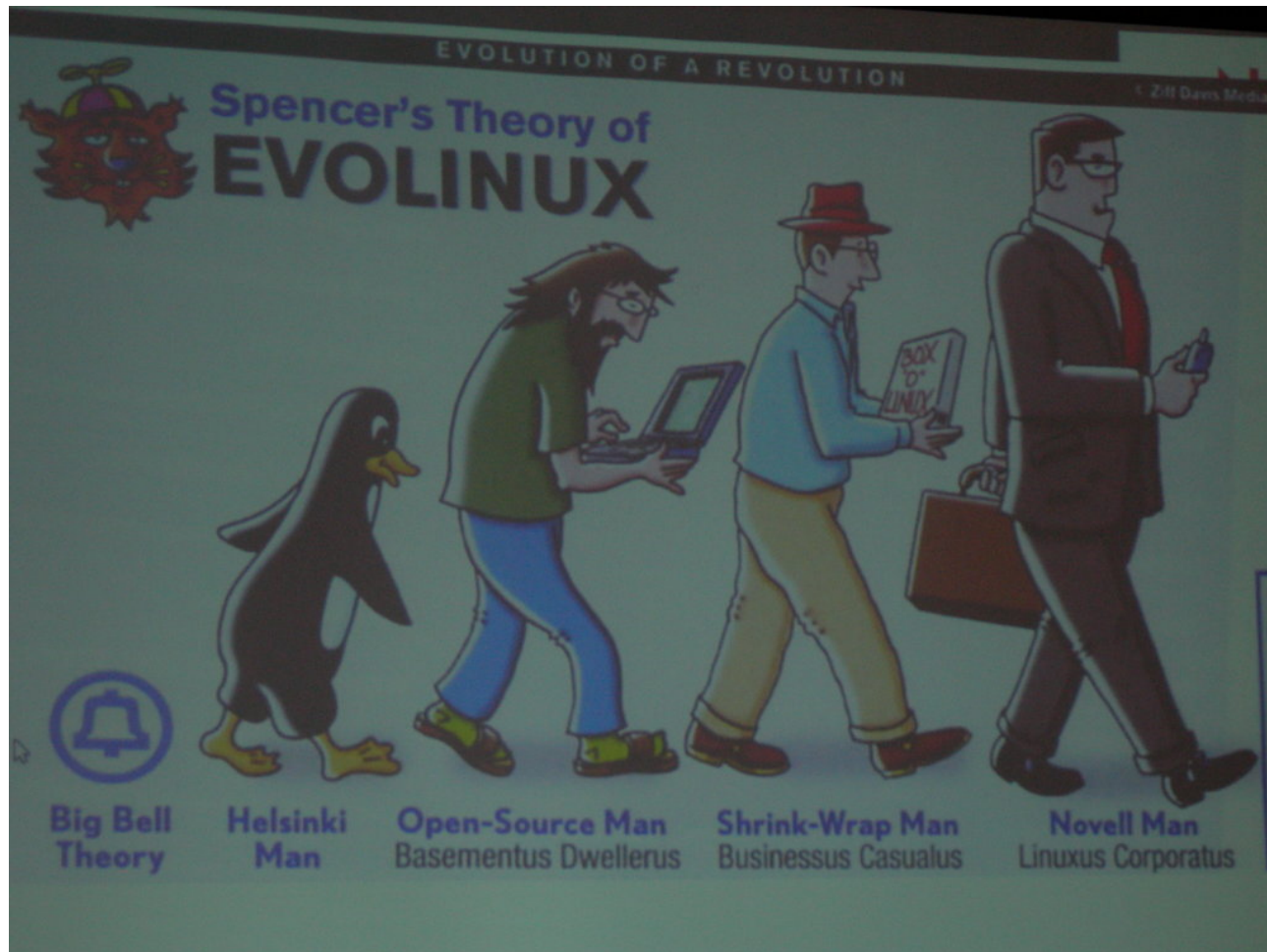
Why not?

- Cryptic commands and command-line options
- Few failsafes
- Coupled with open source movement (both good and bad)
- Designed for geeks by geeks, not *users*

Nevertheless, ...

