NSRC

Network Startup Resource Center

Linux System Administration

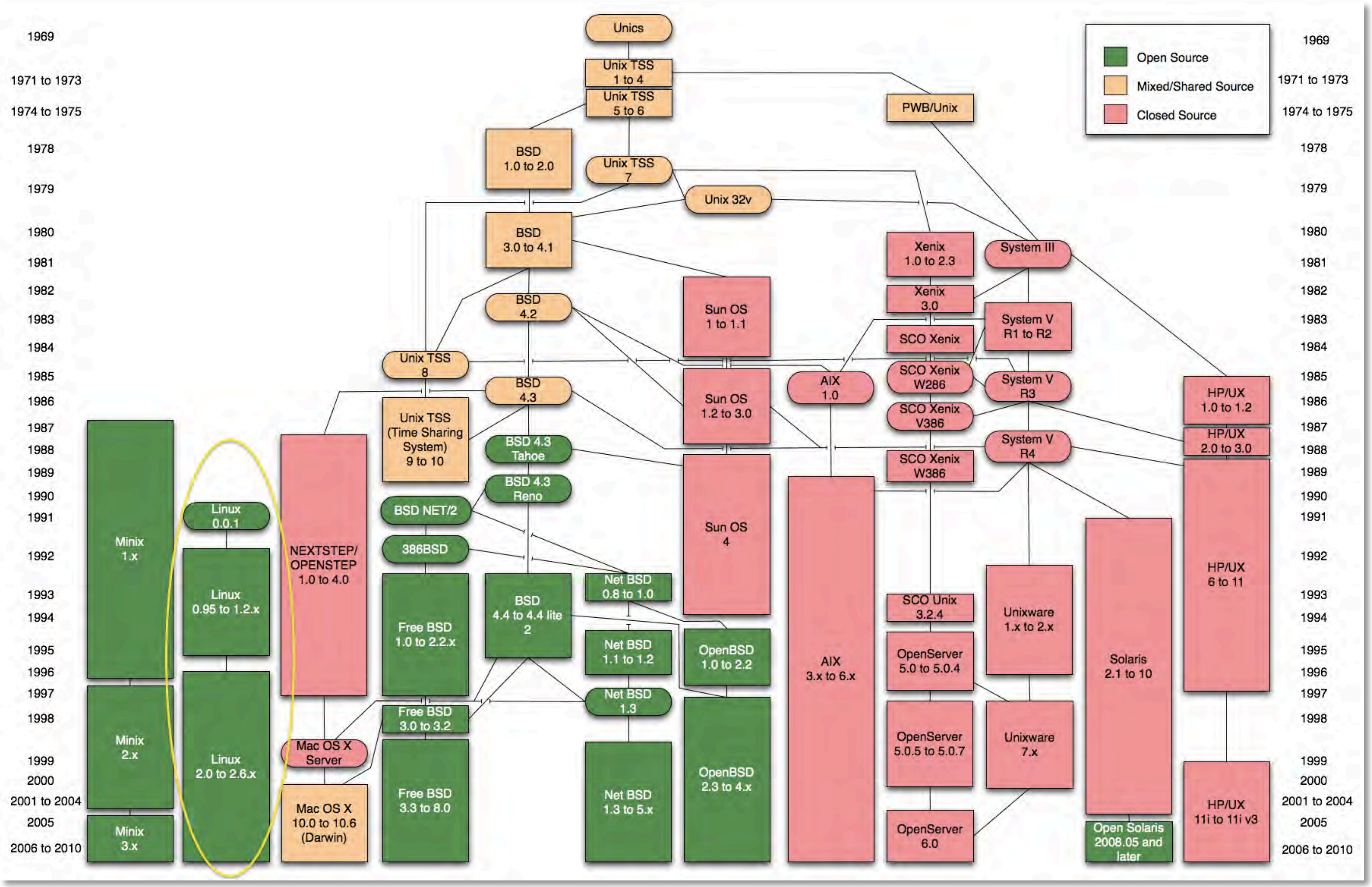
Getting started with Linux

Day 1: Modules

1. Linux overview
2. Command Line Interface or the “CLI”
3. Permissions
4. Editors
5. Ubuntu Linux and more commands

Module 1: Linux Overview

UNIX History



Unix vs. Linux

- Are they the same?
  - Yes, at least in terms of operating system interfaces
  - Linux was developed independently from Unix
  - Unix is much older (1969 vs. 1991)
- Scalability and reliability
  - Both scale very well and work well under heavy load (this is an understatement🙂)
- Flexibility
  - Both emphasize small, interchangeable components
- Manageability
  - Remote logins rather than GUI
  - Scripting is integral
- Security
  - Due to modular design has a reasonable security model
  - Linux and its applications are not without blame

Is free software really any good?!

