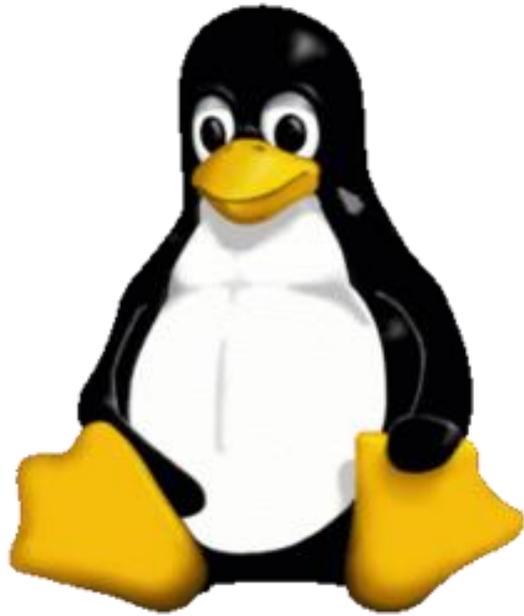


Introduction to Linux



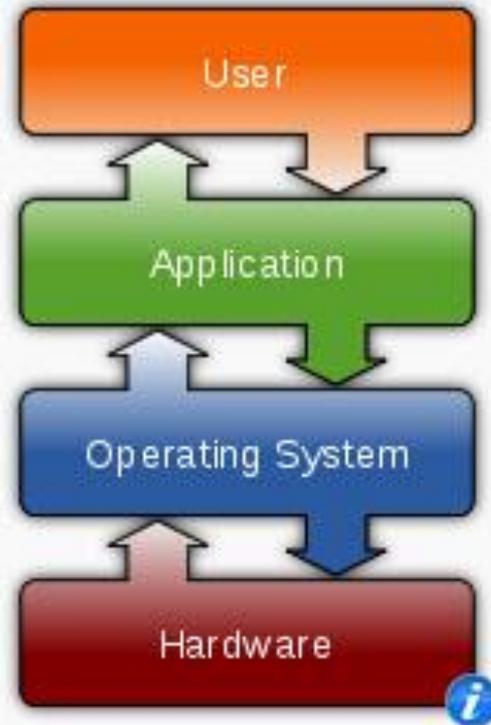
“Linux at the Command Line”

Don Johnson of BU IS&T

What is Linux?