- Linux is widely used in companies and academic departments, including UTCS
- People often assume you know your way around Linux
- John's opinion: excellent environment for working on programming projects and technical papers

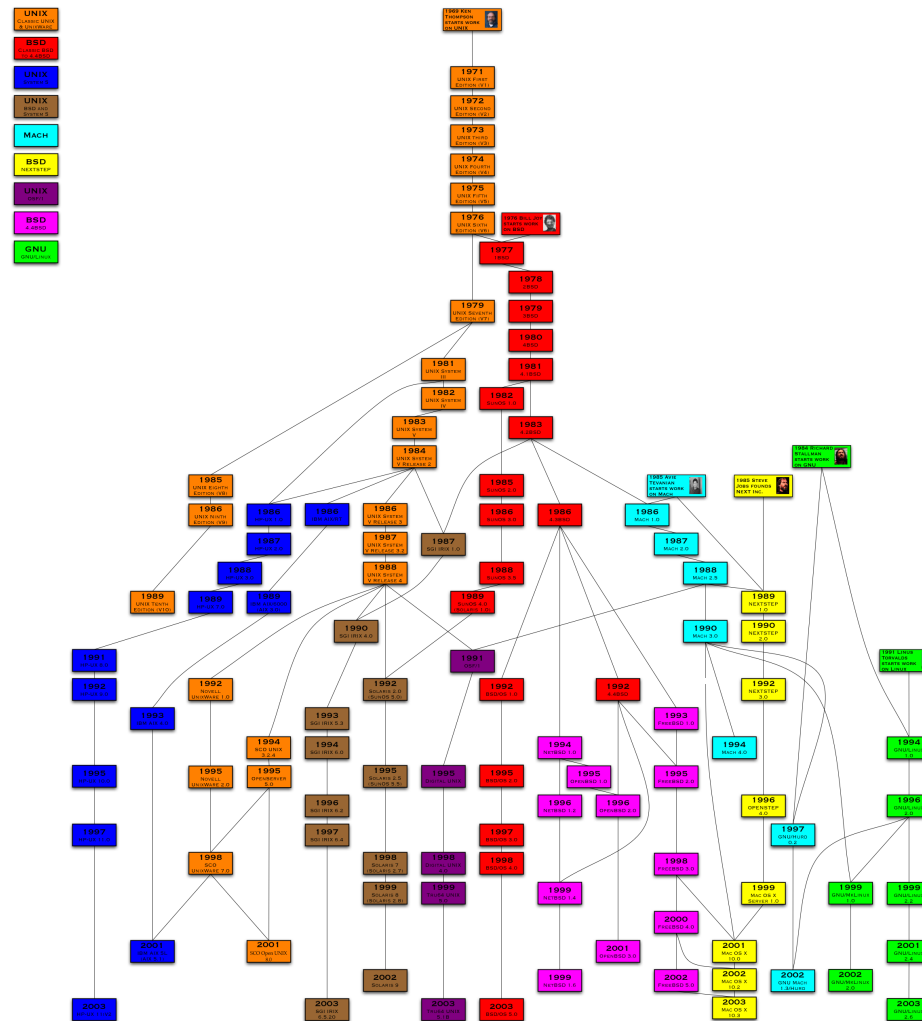
Ancestry of Linux



- 1969:** UNIX developed at Bell Laboratories by Ken Thompson & Dennis Richie, et al., for the DEC PDP-7 minicomputer.
- 1972:** Thompson develops the C programming language; work begins on rewriting UNIX in C.
- 1976:** Thompson introduces UNIX to UC Berkeley; BSD is born.
- 1984:** Richard Stallman begins the GNU project with Emacs.
- 1985:** Richard Stallman founds the Free Software Foundation (FSF) to support the GNU project.
- 1991:** Linus Torvalds writes a Minix-like kernel; GNU/Linux is born.



UNIX/Linux Timeline



The Linux Family

Collection of common software elements form usable OS.
No one “Linux” but rather many *distributions* that contain (nearly) the same software.

- Red Hat/Fedora – A popular and well-developed Linux.
- SuSE – A “professional” Linux.
- Debian – A sophisticated 100% free Linux.
- Ubuntu – A popular Debian-derived version of Linux.