- The people who write it also use it
- Source code is visible to all
  - The quality of their work reflects on the author personally
  - Others can spot errors and make improvements
- What about support?
  - documentation can be good, or not so good
  - mailing lists; search the archives first
  - if you show you've invested time in trying to solve a problem, others will likely help you
  - http://www.catb.org/~esr/faqs/smart-questions.html

Is free software really any good?!

Core Internet services run on free software

- BIND Domain Name Server
- Apache web server (secure SSL as well)
- Sendmail, Postfix, Exim for SMTP/POP/IMAP
- MySQL and PostgreSQL databases
- PHP, PERL, Python, Ruby, C languages

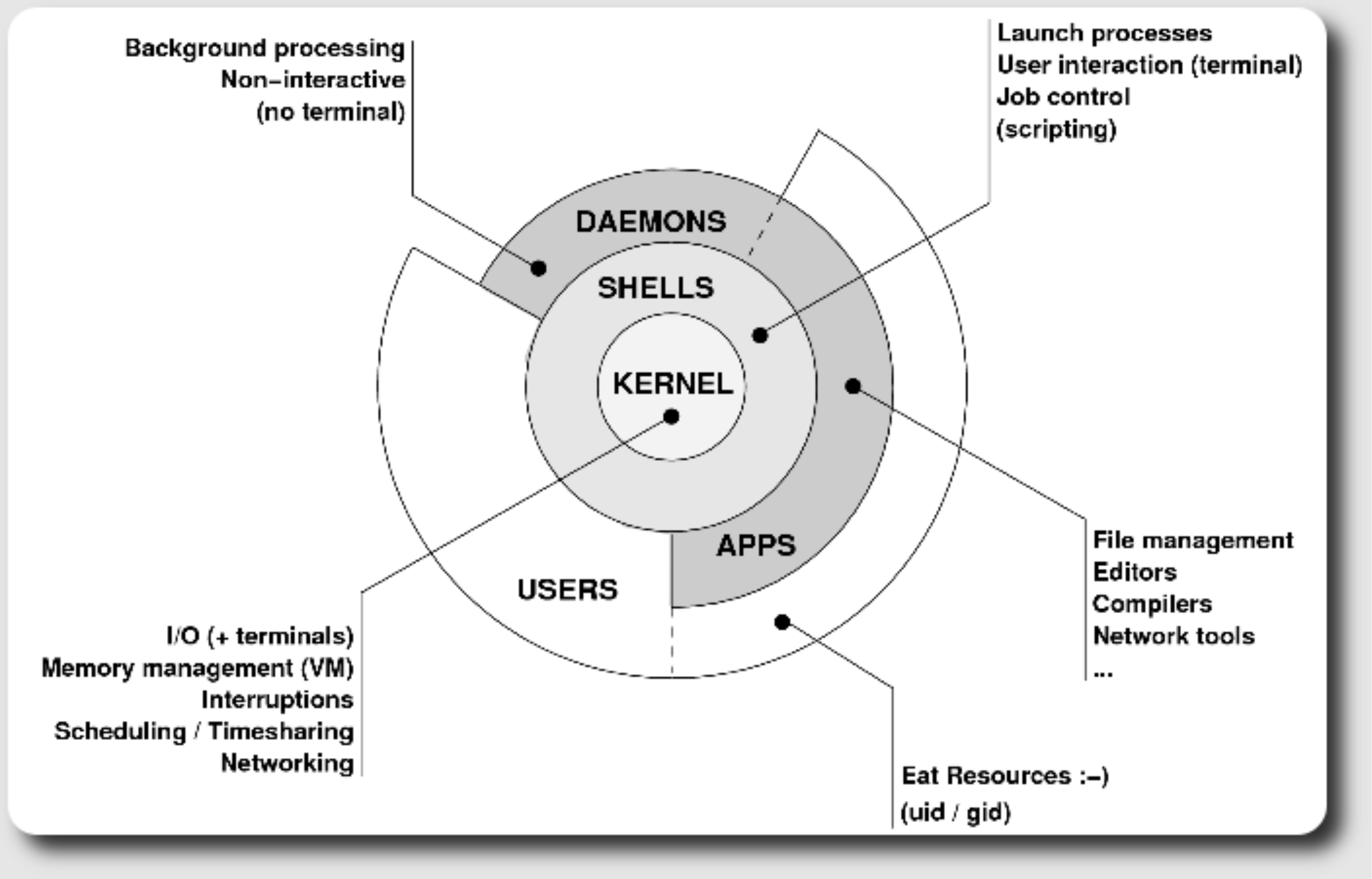
Several very high profile end-user projects

