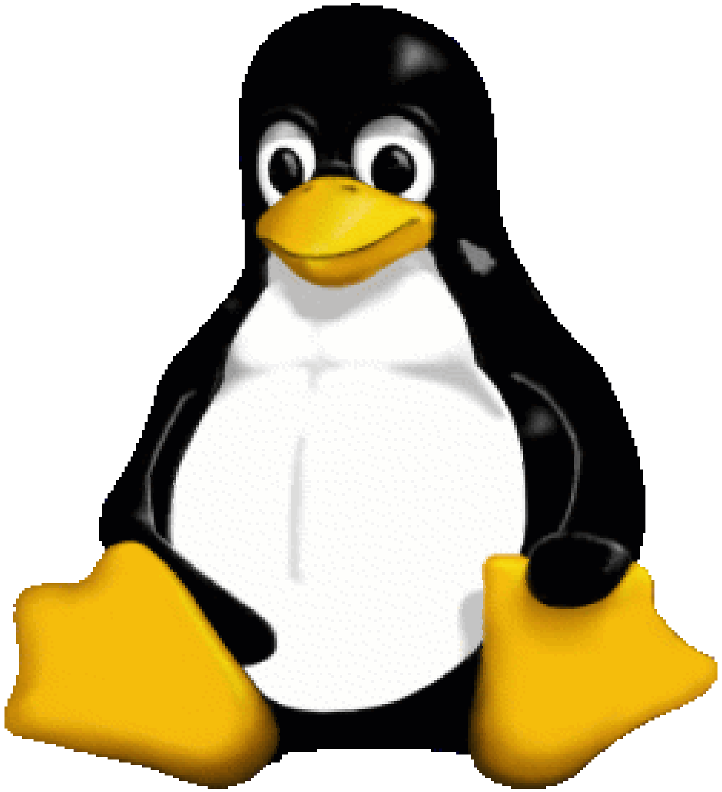
Five-Session Intensive Course on Playing with Linux

Dedicated to the FYP students of Prof. P. C. Wong



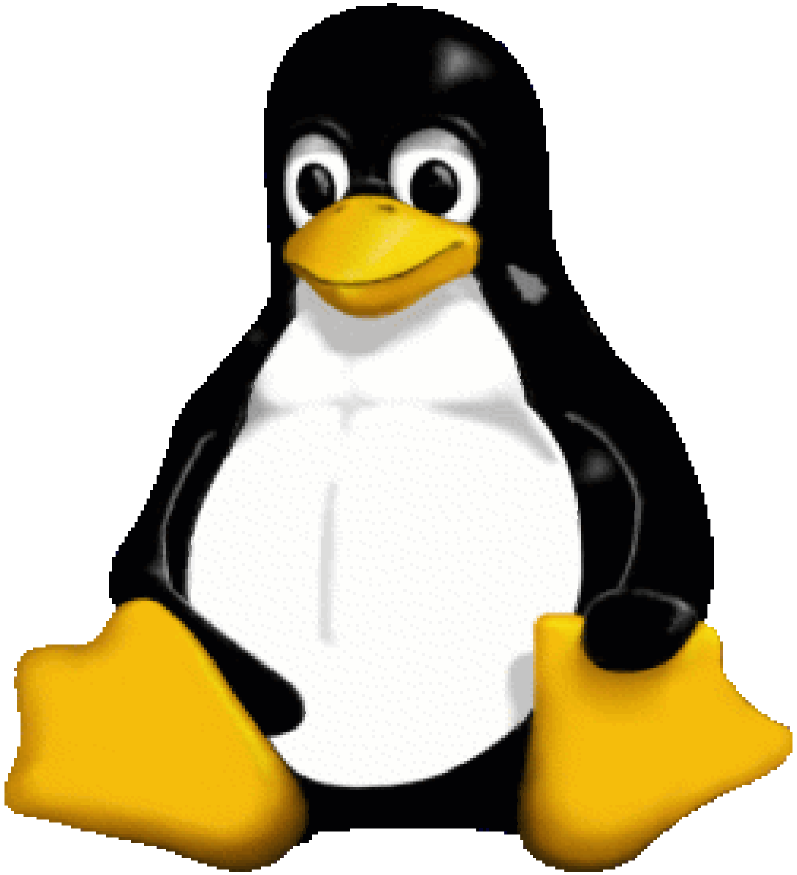
Presented by

Adrian S. W. TAM (adrian.tam@iee.org)

September 2002

The Logo of Linux

- A Penguin
- His name is Tux



Red Hat is not Linux

- Red Hat is a distribution, not a Linux
- List of well-known distributions:
  - Mandrake-Linux (Largest in US)
  - SuSE (Best reputation in Europe)
  - Debian (Official Distribution of Developers)
  - Slackware (Grandfather's memory)
  - Red Hat (Most well-known by non-Linuxians)
  - Red Flag (Let's BOYCOTT this one)
  - Gentoo (Maybe the best distro ever)
  - LFS (Real player's choice)

Other Distributions

- CD Linux
  - DemoLinux, Virtual Linux, Knoppix
- Floppy Linux
  - floppix, tomsrtbt, Tiny Linux
- Firewall / Router
  - gibalter, floppyfw, fli4l

What is Linux?

- Linux = an OS kernel
  - Locates between (app.) software and electronics
  - Coordination
  - Error handling
  - Signal handling
  - Hardware interfacing
  - Process handling
  - Security manipulation

What is Linux?

- OS needs software
  - User interface (sh, csh, bash, tcsh, ash, zsh, pdksh)
  - File manipulation (cp, rm, ln, ls, mkdir, cd, rmdir)
  - Text processing (vi, sed, awk, grep, uniq, sort)
  - Archiving (zip, tar, bzip2, gzip, cpio, dd)
  - User management (useradd, usermod, userdel)
  - Process management (ps, kill, top, nice, renice)
  - Networking (ftp, wget, telnet, ping, snort, tcpdump)
  - Programming (gcc, gmake, g++, g77, gcj, gdb)
  - Automation (cron, at, batch, perl, sh, bg, fg)
  - GUI (X, xfs, gnome, xfce, kde, mozilla, xfig, lyx)

What is Linux?

- GNU (FSF) provides software
- Linux provides kernel
- Combine = GNU/Linux = Complete OS package

What is Linux?

- GNU/Linux is not FREE!
  - Money can be charged
  - Efforts should be paid
- GNU/Linux is FREE!
  - Freedom to do everything
  - Freedom to know everything

Why Linux?

- Linux is POWERFUL
  - Inherits 40 years' experience from UNIX
  - Couples with (?) UNIX software packages
  - Uses the wonderful design of UNIX
  - Portability
  - Efficiency
  - Functionality
  - Availability
  - Reliability

Why Linux?

- Linux is not so good
  - Weak networking
  - Young
  - Not unified
  - Not guarenteed

Why Linux?

- Non-Linux UNIX alternatives:
  - FreeBSD / NetBSD / OpenBSD / BSDi
  - Caldera SCO OpenUnix
  - Sun Microsystems Solaris
  - IBM AIX
  - HP-Compaq HP-UX
  - HP-Compaq OpenVMS
  - Microsoft (?) XENIX
  - Apple MacOS X
  - GNU Hurd
  - OSF Mach

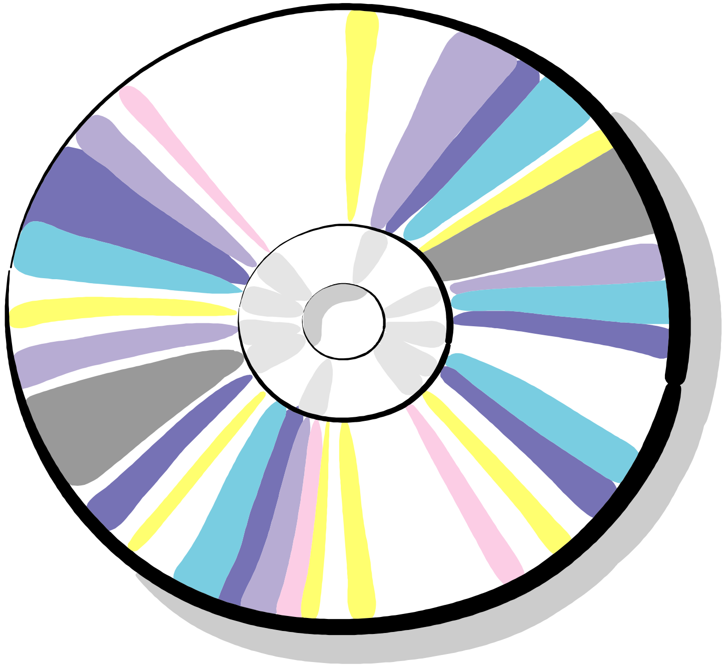
Why Linux?

