Worlds Shortest Course Series

March 19, 2008 – Linux in 15 minutes

1. GNU/Linux Operating System Architecture
   - OS purpose
   - Naming debate
   - GNU project
   - User vs. Kernel Space
   - Monolithic vs. Microkernel debate
   - Kernel subsystems
   - Dynamic loading

2. Top level directory structure of an installed system.
   - Cabrillo classes

3. Flashcard quiz

Basic role of a multi-user multi-tasking operating system

users

programs

email
chat

graphics
photos

spreadsheets
word processing

compiler
editor

System
Administration

Operating System

hardware

Processor | Motherboard | Monitor | Memory | CD/DVD Drives | Network Adapters | Hard Drives | Keyboard

Basic role of a multi-user multi-tasking operating system

users

programs

email
chat

graphics
photos

spreadsheets
word processing

compiler
editor

System
Administration

Operating System

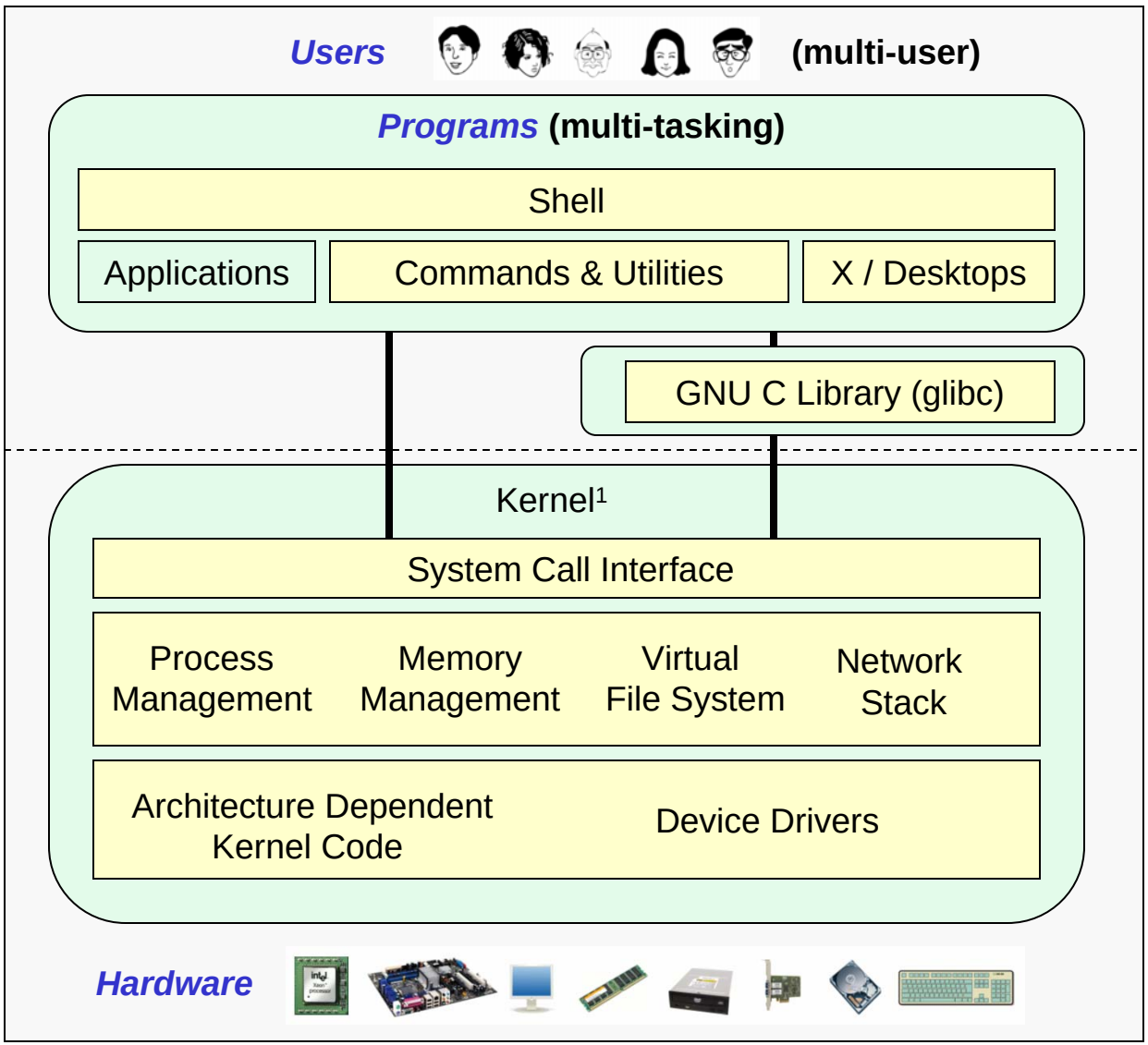
Juggles *users & programs* across limited *hardware* resources

- Runs programs for multiple users
- Provides common services for programs and users
- Shares hardware resources between competing programs and users

hardware

Processor | Motherboard | Monitor | Memory | CD/DVD Drives | Network Adapters | Hard Drives | Keyboard

GNU/Linux Operating System Architecture



Richard Stallman started the GNU project in 1983 to create a free UNIX-like OS. He Founded the Free Software Foundation in 1985. In 1989 he wrote the first version of the GNU General Public License

Linus Torvalds, as a student, initially conceived and assembled the Linux kernel in 1991. The kernel was later re-licensed under the GNU General Public License in 1992.