- Firefox, original Netscape browser
- OpenOffice / LibreOffice
- Thunderbird, Gimp, Gnome and KDE Desktops
- Ubuntu

What’s running Linux?

- 90% of the supercomputer TOP500, including all TOP 10
- Half of the world's 10 most reliable hosting companies
- The Internet of Things (to some extent)
- Maybe your smart phone?
  Android is *based on* Linux

The Unix/Linux System



Background processing
Non–interactive
(no terminal)

Launch processes
User interaction (terminal)
Job control
(scripting)

File management
Editors
Compilers
Network tools
...

Eat Resources :–)
(uid / gid)

I/O (+ terminals)
Memory management (VM)
Interruptions
Scheduling / Timesharing
Networking

Kernel

The "core" of the operating system

Device drivers

- communicate with your hardware
- block devices, character devices, network devices, pseudo devices (/dev/null)

Filesystems

- organize block devices into files and directories

Memory management

Timeslicing (multitasking)

Networking stacks - esp. TCP/IP

Enforces security model

Shells

Command line interface for executing programs

- Windows equivalent: command.com or command.exe

Also programming languages for scripting

- DOS/Windows equivalent: batch files, WSF, VBScript

Choice of similar but slightly different shells

- **sh:** the "Bourne Shell". Standardised in POSIX
- **csh:** the "C Shell". Not standard, but includes command history
- **bash:** the "Bourne-Again Shell". Combines POSIX standard with command history.
- Others: **ksh**, **tcsh**, **zsh**

User processes

- The programs that you choose to run
- Frequently-used programs tend to have short cryptic names
  - "**`ls`**" = list files
  - "**`cp`**" = copy file
  - "**`rm`**" = remove (delete) file
- Lots of stuff included in most base systems
  - editors, compilers, system admin tools
- Lots more stuff available to install too
  - Using the Debian/Ubuntu repositories

System processes

Programs that run in the background; also known as "daemons" ==> *

<u>Examples:</u>

- **cron**: executes programs at certain times of day
- **syslog<u>d</u>**: takes log messages and writes them to files
- **inet<u>d</u>**: accepts incoming TCP/IP connections and starts programs for each one
- **ssh<u>d</u>**: accepts incoming logins
- **sendmail** (or other MTA daemon like Postfix): accepts incoming mail

* ”Sparky” from the FreeBSD world

Security model

Numeric IDs

- user id (uid 0 = "*root*", the superuser)
- group id
- supplementary groups

Mapped to names

- /etc/passwd, /etc/group (plain text files)

Suitable security rules enforced

- e.g. you cannot kill a process running as a different user, unless you are "*root*"

Filesystem security

- Each file and directory has three sets of permissions
  - For the file's uid (user)
  - For the file's gid (group)
  - For everyone else (other)
- Each set of permissions has three bits: **rwx**
  - File: **r**=read, **w**=write, **x**=execute
  - Directory: **r**=list directory contents, **w**=create/delete files within this directory, **x**=enter directory (e**x**ecutable)

Filesystem security

The permission flags are read as follows left to right:

`-rw-r--r--`	for regular files,
`drwxr-xr-x`	for directories

We will see permissions in detail later in the day.

Any questions?

?