- UNIX-like Non-UNIX alternatives:
    - AT&T Plan9
    - BeOS
    - AtheOS
    - NeXT
- Non-UNIX-like non-UNIX non-alternatives:
    - Apple MacOS <=9
    - Microsoft Windows
    - PalmOS
    - DOS
    - CP/M

Mottos

- Everything is a file
- Command line is wonderful
- Join the building blocks
- Man is powerful
- Google is almighty

Installation



Installing Linux

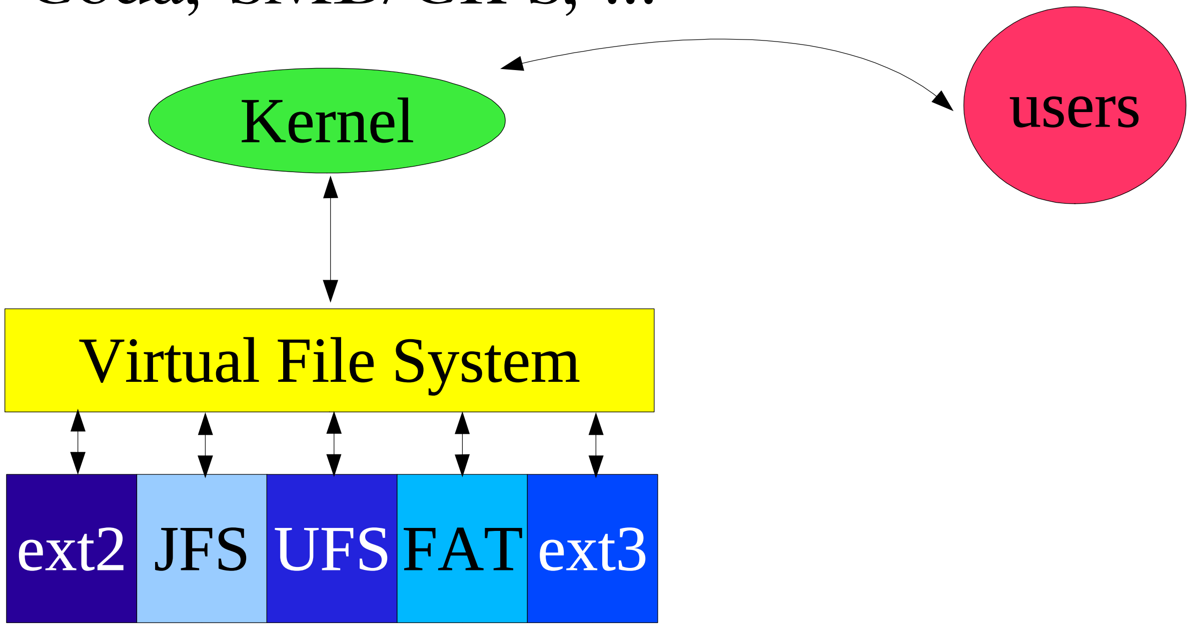
- Hard Disk Partitioning
  - IBM PC Compatibles with EIDE interface hard disk
  - 1 HD <= 20 Partitions
  - Partitions = {Primary, Extended}
  - Primary Partition <= 4
  - Extended Partition <= 16
  - U{Extended Partition} = Primary Partition 4

Installing Linux

- Hard Disk Partitioning
  - Primary Partition = /dev/hda1 to /dev/hda4
  - Extended Partition = /dev/hda5 to /dev/hda20
  - Primary master = /dev/hda
  - Primary slave = /dev/hdb
  - Secondary master = /dev/hdc
  - Secondary slave = /dev/hdd

Installing Linux

- File Systems
  - Native file system: ext2
  - Journaling: ext3 (Recommanded)
  - Other Journaling: ReiserFS, XFS, JFS
  - Other FS: NTFS, UFS, FAT, Minix, Novell, ...
  - Network: NFS, Coda, SMB/CIFS, ...



Installing Linux

- Account
- Root Account
- Group
- UID/GID
- File mode (Permissions)
- File attribute
- Process
- PID
- Signal

Installing Linux

- File
- Device
- Daemon
- Plumbing
- Foreground / Background
- Virtual Console / Terminal
- Shell
- Compilation

Installing Linux isn't difficult, but there are many details to remember
—— Running Linux, Welsh et al

Login / Logout

Login

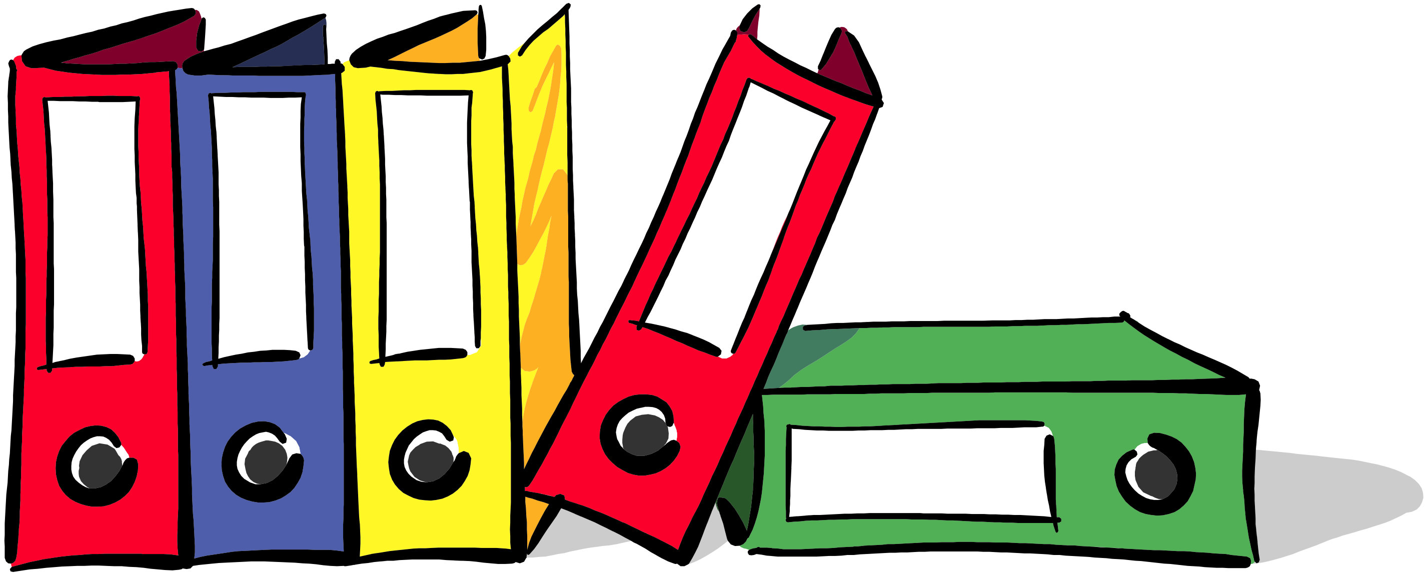
- Login

```
Debian GNU/Linux stable server1 tty1
server1 login: root
Password: xxxxx
Last login: Mon Sep 2 09:32:28 2002 on tty1
Linux server1 2.4.19 #24 Sun Aug 25 20:13:22 HKT 2002 i686 unknown unknown GNU/Linux
server1:~#
```

Logout

- exit
- shutdown -h now
- shutdown -r now

File Hierarchy



File Hierarchy

- /boot

  Boot files (kernel, System.map, boot loader)
- /bin

  Essential binary files (programs)
- /sbin

  Essential system binary files
- /dev

  Device files resides here
- /proc

  Process files resides here

File Hierarchy

- /etc

  Usually configuration files stores here
- /lib

  Dynamic linking libraries, system modules
- /tmp

  Temp dir
- /var

  Variable data (log files, caches, spools)
- /usr

  Static data (C:\Program Files\ ?)