More distribution information at

<http://www.distrowatch.com/>

GNU

- GNU (Gnu's Not UNIX) - “complete UNIX-compatible software system”
- FSF - Free Software Foundation (Richard Stallman)
 - “Free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer.” – GNU Free Software Definition
 - “Free software is software that users have the freedom to distribute and change. Some users may obtain copies at no charge, while others pay to obtain copies.” – Richard Stallman

Components of a Linux System

- **Kernel** – The core OS code that manages physical resources (CPU, memory, hard drives, etc.)
- **Shells** – Programs providing a text-based interface to the OS, a.k.a. the command line interface (CLI).
- **Commands, utilities, and applications** – The large set of programs that perform specific tasks, e.g. moving files, printing documents, and browsing the web.
- **Compilers** – Software development tools for converting human-understood languages into computer-understood languages.
- **Desktops / Window Managers** – Software providing a graphical user interface (GUI).

The Two Faces of Linux

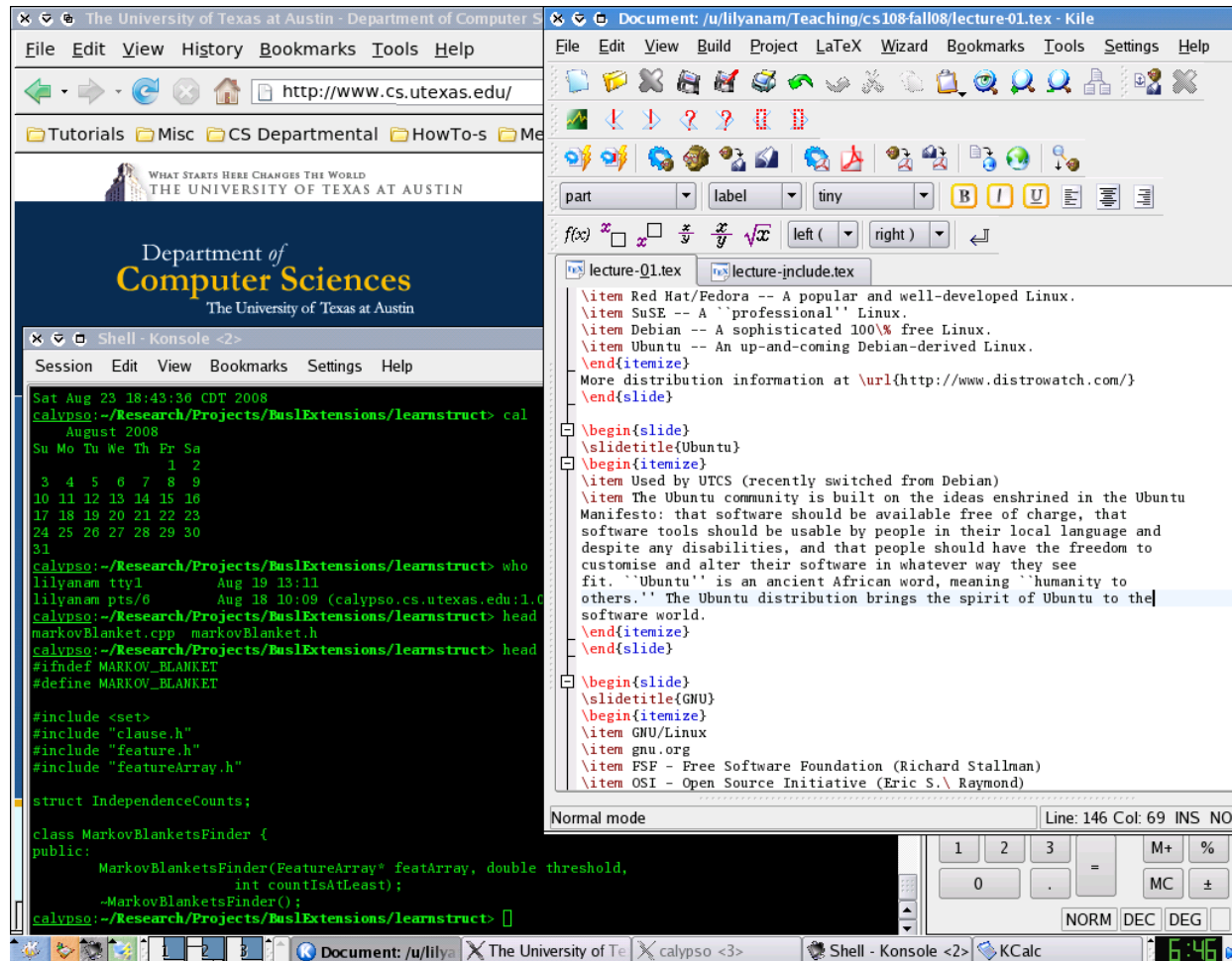
```
Sat Aug 23 18:43:36 CDT 2008
calypso:~/Research/Projects/BusExtensions/learnstruct> cal
  August 2008
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
calypso:~/Research/Projects/BusExtensions/learnstruct> who
lilyanam tty1          Aug 19 13:11
lilyanam pts/6        Aug 18 10:09 (calypso.cs.utexas.edu:1.0)
calypso:~/Research/Projects/BusExtensions/learnstruct> head -n15 markov
markovBlanket.cpp  markovBlanket.h
calypso:~/Research/Projects/BusExtensions/learnstruct> head -n15 markovBlanket.h
#ifndef MARKOV_BLANKET
#define MARKOV_BLANKET

#include <set>
#include "clause.h"
#include "feature.h"
#include "featureArray.h"

struct IndependenceCounts;

class MarkovBlanketsFinder {
public:
    MarkovBlanketsFinder(FeatureArray* featArray, double threshold,
                        int countIsAtLeast);
    ~MarkovBlanketsFinder();
calypso:~/Research/Projects/BusExtensions/learnstruct> █
```