[1]See "Anatomy of the Linux kernel" by M. Tim Jones at http://www-128.ibm.com/developerworks/linux/library/l-linux-kernel/

User Space

Components

- Shell (in /bin)
  - Command interpreter and programming language (scripting)
- Commands and utilities (in /bin, /sbin, /usr)
  - cat, ping, ls, fdisk, chmod, man, ifconfig, ... 100's more
- X / Desktops (in /usr)
  - X window managers, gnome, kde, etc.
- GNU C Library (in /lib)
  - Math, string, input, output, logging, kernel system calls, etc.
- Applications (in /usr, /opt)
  - Browsers, word processing, spreadsheets, software development, administration, databases, web servers, etc.

Design

- Programs restricted to the privileges of the user running them
- Unlike Windows, the GUI does not run in the kernel
- Unlike Windows, multiple graphical desktops available

The Linux Kernel (in /boot)

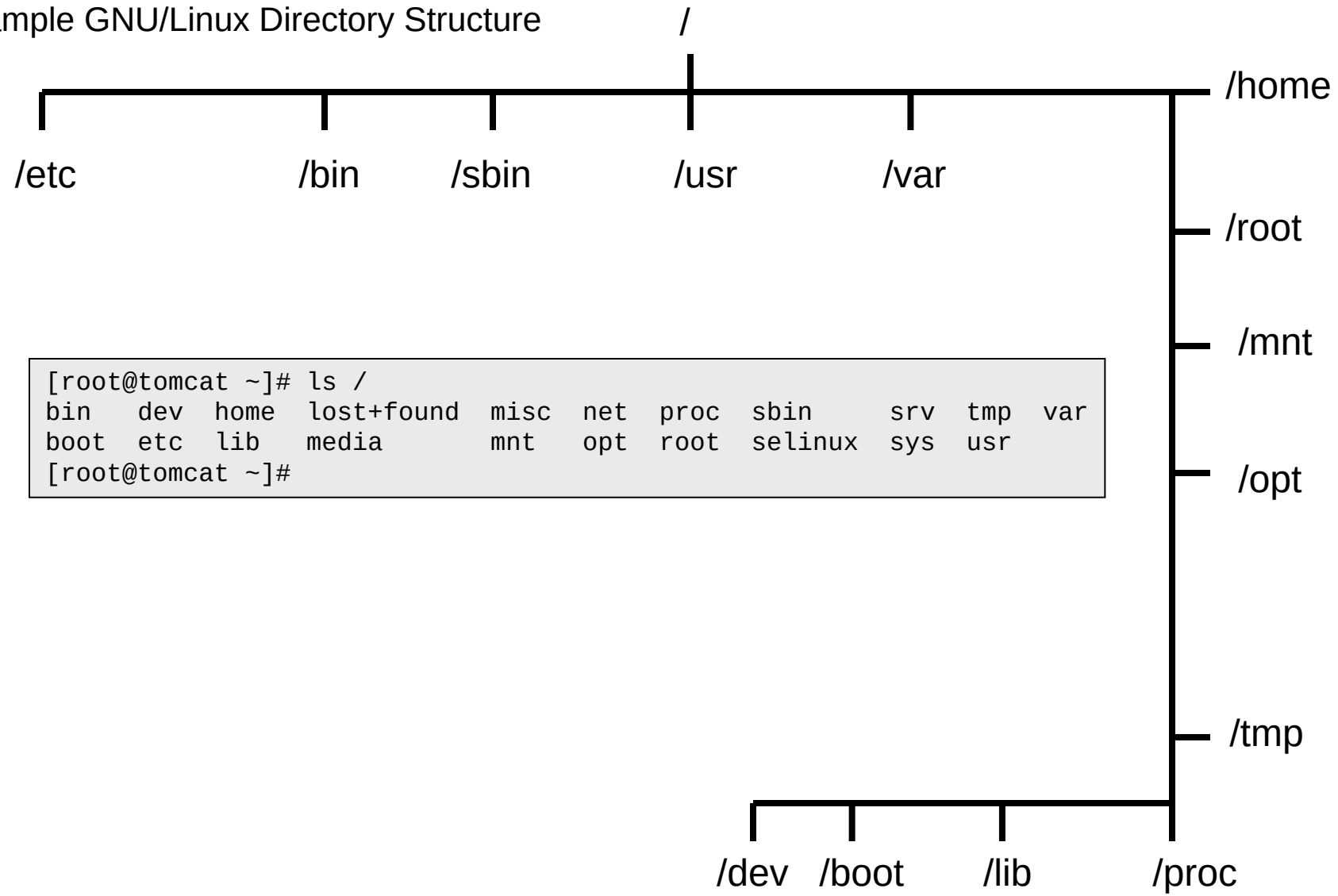
Major Subsystems:

- System Call Interface
  - mechanism for user space programs to request kernel services.
- Process Management
  - handles fork, exec, exit, kill, signals, CPU scheduling, etc.
- Memory Management
  - allocation, usage tracking, paging, etc.
- Virtual File System
  - open, close, read, write, caching, etc.
- Architecture Dependent Kernel Code
- Drivers (in /lib)

Design

- Linux kernel is “monolithic”, not a “microkernel”
- Dynamic – can load and unload modules on the fly
- Overtime has become efficient, stable and portable