It's an Operating System

Operating systems

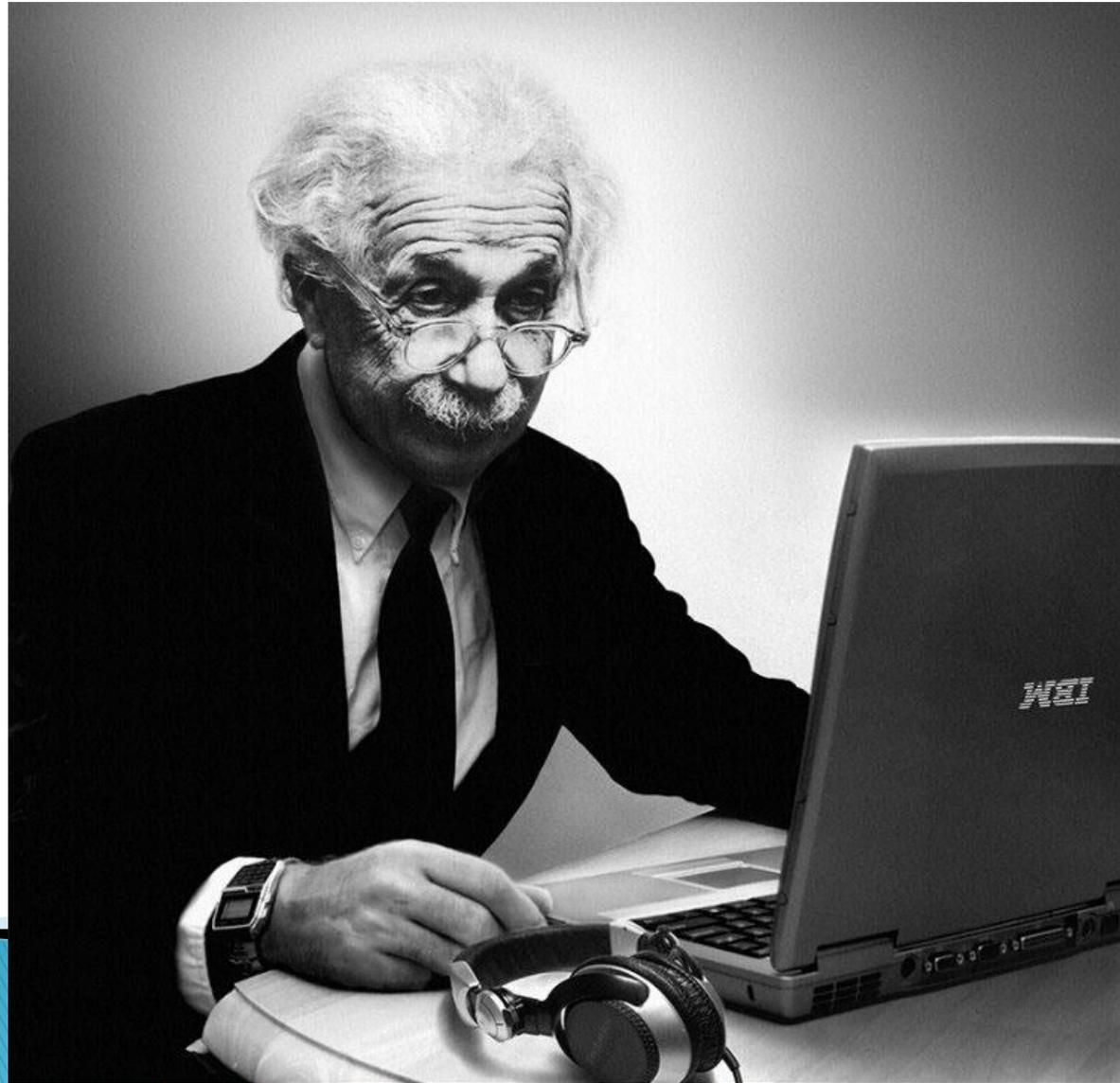


Common features

- Process management
- Interrupts
- Memory management
- File system
- Device drivers
- Networking (TCP/IP, UDP)
- Security (Process/Memory protection)
- I/O



What is Linux?



The Most
Common O/S
Used By BU
Researchers When
Working on a
Server or
Computer Cluster

What is Linux?

- ▶ Linux is a Unix clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net.
 - ▶ Unix is a multitasking, multi-user computer operating system originally developed in 1969 by a group of AT&T employees at Bell Labs.
 - ▶ Linux and Unix strive to be POSIX compliant.
 - ▶ 64% of the world's servers run some variant of Unix or Linux. The Android phone and the Kindle run Linux.
- 

The Linux Philosophy

*The *Nix Philosophy of Doug McIlroy*

- (i) Make each program do one thing well. To do a new job, build afresh rather than complicate old programs by adding new features.
 - (ii) Expect the output of every program to become the input to another, as yet unknown, program. Don't clutter output with extraneous information. Avoid stringently columnar or binary input formats. Don't insist on interactive input.
 - (iii) Use tools in preference to unskilled help to lighten a programming task, even if you have to detour to build the tools and expect to throw some of them out after you've finished using them.
- 

Linux Has Many Distributions



What is Linux?