Standard filesystem layout

`/bin`	**essential binaries**
`/boot`	**kernel and boot support**
`/dev`	**device access nodes**
`/proc`	**pseudo-filesystem with config/system info**
`/etc`	**configuration data**
`/etc/default`	**package startup defaults**
`/etc/init.d`	**startup scripts**
/home/*username*	**user's “home” directory**
`/lib`	**essential libraries**
`/sbin`	**essential sysadmin tools**
`/tmp`	**temporary files**
`/usr`	**programs & appl. data**
`/var`	**changing files (logs, E-mail messages, queues, …)**

Don't confuse the the “root account” (/root) with the “root” (“/”) partition.

More filesystem details

/usr	
/usr/bin	binaries
/usr/lib	libraries
/usr/sbin	sysadmin binaries
/usr/share	misc application data
/usr/src	kernel source code
/usr/local/...	3rd party applications not installed with apt
/var	
/var/log	log files
/var/mail	mailboxes
/var/run	process status
/var/spool	queue data files
/var/tmp	temporary files

Log files (a few examples)

```
/var
  /var/log
  /var/log/apache2
  /var/log/apache2/access.log
  /var/log/apache3/error.log
  /var/log/auth.log
  /var/log/boot.log
  /var/log/dmesg
  /var/log/kern.log
  /var/log/mail.info
  /var/log/mail.err
  /var/log/mail.log
  /var/log/messages
  /var/log/mysql
  /var/log/syslog
```

Log file: who & what’s doing what

The most critical place to solve problems

- System messages, including:
  - Problems
  - Security issues
  - Configuration errors
  - Access issues
- Service messages, including:
  - Same as above

When something does not work…

…Look in your log files first!

Partitioning considerations

- Single large partition or multiple?
- A single partition is flexible, but a rogue program can fill it up…
- Multiple partitions provides a more “protected” approach, but you may need to resize later, on older filesystems, or without a “Volume Manager”
  - Is ***/var*** big enough? /tmp?
  - How much *swap* should you define?

Note...

- Partitioning is just a logical division
- If your hard drive dies, most likely *everything* will be lost.
- If you want data security, then you need to set up mirroring or RAID with a separate drive.

Remember, “`rm –rf /`” on a mirror will erase everything on both disks ☺

Data Security <==> Backup

/dev

Virtual files pointing to hardware or other

/dev/sda = the first harddisk (SCSI/SATA/SAS or IDE)