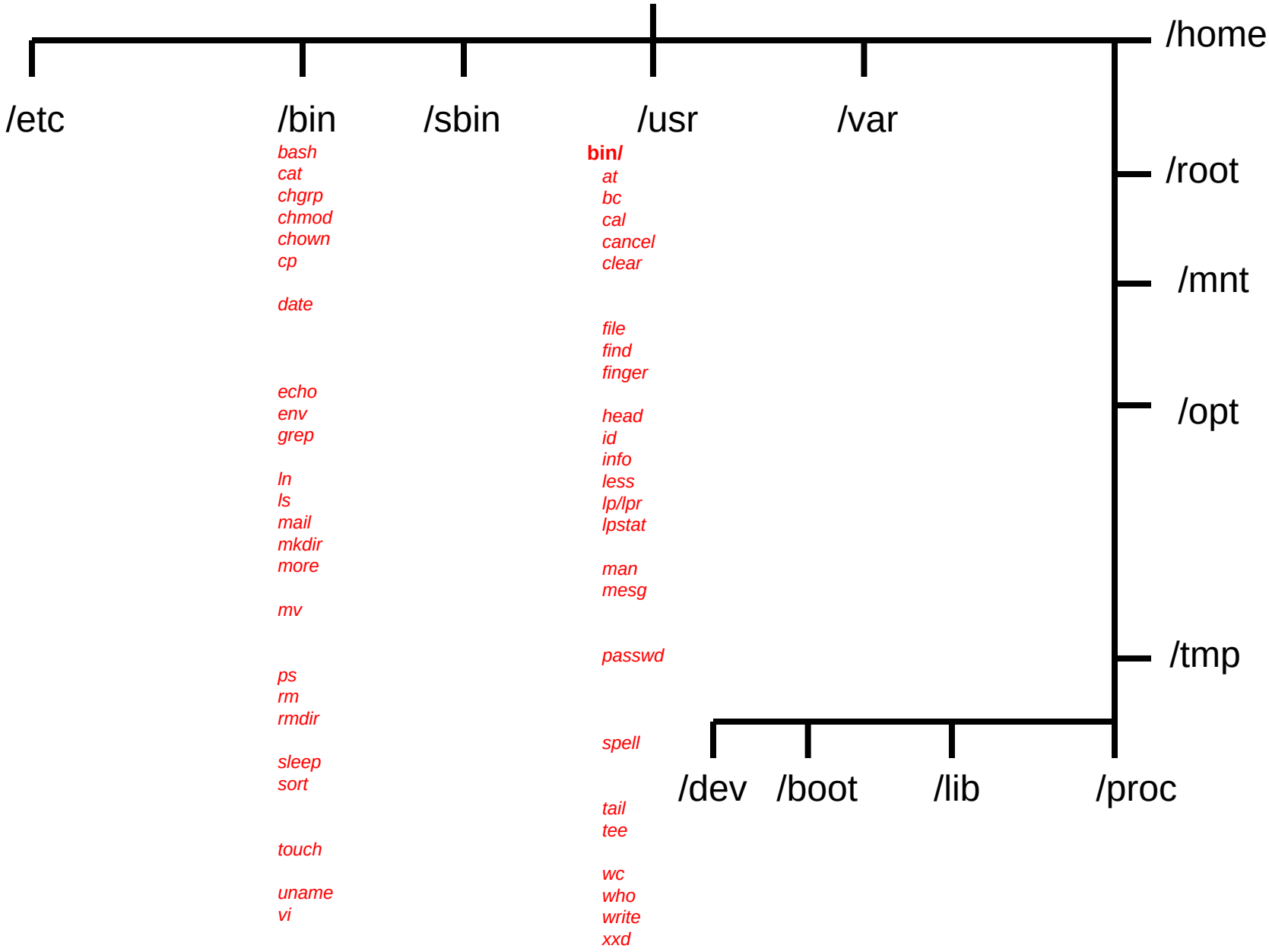
Example GNU/Linux Directory Structure




```
[root@tomcat ~]# ls /
bin   dev  home  lost+found  misc  net  proc  sbin     srv  tmp  var
boot  etc  lib   media       mnt   opt  root  selinux  sys  usr
[root@tomcat ~]#
```