File Hierarchy

- /root

  The home directory of root
- /home

  The home directories of other users
- /home/adrian

  The home directory of user 'adrian'

File Hierarchy

- /usr/bin: Not-so-essential binary
- /usr/sbin: Not-so-essential system binary
- /usr/lib: Not-so-essential libraries
- /usr/share: Shared data
- /usr/share/doc: Documentation
- /usr/local: Local data (user-made programs)
- /usr/local/bin: User-made binary programs
- /usr/local/sbin: User-made system binary programs

File Hierarchy

- /var/log: Log files
- /var/cache: Cache files
- /var/spool: Spools (print spool, etc.)
- /var/tmp: Temp files

File Hierarchy

```
/
|-- bin          binary executables (essential)
|-- boot         boot files
|-- dev          device file system
|-- etc          configuration files, startup scripts
|-- home         home directories of users
|   |-- adrian   home dir. of Adrian
|   |-- brian    home dir. of Brian
|   `-- carson   home dir. of Carson
|-- lib          dynamic linking libraries
|-- misc         miscellaneous (empty)
|-- mnt          mount points
|-- net          network mounts (empty)
|-- opt          optionals (empty)
|-- proc         process file system
|-- root         home dir. of root user
|-- sbin         binary executables for system admin use (essential)
|-- swap         swaps (optional)
|-- tmp          temporaries
|-- usr          (user) static data
|   |-- X11R6    X-Window
|   |-- bin      application executables
|   |-- etc
|   |-- include  C/C++ header files
|   |-- lib      C/C++ static linking libraries
|   |-- local
|   |-- man      man pages
|   |-- sbin     application executables for system admin use
|   |-- share    share files (pics, icons, ...)
|   |-- src      source
`-- var          dynamic data
```

Important Files

- /etc/X11/XF86Config: XFree86 configuration
- /etc/inittab: init table
- /etc/fstab: mount table
- /etc/passwd: password file
- /etc/group: group assignments
- /etc/crontab: table of cron jobs
- /var/log/messages: Program messages
- /var/log/syslog: System logs
- /var/log/auth.log: Authentication logs

Strange??

- No 'drive' concept
- Unified directory tree
- Different media are connected via a 'mount' process
- *BSD can use mount to enlarge a storage space! (Not possible in Linux, though)

So...re-partitioning

- Example:

/dev/hda1	500MB	Mounted at /
/dev/hda2	2 GB	Mounted at /usr
/dev/hda3	2 GB	Mounted at /var
/dev/hda4	5.5 GB	
/dev/hda5	500MB	Swap
/dev/hda6	4 GB	Mounted at /home
/dev/hda7	1 GB	Mounted at /root

Boot Loader

x86 Booting Procedure

- System startup
- Checking (CPU, RAM)
- Bootstraping all components together
- Do critical checkings (a.k.a. POST)
- Seek for peripherial devices
- Following the booting procedure to seek for OS
- Boot sector is loaded
- Control is passed on to the boot sector from the BIOS

x86 IDE Hard Disk

- First block = Partition Table
- Second block = Boot sector (A program)

Boot loader

- LILO (Linux Loader)
- Grub
- The boot loader will first do some basic job
- Then loads the OS kernel and pass the control to it
- The kernel then do several things:
  - Call the start-up scripts
  - Load user interfaces (CLI / GUI)
  - Start background jobs (daemons)

Boot loader

- /etc/lilo.conf

  Configuration file of LILO
- /boot/grub/menu.lst

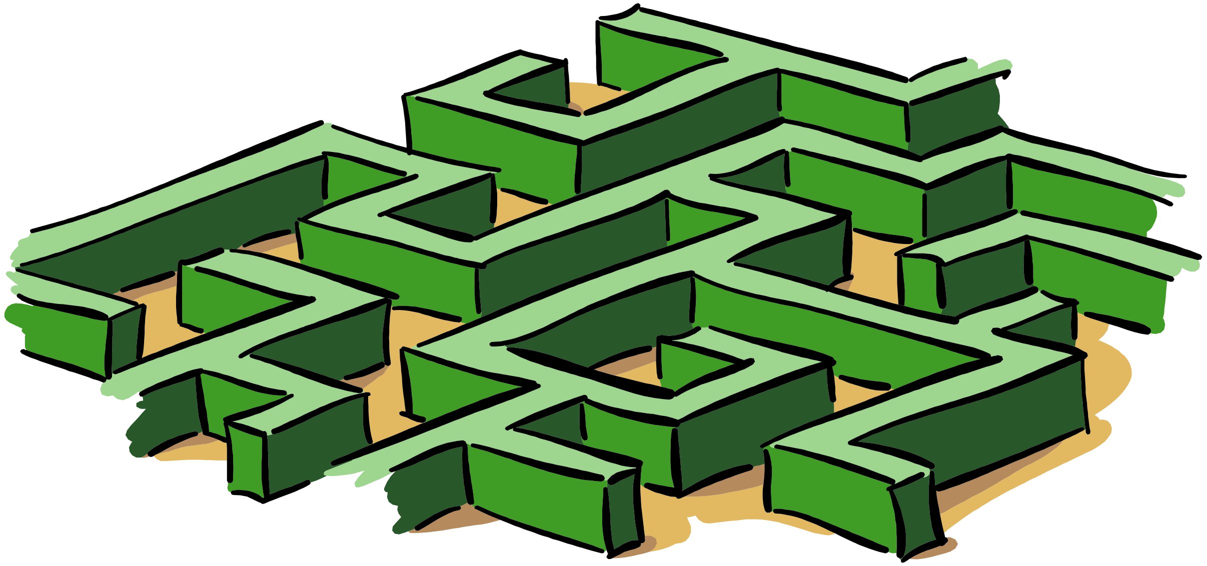
  Configuration file (menu definition) of Grub

LILO Configuration

- /etc/lilo.conf

```
lba32                  #Support >1024 cylinder
boot=/dev/hda          #Boot sector
root=/dev/hda1         #Default root partition

                       #Select boot sector: bmp/compat/menu/text
install=/boot/menu.b
map=/boot/map

delay=20               #Wait 2 second before choosing default
vga=794                #1280x1024 framebuffer display

default=Linux          #Default boot option

image=/boot/vmlinuz
      label=Linux
      read-only

other=/dev/hda2
      label=Win2000
      loader=/boot/chain.b
```

