Linux Demystified

Slides prepared by:
Jeremy Cushman (jscushman@hcs.harvard.edu)
Greg Brockman (brockman@hcs.harvard.edu)
Andy Brody (abrody@hcs.harvard.edu)
What is Linux?
What is GNU/Linux?
GNU’s Not Unix
It’s recursive... get it?

The Free Software Foundation
Linux is: $0.00
Linux is:
Free Software
Free as in free speech and free beer
Available in many distributions on many platforms
Ultimate flexibility and control.
Linux General Principles
General Principles

- Less is more (no news is good news)
- Small programs that do one thing really well
- (Almost) everything is open-source
- Input and output to programs is plain-text: easy to see what programs do
Principles cont.

- Multi-user computing environment with permissions
- Everything is a file. *Everything.*
- Ctrl-Z, Ctrl-D, Ctrl-C – typical ways to get out of something
- If you don’t know, RTFM: man
Where is everything?

```
brockman@cato:/$ ls

bin/  etc/  lib/  misc/  oracle@  scratch@  tmp/
boot/  home@  local/  mnt/  proc/  shells/  usr/
cdrom@  initrd/  lost+found/  nfs/  root/  srv/  var/
dev/  initrd.img@  media/  opt/  sbin/  sys/  vmlinuz@
```
Your Home Directory
~ /
The Linux filetree is flexible.

/  .  /  . . /
symlinks
User configurations
Hidden files:
.hiddenstuff
Useful programs
Useful Linux Programs

- finger, write
- find, which, whereis
- grep
- ps, kill, killall, top
- jobs, fg, screen
- quota, du, df
- ln
- dig -trace
- ping, wget, curl
- emacs, vim
- echo, cat
- head, tail, less, more
- chown, chmod
Package Manager

- Can install software from centralized repositories
- `apt-get install <package>`
- Want a webserver?
  - `apt-get install apache2`
- Want Open Office?
  - `apt-get install openoffice.org`
So how do I do anything useful?

PIPE

stdin  stdout/stderr
Commands & Pipes

- last | less
- find ./ -name "Thumbs.db" -delete
- fortune | cowsay

< It's all in the mind, ya know. >

```
\ ^__^  
\ (oo)\_____
(____)
\ ||----w |
\ ||     |
```
Getting to know your shell

tcsh  bash  zsh
Environment variables

- Control the characteristics of the shell
  - View them with `set env`, or `$VARIABLE`
  - Set them with `export`
  - Change up your prompt! `export PS1="myCOOLprompt: "`

- But these have to be declared every time you use your shell.
  - Solution: `~/.profile`, `~/.bash_profile`, etc.
But if we have variables...

- And we have all these nifty little programs that can be strung together...
- Can we make our own programs?
- YES.
- Linux is beautiful.
Shell Scripts

- Shell scripts are “programs” that are completely uncompiled, but read and executed by the shell line by line.
- Typically end in .sh
- Must be chmod’ed executable.
- Start with a “shebang” – tells the shell what to use to interpret it. e.g.,
  - `#!/bin/bash` for a bash script.
Quick overview of BASH scripting

- Easy hello world program:

- `#!/bin/bash`
  `echo "Hello World"`
BASH

```bash
#!/bin/bash

number=3
name="bob"

echo "$name is your chosen name, $number your chosen number."

let "inc=number+1"
if [ "$inc" -eq "4" ]
then echo "Addition works like a charm."
fi
```

C

```c
#include <stdio.h>
#include <cs50.h>

int number = 3;
string name = "bob";
printf("%s is your chosen name, %d your chosen number.\n", name, number, name);

int inc = number++;
if ( inc == 4 ) {
    printf("Addition works like a charm.\n");
}
```
BASH vs. C

- All variables are strings
- Variables are accessed with $VAR
- Runs other Linux programs to do its work
- Spacing usually matters.
- No line endings

- Multiple types, must be declared
- Variables do not have prefixes
- Runs subroutines or functions from libraries to do work
- Spacing matters a lot less.
- Lines end in ;
As you can see there are many similarities...

BASH is a programming language in and of itself.

You put all the little pieces of Linux together in the ways that suit you best. It’s your computer to control.
#!/bin/bash

event='-e close_write'
inotifywait -mrq --format '%w %f' $event $exclude $1 | \
while read path file; do
    echo "$(date '+%F %R') sent ${path}${file} $2"
    rsync -CR ${path}${file} $2 || echo "ERROR"
    done

- Bash makes for convenient glue code
Focus on command line, a GUI is often secondary
Trinity uses Linux.
Starting nmap V. 2.54BETA25
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
Interesting ports on 10.2.2.2:
(The 1539 ports scanned but not shown below are in state: closed)
<table>
<thead>
<tr>
<th>Port</th>
<th>State</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>22/tcp</td>
<td>open</td>
<td>ssh</td>
</tr>
</tbody>
</table>

No exact OS matches for host

Nmap run completed -- 1 IP address (1 host up) scanned
# sshnuke 10.2.2.2 -rootpw="Z1ON0101"
Connecting to 10.2.2.2:ssh ... successful.
Attempting to exploit SSHv1 CRC32 ... successful.
Resetting root password to "Z1ON0101".
System open: Access Level <9>
# ssh 10.2.2.2 -1 root
root@10.2.2.2's password:

<pre> CONTROL</pre> disable grid nodes 21 - 48
Try doing that by point-and-click... oh wait.
got graphix?

- So far we’ve been staring a lot at text consoles.
- Linux does allow for the display of graphics:
  - X11 on nice – demo it!
  - Window managers: Gnome, KDE
  - You can see these on the SC Lab computers, or in most desktop Linux distros, e.g., Ubuntu, Suse, Red Hat, etc.
Cool programs with graphics

- Firefox
- Wireshark
- VNC
- XiaoS
Anatomy of a Distribution

Building an OS with Linux is complex!

Ubuntu

GUI
- Gnome
- KDE
- X.org

CLI
- GNU coreutils
- bash
- GCC
- other libraries
- GNU C Library
- Linux Kernel
Try it out!

- You can try Linux in a VM, with a LiveCD, or by installing alongside your current OS (dual booting).
- The internets are your friend: there are lots of forums and email lists.
- See Wikipedia for history, etc.
- Read the man pages for details on any particular command (`man ls`