“Small programs that do one thing well”

(see [unix-reference.pdf](#))

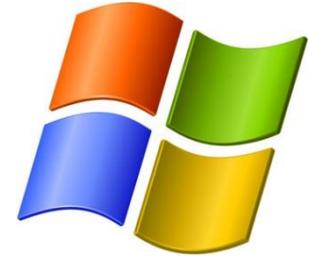
- ▶ **Network:** ssh, scp, ping, telnet, nslookup, wget
- ▶ **Shells:** BASH, TCSH, alias, watch, clear, history, chsh, echo, set, setenv, xargs
- ▶ **System Information:** w, whoami, man, info, which, free, echo, date, cal, df, free, man, info
- ▶ **Command Information:** man, info
- ▶ **Symbols:** |, >, >>, <, &, >&, 2>&1, ;, ~, ., .., \$!, !:<n>, !<n>
- ▶ **Filters:** grep, egrep, more, less, head, tail
- ▶ **Hotkeys:** <ctrl><c>, <ctrl><d>
- ▶ **File System:** ls, mkdir, cd, pwd, mv, ln, touch, cat, file, find, diff, cmp, /net/<hostname>/<path>, mount, du, df, chmod, find
- ▶ **Line Editors:** awk, sed
- ▶ **File Editors:** vim, gvim, emacs -nw, emacs

Connecting to a Linux Host

- ▶ You need a “xterm” emulator: software that emulates an “X” terminal and connects using the “SSH” secure shell protocol.
- ▶ You are sitting at the “client,” either a Windows, Macintosh or even possibly a Linux machine.
- ▶ You are connecting to a “server,” typically the “head” or “gateway” node of a cluster of computers. You will be working on the head node or submitting jobs to execution nodes, all of them, Linux machines.
- ▶ You can also connect to a Linux machine by using VNC to get a whole desktop if it’s supported by the server.

Connecting to a Linux Host – Windows Client Software

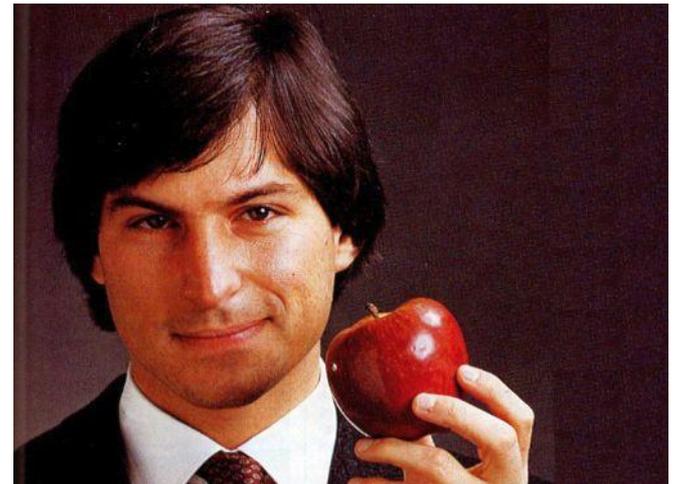
- ▶ You need a “xterm” emulation – software that emulates an “X” terminal and that connects using the “SSH” Secure Shell protocol.
 - Windows
 - If you don’t need windowing, “putty” is good:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
 - If you need windowing, use StarNet “X-Win32:”
<http://www.bu.edu/tech/desktop/site-licensed-software/xwindows/xwin32/>



Connecting to a Linux Host – Mac OS X Client Software



- Mac OS X
 - “Terminal” is already installed
 - Why? Darwin, the system on which Apple's Mac OS X is built, is a derivative of 4.4BSD-Lite2 and FreeBSD. In other words, the Mac is a Unix system!



The Shell

- ▶ A shell is a computer program that interprets the commands you type and sends them to the operating system. Secondly, it provide a programming environment consisting of environment variables.
- ▶ To determine your shell type:
 - `echo $SHELL` (shell prints contents of env)
 - `echo "$SHELL"` (shell still processes env. variable)
 - `echo '$SHELL'` (shell treats env. variable as simple literal)
- ▶ The complete environment can be printed with **set**, **setenv** (TCSH) and **set** (BASH).
- ▶ To determine the path to the shell program, type:
 - `which bash`
 - `which tcsh`
- ▶ Change the shell with "**chsh /bin/bash**" (provide path to new shell as a "parameter," meaning to be explained soon)

```
xterm
tuta0@katana:~$ echo $SHELL
/bin/bash
tuta0@katana:~$ which tesh
/bin/tesh
tuta0@katana:~$ which bash
/bin/bash
tuta0@katana:~$ chsh /bin/bash
Old password:
Your current shell already is /bin/bash.
tuta0@katana:~$ chsh /bin/tesh
Old password:
Changing shell for tuta0...
succeeded. The change will take effect within 15 minutes on all SCF systems.
tuta0@katana:~$
```