Shell Basics



Shell Basics

- Root Prompt: #
- User Prompt: $

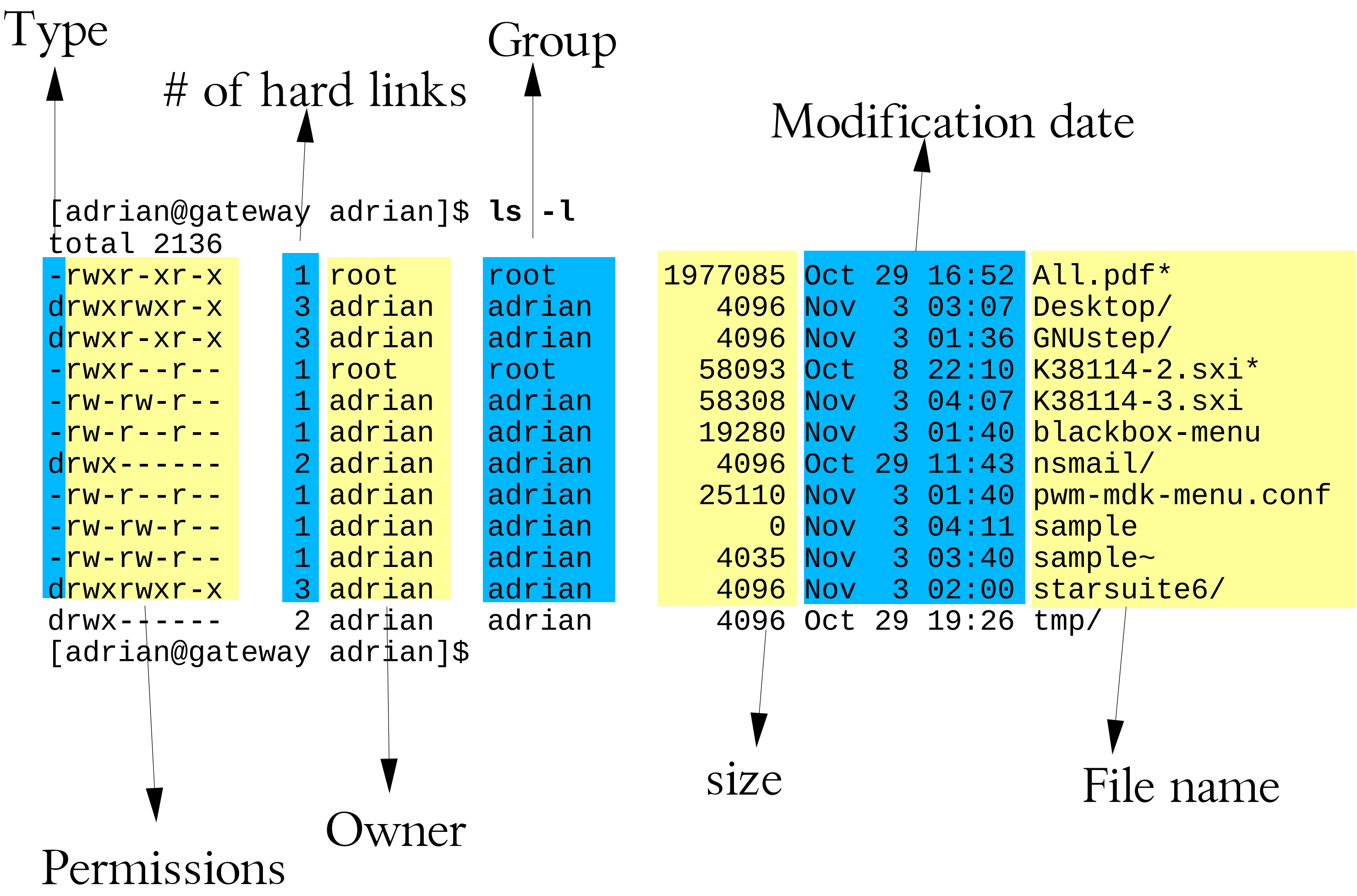
Shell Basics

- Get help:
  - man
  - apropos
  - /usr/share/doc/*
  - Googles

Shell Basics

- Listing directories: ls
  - Long listing: ls -l
  - Include hidden file: ls -a
  - With color: ls --color
  - With mark: ls -F
  - Recurrsive: ls -R

Shell Basics



```
[adrian@gateway adrian]$ ls -l
total 2136
-rwxr-xr-x    1 adrian   root       1977085 Oct 29 16:52 All.pdf*
drwxrwxr-x    3 adrian   adrian        4096 Nov  3 03:07 Desktop/
drwxr-xr-x    3 adrian   adrian        4096 Nov  3 01:36 GNUstep/
-rwxr--r--    1 root     root         58093 Oct  8 22:10 K38114-2.sxi*
-rw-rw-r--    1 adrian   adrian       58308 Nov  3 04:07 K38114-3.sxi
-rw-r--r--    1 adrian   adrian       19280 Nov  3 01:40 blackbox-menu
drwx------    2 adrian   adrian        4096 Oct 29 11:43 nsmail/
-rw-r--r--    1 adrian   adrian       25110 Nov  3 01:40 pwm-mdk-menu.conf
-rw-rw-r--    1 adrian   adrian           0 Nov  3 04:11 sample
-rw-rw-r--    1 adrian   adrian        4035 Nov  3 03:40 sample~
drwxrwxr-x    3 adrian   adrian        4096 Nov  3 02:00 starsuite6/
drwx------    2 adrian   adrian        4096 Oct 29 19:26 tmp/
[adrian@gateway adrian]$
```

File Links

- Hard Links
  - Two symbols pointed to same *content* in FS
  - Not for directories
- Soft Links
  - A symbol pointed to another file
  - Also known as symbolic links
  - It is clear which is the master copy

File Handling

- Viewing content = cat
- Viewing by pages = more / less
- Copy files = cp
- Moving files or rename = mv
- Remove files = rm
- Make directory = mkdir
- Change directory = cd
- Remove directory = rmdir
- Create links = ln

Nomenclature

- Directory separator = /
- Root directory = /
- Local directory = .
- Parent directory = ..
- Home directory = ~
- Escape character = \
- Chars to be escaped = {space,\,/,',",`,*,?,brackets}
- Names are case-sensitive

Streams

- Make output to a file
  - command > file
- Make file as input
  - command < file
- Make command1's output be command2's input
  - command1 | command2
- Append output to file
  - command >> file

Streams

- Make error and output join together
  - command 2>&1
- Here document
  - command << endmark
- Command substitution
  - command `command1`

Filename expansion

- Wildcards: * and ?
- Single character substitution: ls pic-[abcdefg].jpeg
- Single character substitution: ls pic-[a-gA-G].jpeg
- Single character substitution: ls pic-[^h-z].jpeg
- String substitution: ls pic-{mother,father}.jpeg

File modes

- A file can be assigned to have one user and one group of ownership
- Change user owner: chown
- Change group owner: chgrp
- Change permission: chmod

```
[adrian@gateway adrian]$ ls -l
total 2136
-rwxr-xr-x    1 root     root      1977085 Oct 29 16:52 All.pdf*
drwxrwxr-x    3 adrian   adrian       4096 Nov  3 03:07 Desktop/
drwxr-xr-x    3 adrian   adrian       4096 Nov  3 01:36 GNUstep/
-rwxr--r--    1 root     root        58093 Oct  8 22:10 K38114-2.sxi*
-rw-rw-r--    1 adrian   adrian      58308 Nov  3 04:07 K38114-3.sxi
-rw-r--r--    1 adrian   adrian      19280 Nov  3 01:40 blackbox-menu
drwx------    2 adrian   adrian       4096 Oct 29 11:43 nsmail/
-rw-r--r--    1 adrian   adrian      25110 Nov  3 01:40 pwm-mdk-menu.conf
-rw-rw-r--    1 adrian   adrian          0 Nov  3 04:11 sample
-rw-rw-r--    1 adrian   adrian       4035 Nov  3 03:40 sample~
drwxrwxr-x    3 adrian   adrian       4096 Nov  3 02:00 starsuite6/
drwx------    2 adrian   adrian       4096 Oct 29 19:26 tmp/
[adrian@gateway adrian]$
```

File modes

- chown *owner filename*
- chgrp *group filename*
- chmod [augo][+-=][rwxX] *filename*
    - [augo] = {all,user,group,other}
    - [+-=] = {allow,disallow,only}
    - [rwxX] = {read,write,execute,execute}
- chmod *octal_mode filename*
- Change attribute on EXT2: chattr

File modes

- `---` = No access to this file
- `r--` = Read only
- `-w-` = Write only
- `--x` = Execute only
- A directory needs x to cd to
- A directory needs r to ls

Process tracking

- Every running program is a process
- Each process has a process ID, PID
- List process: ps
  - Common usage: ps ax / ps aux / ps
- Continuously list process: top
- Killing process: kill *pid*
- Stronger kill: kill -9 *pid*

Users and Groups

- Add user: useradd
- Remove user: userdel
- Modify user: usermod
- Assign password: passwd
- Add groups: groupadd
- Remove groups: groupdel
- Modify groups: groupmod
- Easier to do: linuxconf

Further study

- Highly recommended:
  Learning the bash Shell 2/e
  Cameron Newham and Bill Rosenblatt
  O'Reilly & Associates

Further Study

- Environment variables
- Echo command
- $[] calculation
- Shell programming
- Streamline editor: sed

  http://pegasus.rutgers.edu/~elflord/unix/sed.html
- Commands:

  sort, grep, head, tail, whoami, pwd, su, tr, cut, uniq, df, du, tar, gzip, compress, bzip2, more, less
- Regular Expressions

Software Management

Software for *nix

- Everything is a file
  - Unlike MS Windows, we have no registry
  - Install/Uninstall = Create/Delete files
- Installation
  - Put files into correct places
  - Execute by calling the name of the executables
- Uninstall
  - Delete corresponding executables
  - Delete correcponding auxillary files
  - Notify other program (sometimes, if needed)