Example GNU/Linux Directory Structure

CIS 90 files, directories, commands

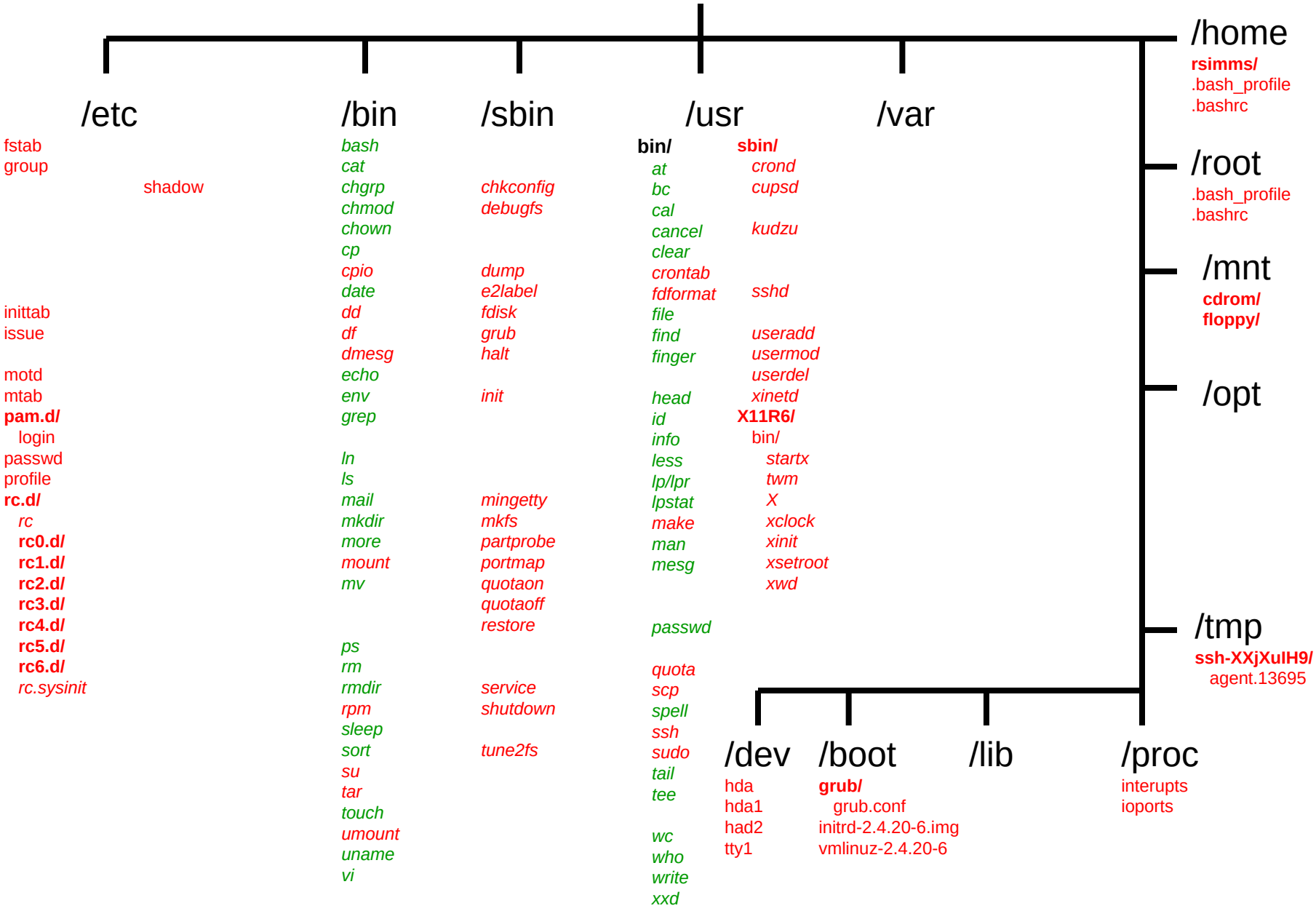


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset

Example GNU/Linux Directory Structure

/

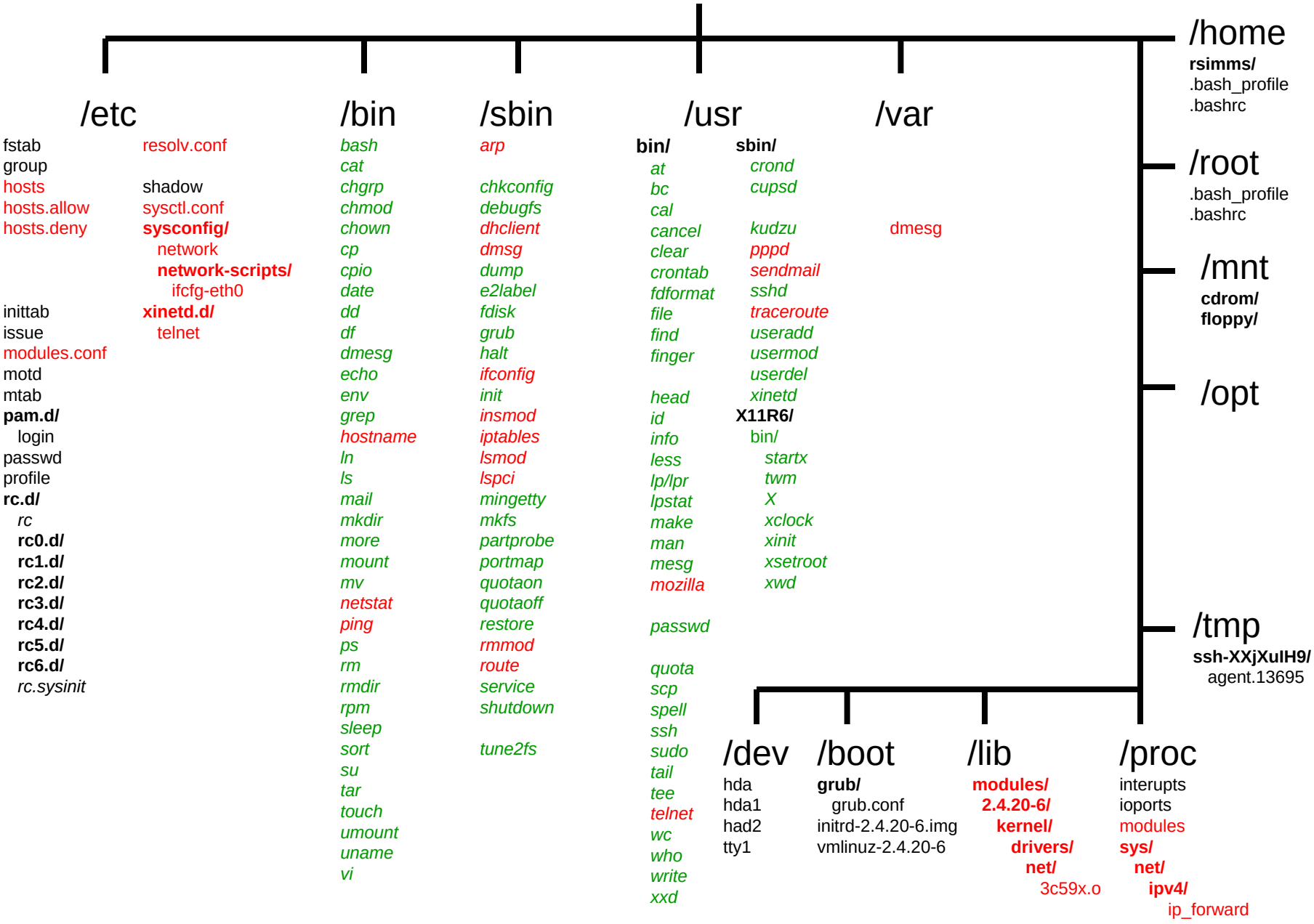
CIS 191 files, directories, commands



Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset

Example GNU/Linux Directory Structure

CIS 192 files, directories, commands

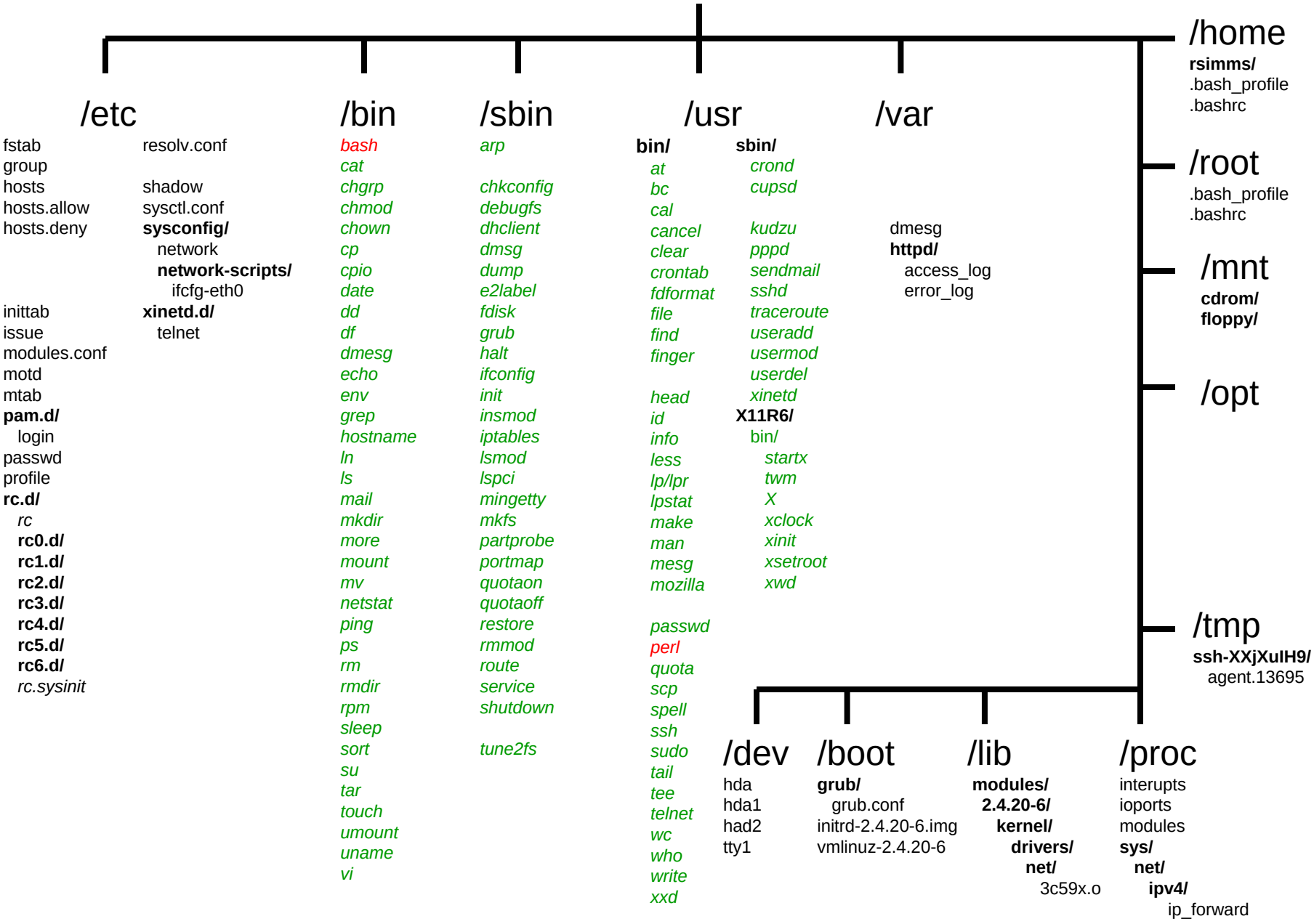


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset

Example GNU/Linux Directory Structure

/

CIS 130 files, directories, commands



/etc

fstab
group
hosts
hosts.allow
hosts.deny
inittab
issue
modules.conf
motd
mtab
pam.d/
  login
passwd
profile
rc.d/
  rc
  rc0.d/
  rc1.d/
  rc2.d/
  rc3.d/
  rc4.d/
  rc5.d/
  rc6.d/
  rc.sysinit
resolv.conf
shadow
sysctl.conf
sysconfig/
  network
  network-scripts/
    ifcfg-eth0
xinetd.d/
  telnet

/bin

bash
cat
chgrp
chmod
chown
cp
cpio
date
dd
df
dmesg
echo
env
grep
hostname
ln
ls
mail
mkdir
more
mount
mv
netstat
ping
ps
rm
rmdir
rpm
sleep
sort
su
tar
touch
umount
uname
vi

/sbin

arp

chkconfig
debugfs
dhclient
dmsg
dump
e2label
fdisk
grub
halt
ifconfig
init
insmod
iptables
lsmod
lspci
mingetty
mkfs
partprobe
portmap
quotaon
quotaoff
restore
rmmod
route
service
shutdown

tune2fs

/usr

bin/
  at
  bc
  cal
  cancel
  clear
  crontab
  fdformat
  file
  find
  finger

  head
  id
  info
  less
  lp/lpr
  lpstat
  make
  man
  mesg
  mozilla

  passwd
  perl
  quota
  scp
  spell
  ssh
  sudo
  tail
  tee
  telnet
  wc
  who
  write
  xxd

sbin/
  crond
  cupsd

  kudzu
  pppd
  sendmail
  sshd
  traceroute
  useradd
  usermod
  userdel
  xinetd