The Two Faces of Linux



Connecting to a Linux System

Every user of a Linux system has a ***user account***, identified by a ***username*** and authenticated with a ***password***.

Linux systems may be accessed in a variety of ways:

- Console terminal
- `telnet` (insecure)
- `ssh` (secure)
- VNC

The Linux architecture allows almost all locally performable operations to be performed remotely as well.

Interacting with the Shell

The textual interface of a Linux system is provided by a *shell* program, which performs a READ-EVALUATE-EXECUTE loop.

1. Display a *prompt*, indicating readiness.
2. READ a line of input.
3. EVALUATE any meta-syntax.
4. EXECUTE the specified command(s).

Remote Log-in Using SSH

SSH allows you to establish a secure connection to a UNIX host from your local machine. To connect to a UTCS host:

1. Find a host that has a low load:

```
apps.cs.utexas.edu/unixlabstatus
```

2. Use a client SSH program to connect to theMachine.cs.utexas.edu:

- OS X:

- start the Terminal *or* X11 and type:

```
ssh user@theMachine.cs.utexas.edu
```

- Windows:

- SSH Secure Shell @ <http://www.ssh.com/>
- PuTTY @
www.chiark.greenend.org.uk/~sgtatham/putty/
- Enter the hostname (theMachine.cs.utexas.edu)
(and user name if available) and press connect.
- Another UTCS UNIX host:
 - type
`ssh theMachine.cs.utexas.edu` *or just*
`ssh theMachine`

Using VNC

VNC is a two-part system consisting of a server on a UTCS machine and viewer on the local machine (where the display is).

1. Find an available VNC Session

```
http://www.cs.utexas.edu/facilities/  
accommodations/public_labs/vnc_status/vnc.html
```

2. SSH to that UTCS machine (for me, this is antones)

3. Start a `vncserver` on the UTCS machine:

```
antones:~> vncserver
Found /lusr/opt/realvnc-4.1.1/lib for http connections.

New 'antones.cs.utexas.edu:1 (erozner)' desktop is antones.cs.utexas.edu:1

Starting applications specified in /u/erozner/.vnc/xstartup
Log file is /u/erozner/.vnc/antones.cs.utexas.edu:1.log
```

4. The VNC viewer can now connect to the VNC server, e.g.

`antones.cs.utexas.edu:1`

- OS X
 - Chicken of the VNC@http:
`//www.apple.com/downloads/macosx/networking_security/chickenofthevnc.html`
- Windows
 - RealVNC @ `http://www.realvnc.com/`
 - TightVNC @ `http://www.tightvnc.com/`
 - UltraVNC @ `http://ultravnc.sourceforge.net/`

5. When finished, terminate the VNC server from the UTCS machine:

```
antones:~> vncserver -kill :1  
Killing Xvnc process ID 16384
```

6. For more detailed instructions, see:

[http://www.cs.utexas.edu/facilities/
documentation/using_vnc/index.shtml](http://www.cs.utexas.edu/facilities/documentation/using_vnc/index.shtml)

Homework 1

`cs.utexas.edu/~edwardsj/teaching/2011fall/cs108/getting_started.html`