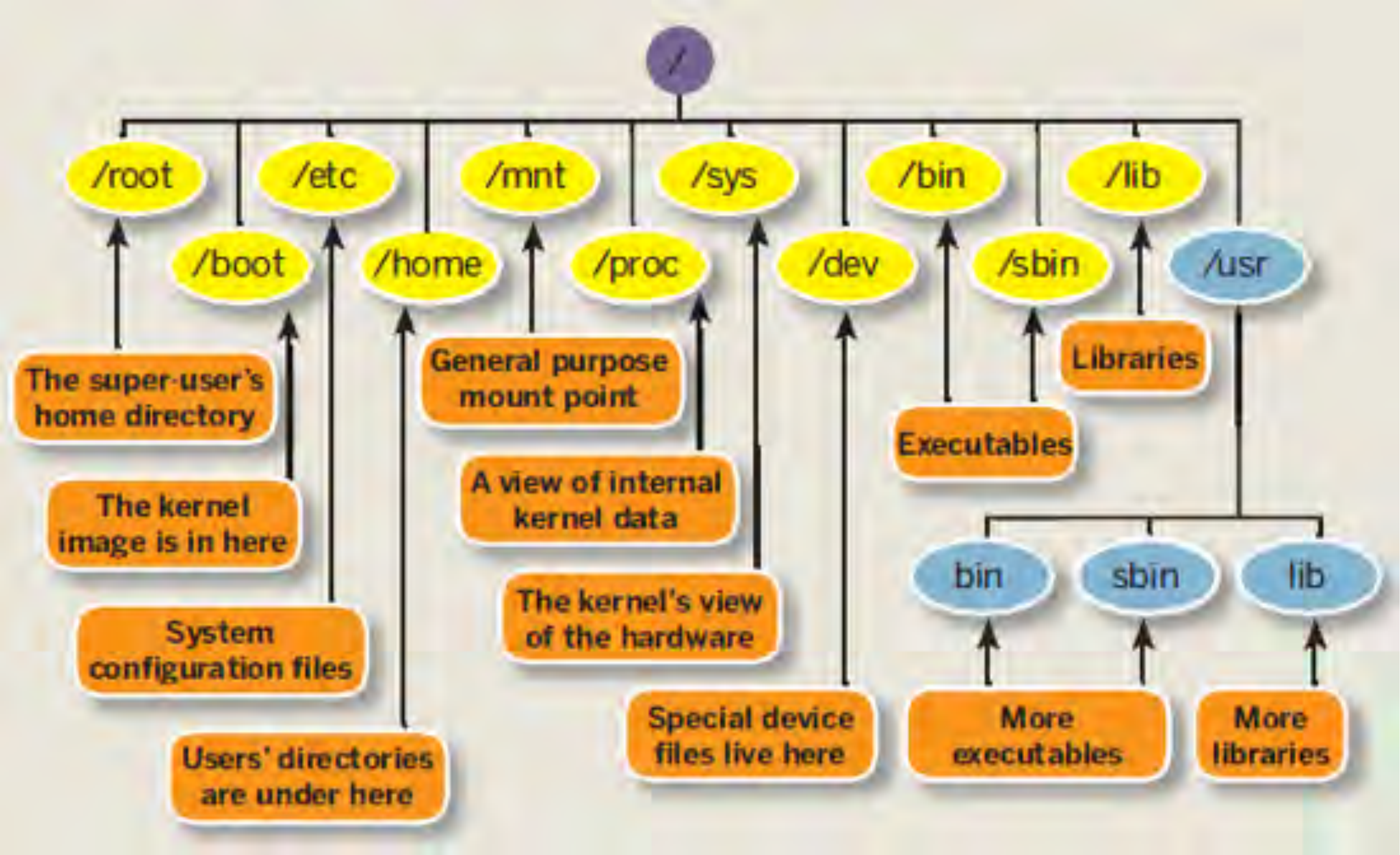
Dynamically created /dev entries

- e.g. when you plug in a new USB device

pseudo-devices:

- /dev/null
- /dev/random

Sample Linux File System


/
/root
/etc
/mnt
/sys
/bin
/lib
/boot
/home
/proc
/dev
/sbin
/usr
The super-user's home directory
The kernel image is in here
System configuration files
Users' directories are under here
General purpose mount point
A view of internal kernel data
The kernel's view of the hardware
Special device files live here
Executables
Libraries
bin
sbin
lib
More executables
More libraries

http://www.tuxradar.com/content/take-linux-filesystem-tour

How Does Linux boot?

- The *BIOS* loads and runs the *MBR*:
  - The ***Master Boot Record*** points to a default partition, or lets you select the boot partition

- MBR code then loads the boot loader, such as GRUB
- Boot loader reads configuration parameters (/boot) presents the user with options on how to boot system
- kernel is loaded and started, filesystems are mounted, modules are loaded
- init(8) process is started
- system daemons are started

http://en.wikipedia.org/wiki/Linux_startup_process

Any questions?

?

Packages & Exercises

We'll reinforce some of these concepts using exercises...

Right now please connect to your virtual Linux machine using SSH. Your instructor and workshop assistants will assist you with this:

- Windows ssh client available at
  http://noc.ws.nsrc.org/downloads/putty.exe
- ssh sysadm@pcX.ws.nsrc.org
  - User: *sysadm*
  - Host: *pcX.ws.nsrc.org*
- Accept the SSH key when asked
- Use password given in class
- `# exit`

Packages & Exercises

We’ll run a few commands to get started:

- `ls` (list files / directories)
- `pwd` (current working directory)
- `man man` (manual or help)
- …
- …
- …

There's More

But, hopefully enough to get us started...

Some Resources

http://www.ubuntu.com

http://ubuntuforums.org

http://www.debian.org

http://ubuntuguide.org

http://en.wikipedia.org/wiki/Debian

http://en.wikipedia.org/wiki/Ubuntu_(Linux_distribution)

GIYF (Google Is Your Friend)