Software Packages

- Source tar ball
  - Archive of source codes
  - Requires compilation
- Binary tar ball
  - Archive of binary program
  - Usually a script is bundled for installation
- Debian Packages
  - dpkg -i *packagefile*
- Red Hat Packages
  - rpm -i *pakcagefile*

Source Tar Ball

- Most UNIX program are written in C/C++
- Install tar ball:

```
# ls
software-1.0.0.tar.gz
# tar zxf software-1.0.0.tar.gz
# ls
software-1.0.0        software-1.0.0.tar.gz
# cd software-1.0.0
# ./configure --prefix=/usr
....
......
# make
....
......
# make install
....
......
#
```

RPM

- The software management system for Red Hat-alike favors
- Widely used
- Dependancy checking
- Software tracking
- Automatic configuration during (un)install is supported

RPM

- Installation
  - rpm -i software-1.0.0-i386.rpm
- Uninstall
  - rpm -e software
- Upgrade
  - rpm -U software-1.0.2-i386.rpm
- Listing
  - rpm -qa
- Package information
  - rpm -qi software
- List files
  - rpm -ql software

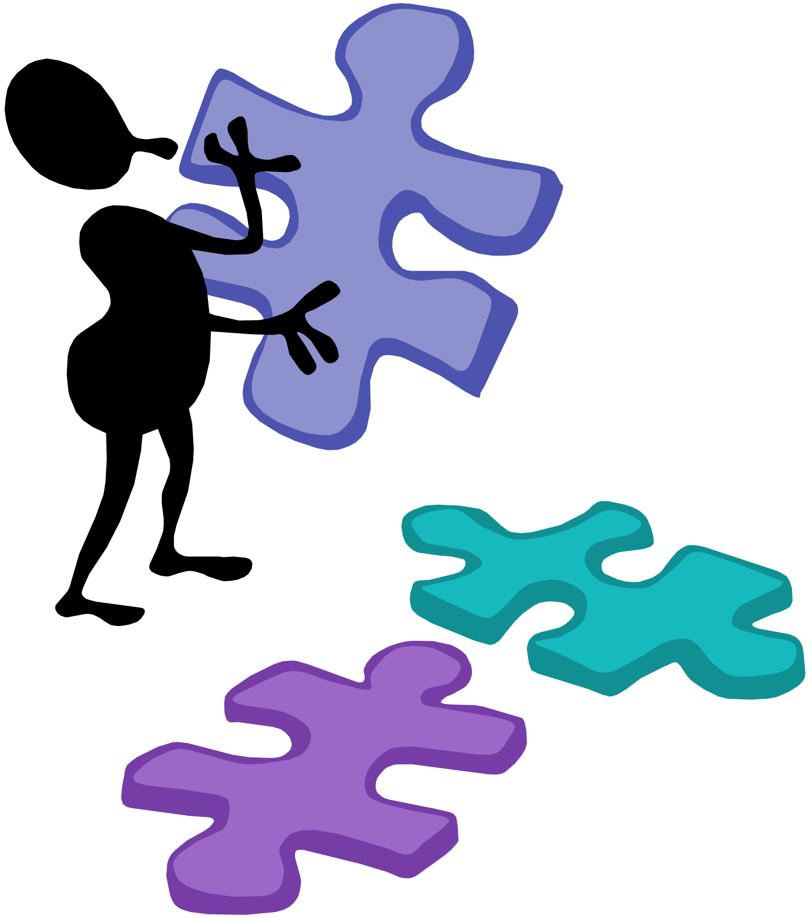
DPKG

- The software management system for Debian-alike favors
- Less-widely used
- Dependancy checking
- Software tracking
- Automatic configuration during (un)install is supported
- Package listing
- Dynamic upgrade
- Internet integration

DPKG

- Installation / Upgrade
  - dpkg -i software-1.0.0.deb
- Remove (Uninstall)
  - dpkg -r software
- Purge
  - dpkg -P softwared
- Listing
  - dpkg -l
- Package information
  - dpkg -p software
- List files
  - dpkg -L software

Kernel Rebuild



Linux Kernel

- Kernel is important, essential, critical
- Develop by Linus Torvalds et al
- Web site at:
  - Main = http://www.kernel.org
  - Crypto = http://www.kerneli.org
- Get it from ftp://ftp.kernel.org

Rebuild Kernel

- We may rebuild kernel because:
  - Upgrade
  - Security fix
  - Modify functions available
  - Add drivers
  - Performance/Stability tuning
  - For fun
  - Other reasons

Rebuild Kernel

- Steps for rebuilding kernel
  - Get a source tar ball from somewhere
  - Extract the tar ball to /usr/src
  - make config / make menuconfig / make xconfig
  - make bzImage / make disk
  - make modules
  - make modules_install
  - make install
  - Re-install boot program (LILO / Grub)
  - Reboot and use the new kernel

Rebuild Kernel

- When make menuconfig, you may see some functions available as linked or available as module
- Monolithic kernel → Linked
- Modules: Load on request → Save memory

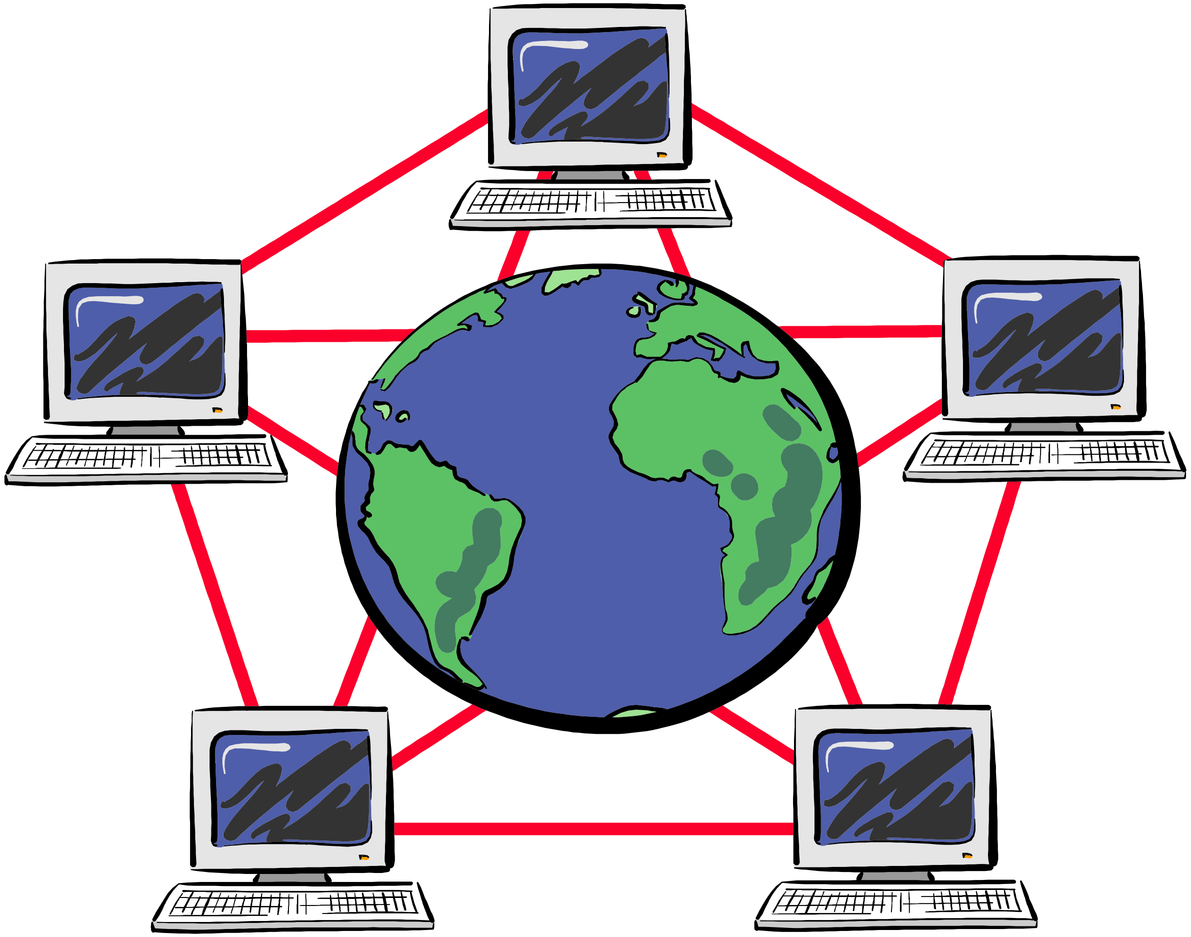
Kernel Modules

- Sometimes, a hardware developer would provide Linux drivers as compiled modules because he do not want to release the source code
- Example: VIA 82C686A Sound Driver