The Shell >>

Output of the echo, which and chsh commands

System Information

- ▶ After you connect, type
 - shazam
 - whoami
 - hostname
 - date
 - cal
 - free
- ▶ Commands have three parts; *command*, *options* and *parameters*. Example: `cal -j 3 1999`. “cal” is the command, “-j” is an option (or switch), “3” and “1999” are parameters.
- ▶ Options have long and short forms. Example:
 - `date -u`
 - `date --universal`

What is the nature of the prompt?

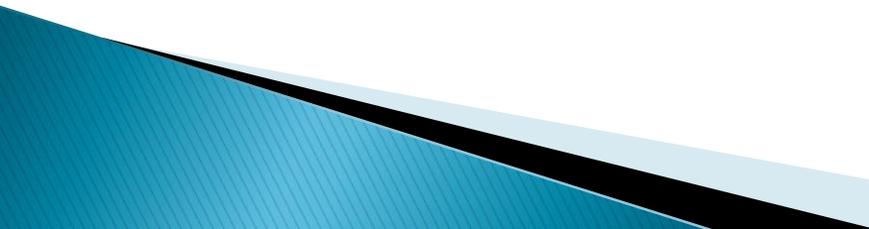
What was the system’s response to the command?

```
xterm
tuta0@katana:~> shazam
-bash: shazam: command not found
tuta0@katana:~> whoami
tuta0
tuta0@katana:~> hostname
katana
tuta0@katana:~> date
Sun Sep  9 12:14:51 EDT 2012
tuta0@katana:~> cal
    September 2012
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
tuta0@katana:~> free
              total        used         free       shared    buffers     cached
Mem:           8109704       7759508        350196           0         100204      5794856
-/+ buffers/cache:    1864448        6245256
Swap:          9244240         502656        8741584
tuta0@katana:~> date -u
Sun Sep  9 16:15:01 UTC 2012
tuta0@katana:~> date --universal
Sun Sep  9 16:15:07 UTC 2012
tuta0@katana:~> █
```

System Information >>

Output of the whoami, hostname, date, cal and free

Command History and Simple Command Line Editing

- ▶ Try the **history** command
 - ▶ Try **<Ctrl><r>** (only works in BASH shell)
 - ▶ Choose from the command history by using the up ↑ and down ↓ arrows
 - ▶ What do the left ← and right → arrow do on the command line?
 - ▶ Try the **** and **<Backspace>** keys
- 

Help with Commands

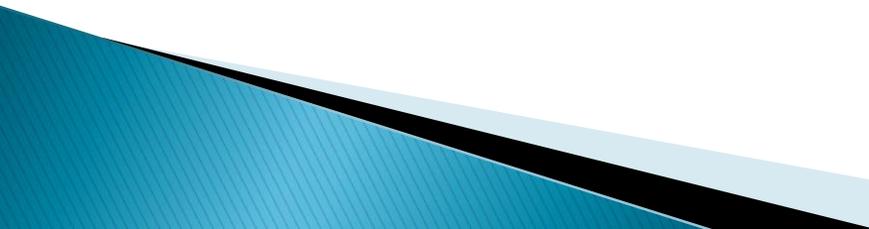
- ▶ Type
 - `hostname --help`
 - `man hostname`
 - `info hostname` (gives the same or most information, but must be paged)
- ▶ And “Yes,” you can always Google it