X11R6/
  bin/
    startx
    twm
    X
    xclock
    xinit
    xsetroot
    xwd

/var

dmesg
httpd/
  access_log
  error_log

/home

rsimms/
.bash_profile
.bashrc

/root

.bash_profile
.bashrc

/mnt

cdrom/
floppy/

/opt

/tmp

ssh-XXjXuIH9/
  agent.13695

/dev

hda
hda1
had2
tty1

/boot

grub/
  grub.conf
initrd-2.4.20-6.img
vmlinuz-2.4.20-6

/lib

modules/
2.4.20-6/
  kernel/
    drivers/
      net/
        3c59x.o

/proc

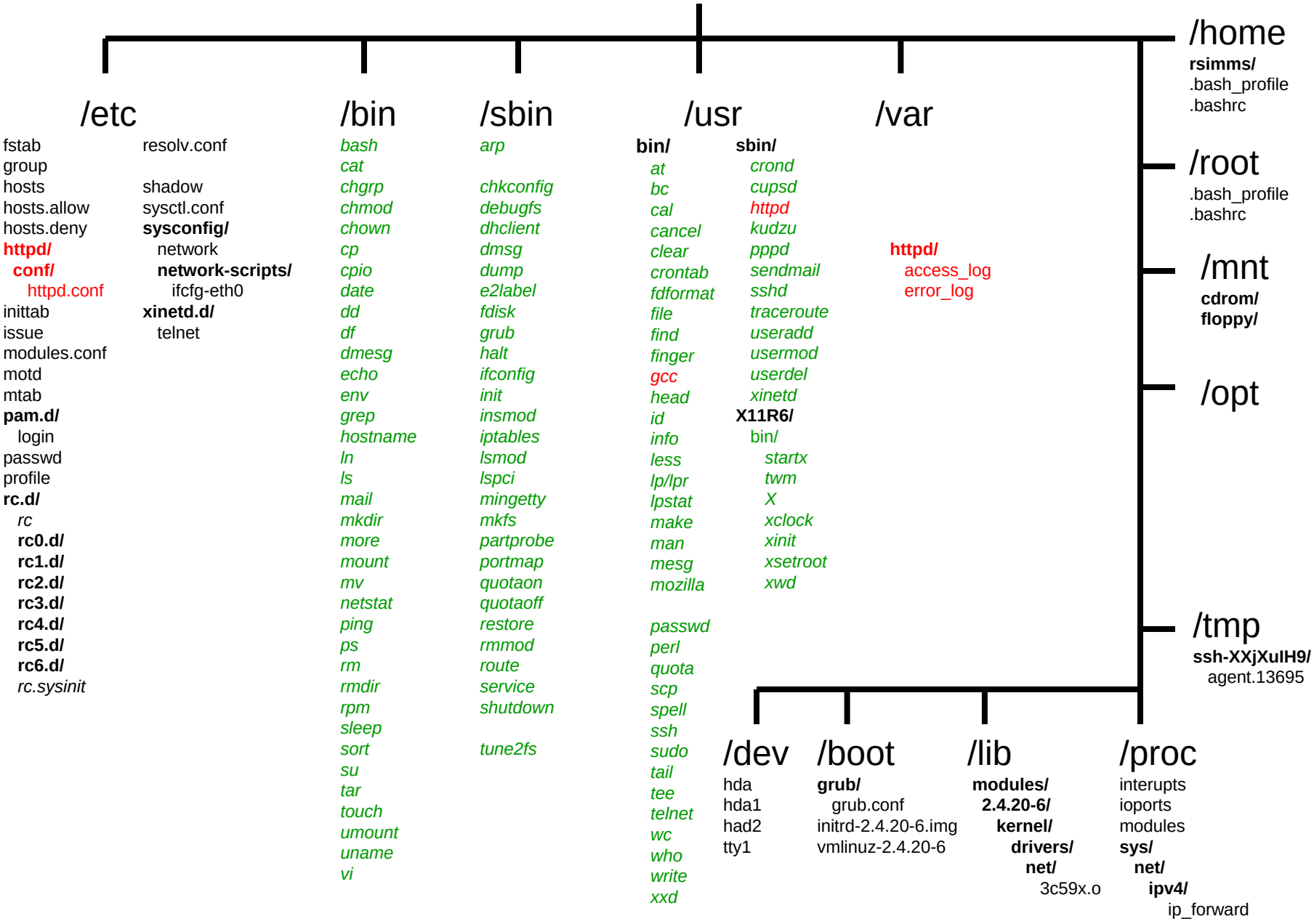
interupts
ioports
modules
sys/
  net/
    ipv4/
      ip_forward

Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

/

CIS 164 files, directories, commands

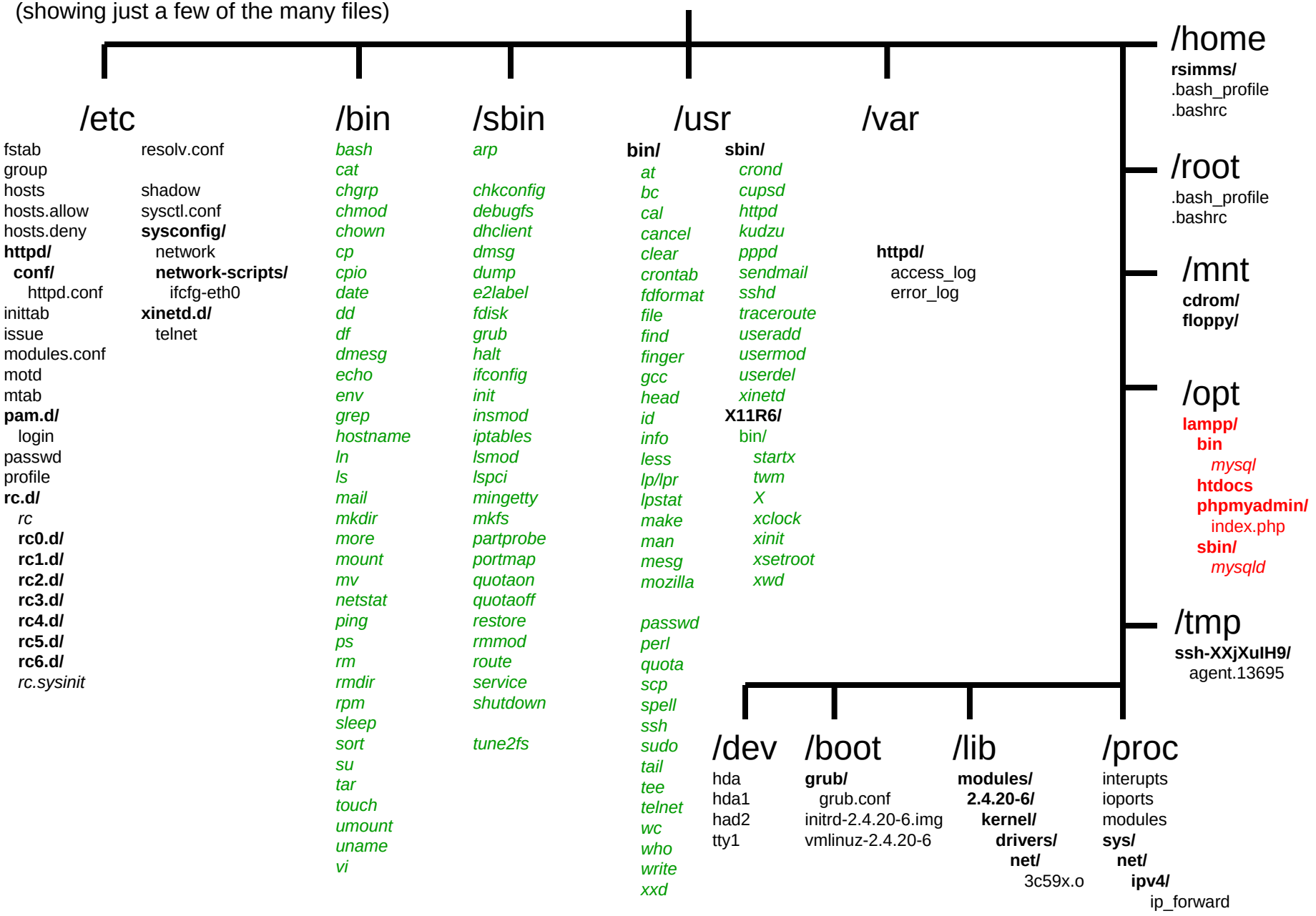


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)

CIS 165PH files, directories, commands



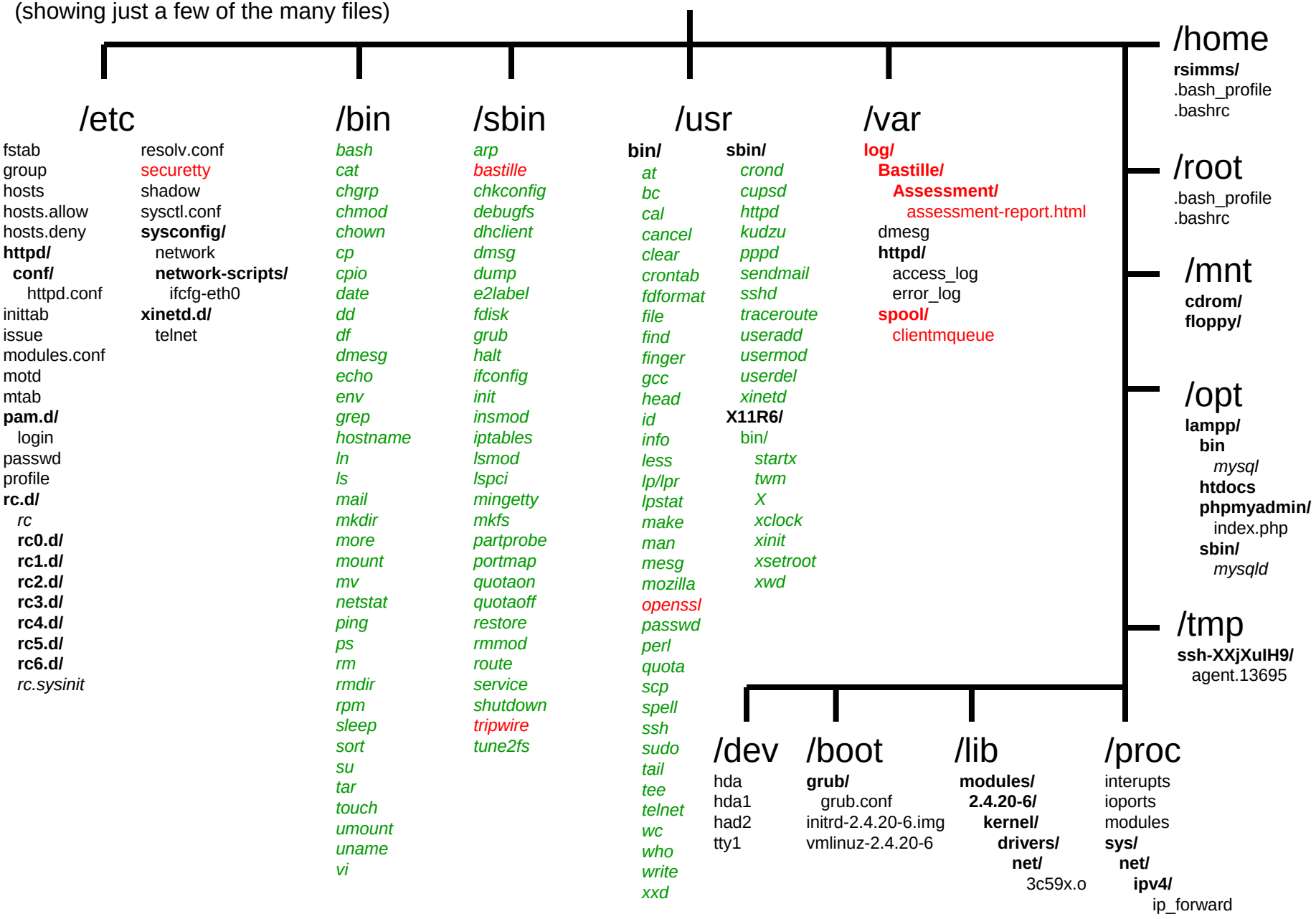
Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)