Kernel Modules

- Modules location: /lib/modules/*version*/*
- List modules: lsmod
- Remove modules: rmmod *module_name*
- Load modules: modprobe *module_name*
- Load modules: insmod *module_name*
- Forcefully load modules: insmod -f *module_name*
- Automatically load modules on boot: /etc/modules
- Automatically load modules on request: /etc/modules.conf

Linux Networking



Linux Networking

- Linux is a UNIX favor
- Native networking: TCP/IP
- Inherits many networking capabilities from BSD

Linux Networking

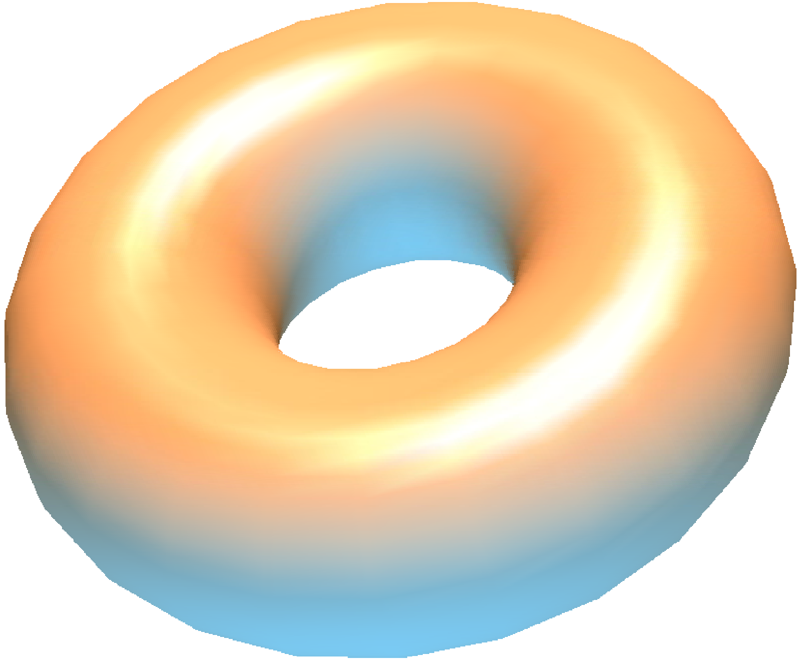
- Network interface configuration
  - RH: /etc/sysconfig/networking
  - Debian: /etc/network/interface
- Device files
  - Ethernet: /dev/eth0, /dev/eth1, ...
  - PPP: /dev/ppp0, /dev/ppp1, ...
  - Tunnels: /dev/tun0, /dev/tun1, ...
- Name Resolution Setting
  - /etc/resolv.conf
  - /etc/hosts

Linux Networking

- Networking commands
  - Config: ifconfig
  - Routing: route
  - Resolution: host / dig / nslookup
  - Ping: ping
  - IP Filtering: iptables / ipchains / ipfwadm
  - States: netstat
  - Download: wget / rsync
  - Browsing: lynx
  - FTP: ftp / ncftp / ...
  - Enable packet forwarding:

    ```
    echo 1 > /proc/sys/net/ipv4/forward
    ```

Basic System Administation



x86 Booting Revisited

- Booting procedure:
  - System loader started
  - Kernel loaded (PID = 0 ?)
  - Initializing essential device drivers (a.k.a. modules)
  - Execute program /sbin/init (PID = 1)
  - init spawns other processes (PID > 1)
    - Follows instructions of /etc/inittab to spawn
    - Modifying /etc/inittab can cause the whole system changed

/etc/inittab

```
# /etc/inittab: init(8) configuration.
id:2:initdefault:               # Default runlevel
si::sysinit:/etc/init.d/rcS     # Run rc script on boot
~~:S:wait:/sbin/sulogin         # What to do in single user mode

# /etc/init.d executes the S and K scripts upon change of runlevel.
l0:0:wait:/etc/init.d/rc 0      # Halt
l1:1:wait:/etc/init.d/rc 1      # single user
l2:2:wait:/etc/init.d/rc 2      # multiuser
l3:3:wait:/etc/init.d/rc 3      # multiuser
l4:4:wait:/etc/init.d/rc 4      # multiuser
l5:5:wait:/etc/init.d/rc 5      # multiuser
l6:6:wait:/etc/init.d/rc 6      # reboot
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin

# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# /sbin/getty invocations for the runlevels.
#  <id>:<runlevels>:<action>:<process>
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

/etc/inittab

- Modifying inittab
  - allows you to change the behavior of system booting
  - you can make a system with no console login
    - unattended server
- Format of inittab
  - Rule of thumb: Read man-pages
  - Every line is:

    code:*runlevel(s)*:*init action*:*command and parameters*
- Reference: Chapter 5 of Running Linux

Runlevels

- Runlevels are defined by /sbin/init
  - Runlevel 1 = Single user mode
  - Runlevel 2,3,4 = CLI multi-user mode
  - Runlevel 5 = GUI multi-user mode
  - Runlevel 6 = Reboot
- /sbin/init calls different set of rc scripts on different runlevels
  - Do different jobs and hence different behaviors on different runlevels

Runlevels

- Change runlevel (root only): init
  - Example: init 5
  - Reboot: init 6
  - Shutdown system: shutdown -h now
    - Do 'init 0' to kill all processes and end-up, then halt the system
- Startup scripts
  - Resides in /etc/rc.d/init.d (RH) or /etc/init.d (Debian)
- rc scripts
  - Resides in /etc/rc.d (RH) or /etc (Debian)
  - Top-level: /etc/rc.d/rc (RH) or /etc/rc (Debian)

Startup Scripts

- Startup scripts
  - Runlevel rc scripts directory: /etc/(rc.d/)rc*N*.d
    - *N* = 0 to 6, correspond to runlevel
  - All files are symlinks to /etc/(rc.d/)init.d/*
  - All files will be executed at that runlevel
  - Filename `Snnxxxx` or `Knnxxxx`
    - `nn` = a number from 00 to 99, marks the sequence
    - `xxxx` = name of the program
    - `K` = killer
    - `S` = Starter

Startup Scripts

- Run all K-script, then all S-script
  - Kill all existing, then
  - Start required programs
- Number indicates the order of execution
  - In ascending order

Virtual Terminals

- After all scripts executed, the system loads VTs
- /etc/inittab contains /sbin/*getty
  - Starts 6 VTs for login, usually
  - Different getty for different behavior
  - Mandrake: mingetty, Debian: getty, Red Hat: agetty
  - XLinux starts a Framebuffer getty for Chinese console on VT #12
  - Switching between VTs: Ctrl+Alt+F*n*
- Sometimes, inittab would load xdm/kdm/gdm for GUI login on runlevel 5

Virtual Terminal

- Kills console login: Delete all getty lines in inittab
  - Unattended server!
- Further detail on /etc/inittab and /sbin/init:
  - Chapter 5 of Running Linux 3/e by Matt Welsh et al

Processes

- Every program are directly or indirectly spawned by /sbin/init
- Every program has a PID > 1
- The information about the program are in /proc/*pid*/*
  - Everything is a file!!
- e.g.: Which command calls this process??
  - cat /proc/*pid*/cmdline
- Process management: kill, killall, ps, top
  - These program just help you to read the data from /proc/*pid*/*

Processes

- ps command
  - ps = List processes running in current login session
  - ps ax = List all processes in the system
  - ps aux = List 'ps ax' with owners' username
- top command
  - Table Of Processes
  - Continuous update

Processes

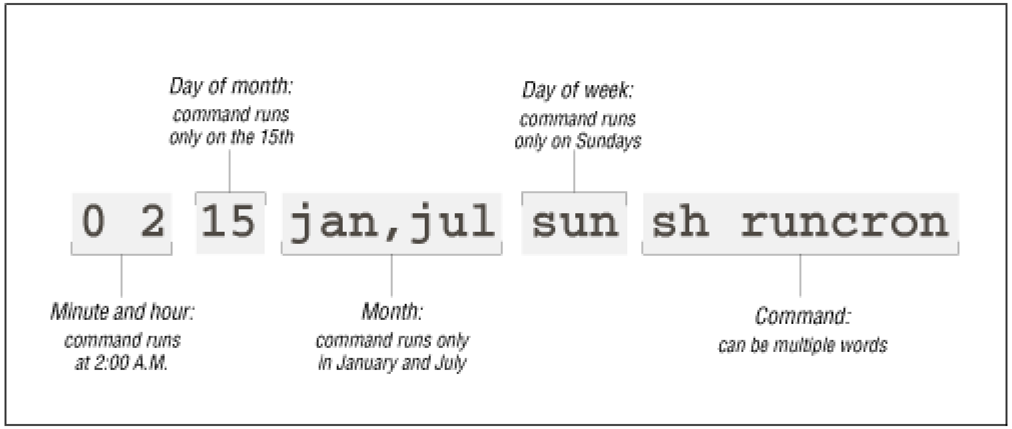
- Kill processes: kill / killall
- Killing mother process may:
  - Kill its child processes
    - Common practice: Kick out a user = Kill its login shell
    - All login consoles are parent of its child processes
  - Make its child process orphan process
    - Those process running in background
    - Those process programmed to run as daemon

/proc/*

- Some system-specific information can be obtained in /proc too
  - PCI bus: /proc/pci
  - IRQ: /proc/interrupt
  - CPU: /proc/cpuinfo
  - I/O port: /proc/ioports
  - Uptime: /proc/uptime
  - CPU loading: /proc/loadavg
  - Memory: /proc/meminfo
- Sometimes we need writing to /proc for changing system behavior (e.g. enable routing)

Automation

- Automation can be done by crond and atd systems
- cron = Process scheduling
  - Regular execution
  - Configuration: /etc/crontab
  - Format: (excerpt from Running Linux 3/e)