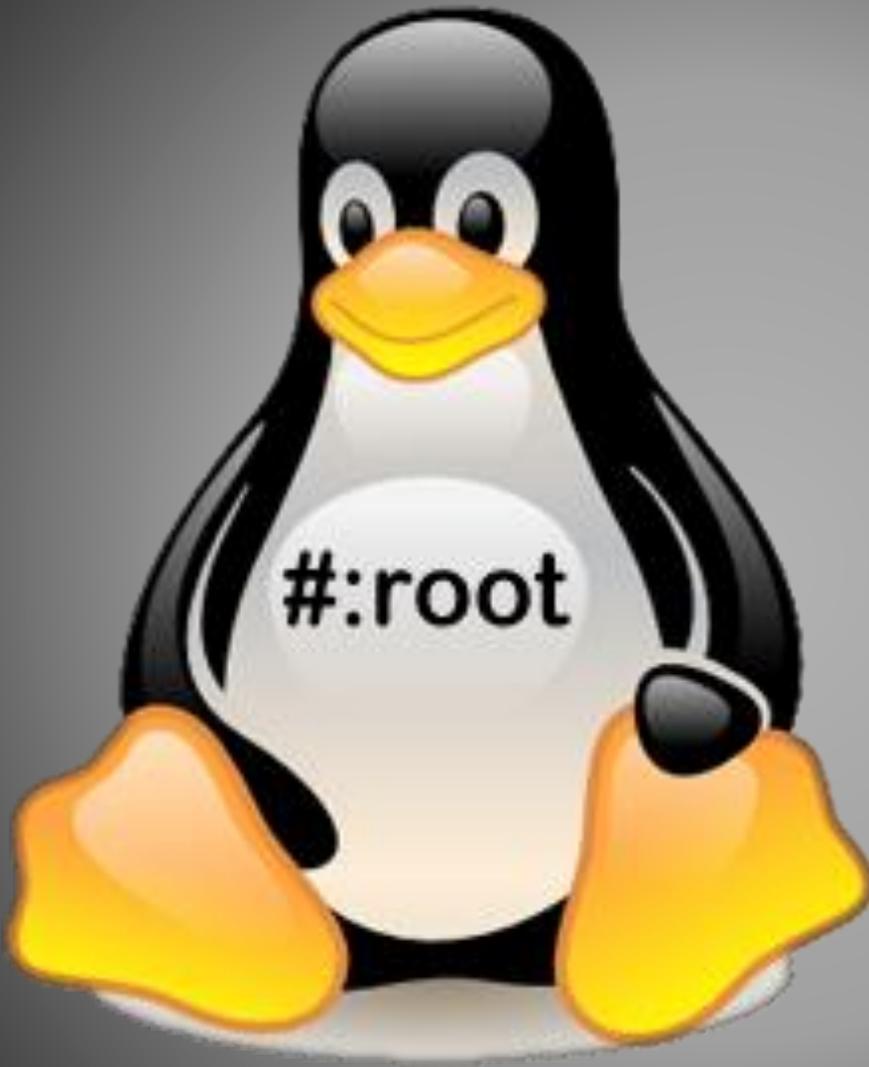


The List Command

- ▶ Useful options for the “ls” command:
 - **ls -a** List all file including hidden file beginning with a period “.”
 - **ls -ld *** List details about a directory and not its contents
 - **ls -F** Put an indicator character at the end of each name
 - **ls -l** Simple long listing
 - **ls -lh** Give human readable file sizes
 - **ls -lS** Sort files by file size
 - **ls -lt** Sort files by modification time

File System Ownership and Permissions

- ▶ All files and directories have a individual and a group *ownership*.
 - ▶ All files and directories have read (r), write (w), and execute (x) *permissions* assigned as octets to the individual owner (u), the group (g) owner and all others (o) that are logged into the system.
 - ▶ You can change permissions if you are the individual owner or a member of the group.
 - ▶ Only root can change ownership.
- 



root >>

The root user is the master

File and Directory Ownership and Permissions

- ▶ Try
 - `cd`
 - `touch myfile` (create file)
 - `mkdir mydir` (create directory)
 - `ls -l myfile` (examine file)
 - `ls -ld mydir` (examine directory)
 - `chmod g+w myfile` (add group write permission)
 - `ls -l myfile`
 - `chmod ugo+x myfile` (add user, group and other execute permission)
 - `ls -l myfile`
 - `chmod ugo+w mydir` (add user, group and other write permission)
 - `ls -ld mydir`
 - `chmod a-w` (a=ALL, remove user, group and other write permission)