/

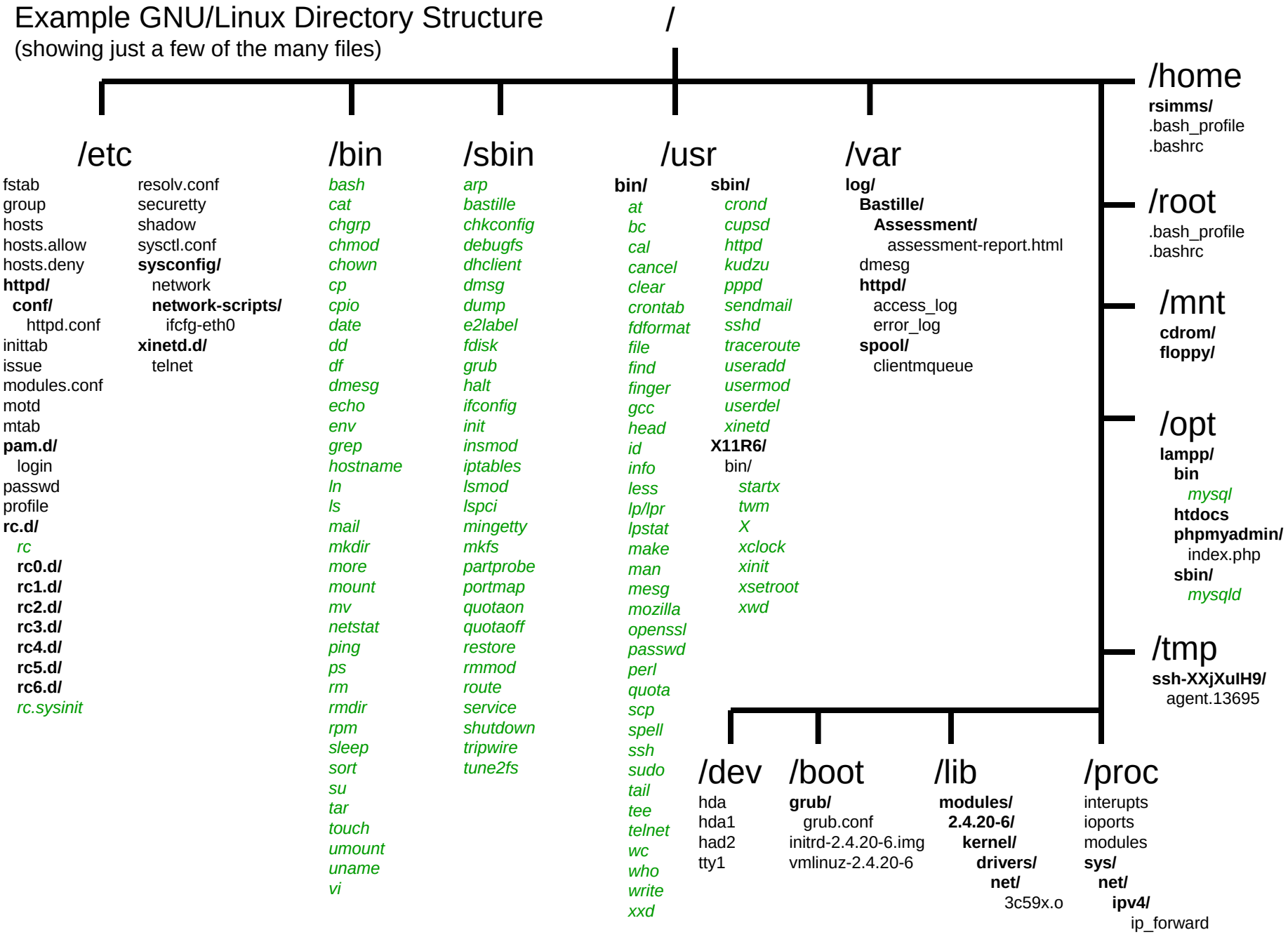
CIS 193 files, directories, commands



Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)



Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

References and optional further reading

Anatomy of the Linux kernel
By Tim Jones, IBM,
http://www.ibm.com/developerworks/linux/library/l-linux-kernel/

Kernel command using Linux system calls
By M. Tim Jones
http://www.ibm.com/developerworks/linux/library/l-system-calls/

Security Report: Windows vs Linux
By Nicolas Petreley
http://www.theregister.co.uk/security/security_report_windows_vs_linux/#windesign

The Tanenbaum-Torvalds Debate
http://www.oreilly.com/catalog/opensources/book/appa.html

GNU/Linux naming controversy
http://en.wikipedia.org/wiki/GNU/Linux_naming_controversy