```
0 2 15 jan,jul sun sh runcron
```

Day of month: command runs only on the 15th
Day of week: command runs only on Sundays
Minute and hour: command runs at 2:00 A.M.
Month: command runs only in January and July
Command: can be multiple words

Automation

- at job = Delayed execution
  - Preset execution
  - Run once only
  - Need to have atd daemon running
  - Example:

```
# at 16:00
at> slocate -u
at> (Ctrl-D)
job 1 at 2002-09-07 16:00
#
```

Final note...

- O'Reilly has tons of books about UNIX SysAdmin
- Running Linux is a very good introductory reference
- A UNIX System Administrator uses vi, not pico
  - Reference:
    - Learning the vi Editor 6/e (O'Reilly & Associate)
    - Vi Pocket Reference (O'Reilly & Associate)
  - Emacs is an alternative to vi, but it's an all-in-one giant
    - created by the GNU godfather, Richard Stallman
  - Pico is simple but not powerful enough
    - Install through pine

Daemons

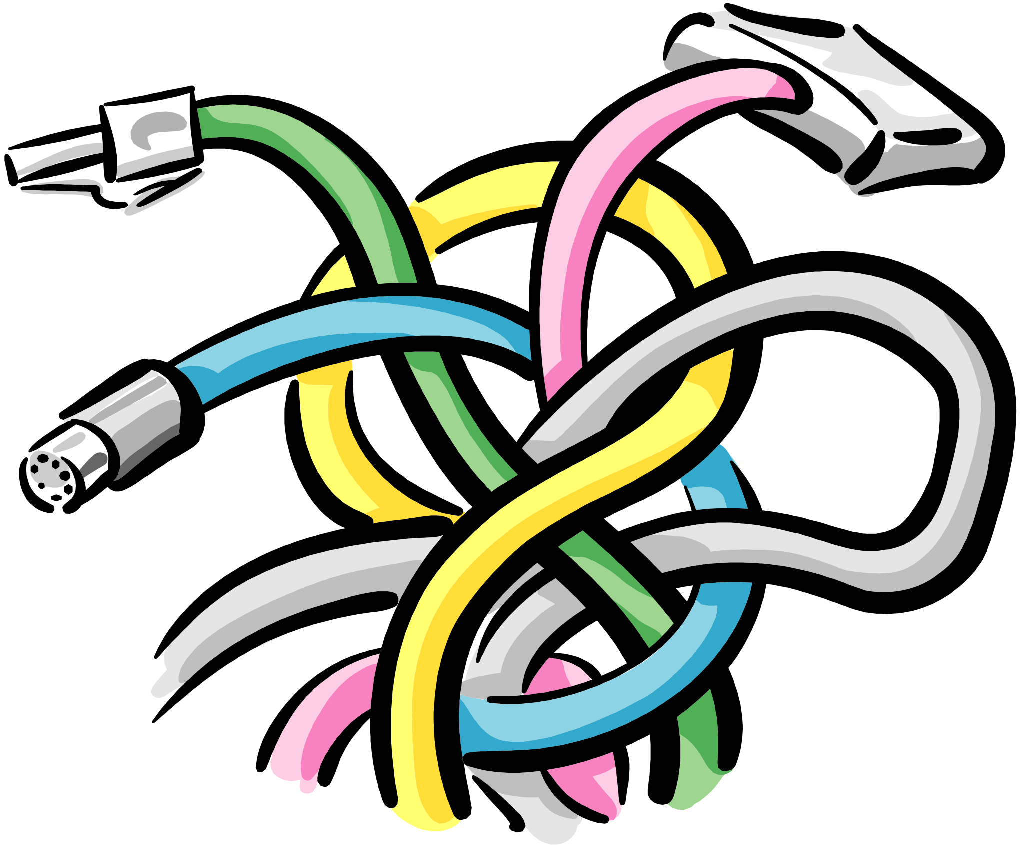
Daemon

- Program?
  - The executable files
- Process?
  - The running program that noticable in ps
- Daemon?
  - A special process that:
    - Generally no parent processes (TTY = “ ? ”)
    - Not disturbing the user, just runs interminably
    - Unless using some method like 'kill' command, it won't stop
    - Mostly listening on some TCP/IP ports (e.g. Apache) or monitoring something (e.g. cron)

Daemon

- Example:
  - Web: /etc/init.d/http
  - FTP: /etc/init.d/proftpd
  - SSH: /etc/init.d/sshd
  - Telnet: /etc/init.d/telnet
  - NFS: /etc/init.d/nfs
  - X Font Server: /etc/init.d/xfs
- Example:
  - cron: /etc/init.d/crond
  - at: /etc/init.d/atd
  - apm: /etc/init.d/apmd

Network Servers



Network Client/Server

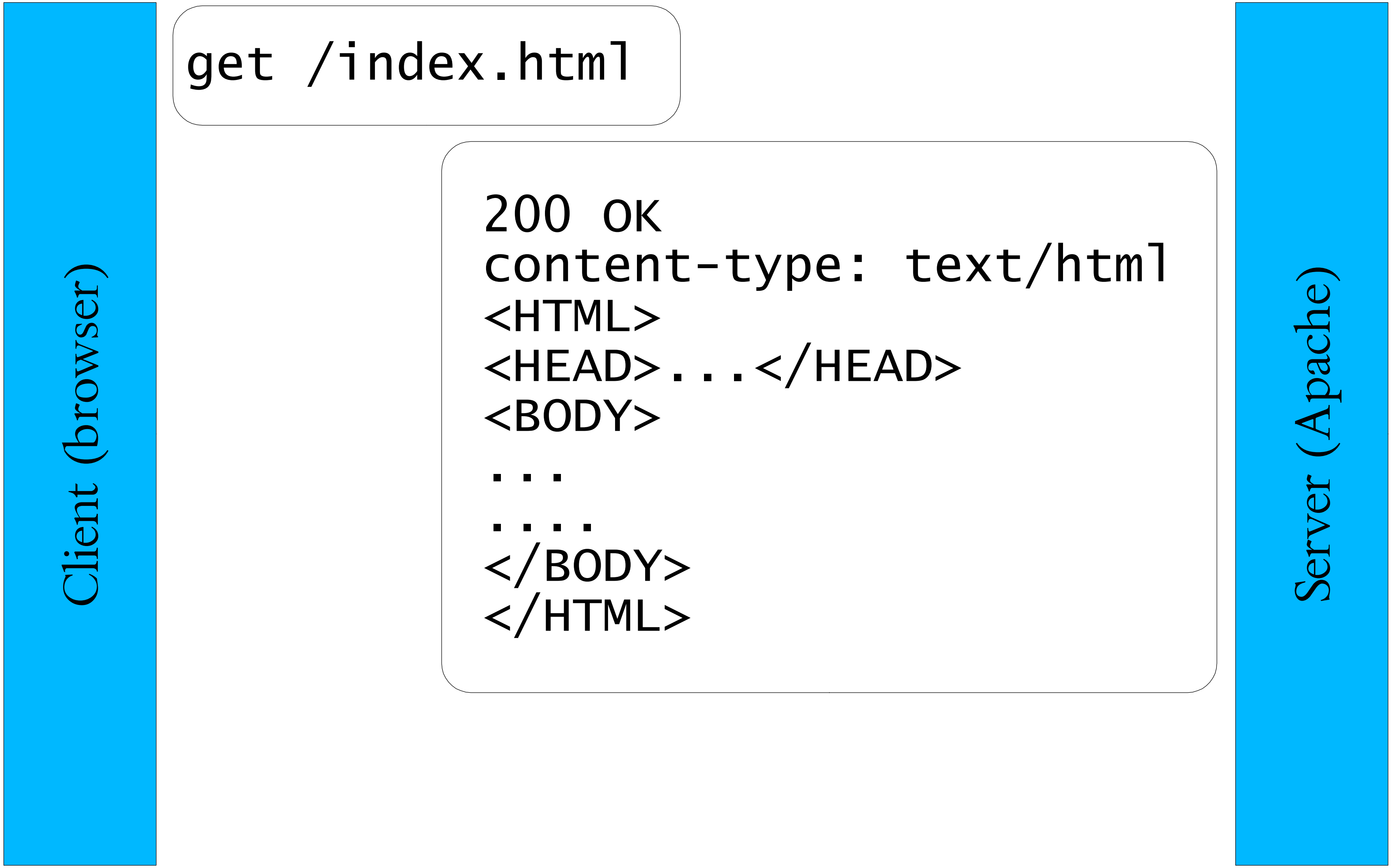
- TCP/IP provides 65536 TCP ports (channel) for communication
- The server takes a port, listen to it
- The client talks to a port, server respond to it
  - Communication!

Network Client/Server

- Example: HTTP
  - Server takes TCP/80 and listen
  - Client sent message “`get /index.html`” to server TCP/80
  - Server response:

```
200 OK
content-type: text/html
<html>
<head>...</head>
<body>............
.........
```

Network Client/Server



Client (browser)

```
get /index.html
```

```
200 OK
content-type: text/html
<HTML>
<HEAD>...</HEAD>
<BODY>
...
....
</BODY>
</HTML>
```

Server (Apache)

Network Client/Server

- Every client-server pair is aimed to communicate between two processes
- They may or may not be in the same host
- Using client-server mechanism for flexibility, expansibility or convention
- Details involved network programming, which is out of our scope here
  - Reference: UNIX Network Programming 2/e Volume 1 by W. Richard Stevens

Common Servers

- Web: Apache (httpd)
- FTP: wu-ftpd or ProFTPd
- Telnet: telnetd
- SSH: OpenSSH
- X-Server: XFree86
- Database: Oracle, MySQL, miniSQL, PostgreSQL
- Mail: Sendmail, postfix, qmail, exim
- DHCP: dhpcd
- News: InterNetNews (innd)
- Web Proxy: Squid
- Routing: Zebra

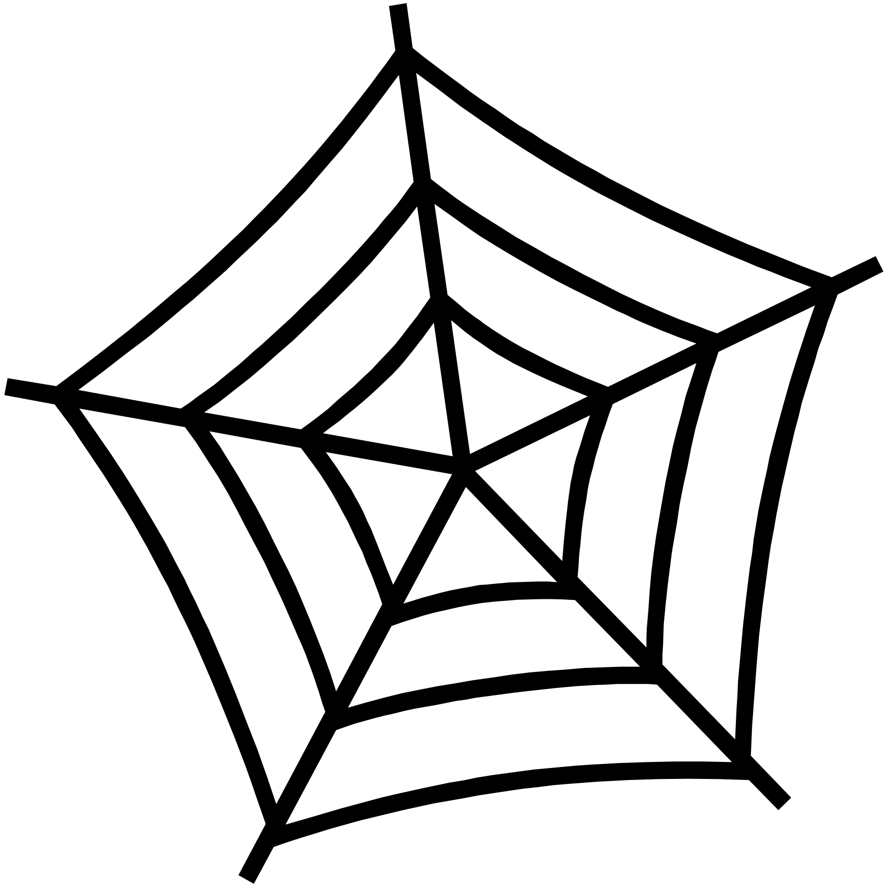
Common Servers

- DNS: BIND
- VPN: PoPToP or FreeS/WAN
- SNMP: UCD-SNMP, mrtg
- File server: Samba, NFS
- Dialup: pppd
- Printing: CUPS, LPRng, LPR
- Firewall: ipfwadm, ipchains, ipfwadm, tcpwrapper
- Groupware: PHPgroupware

Common Servers

- Tons of server softwares available for Linux
- Find what you need through Googles
  - e.g. Find “VPN Linux”

Web Server



Web Server

- Apache
  - Most current version: 2.0
  - 60%+ market share
  - Highly flexible, configurable, robust
- kHTTPd
  - Linux kernel patch
  - Available in all recent kernels
  - Much faster as it is run in kernel mode
  - Plain

Apache Web Server

- After installation,
  - Server program in /usr/sbin
  - Start-up script in /etc(/rc.d)/init.d
  - Configuration file in /etc/apache/httpd.conf
  - Functionality can be extended by using modules
- Configuration: modify httpd.conf

Apache Web Server

- Run it:
  /usr/sbin/apache -d /var/www/data
- Server root: /var/www/data/*
- -d directive: Specify server root
- -f directive: Specify alternative config. file
- Get help:
  - httpd -h
  - http://www.apache.org/

Apache Configration File

```
ServerType standalone
ServerRoot /etc/apache
LockFile /var/lock/apache.lock
PidFile /var/run/apache.pid
ScoreBoardFile /var/run/apache.scoreboard
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150
MaxRequestsPerChild 100

LoadModule config_log_module /usr/lib/apache/1.3/mod_log_config.so
LoadModule mime_magic_module /usr/lib/apache/1.3/mod_mime_magic.so
LoadModule mime_module /usr/lib/apache/1.3/mod_mime.so
LoadModule negotiation_module /usr/lib/apache/1.3/mod_negotiation.so
LoadModule status_module /usr/lib/apache/1.3/mod_status.so
LoadModule autoindex_module /usr/lib/apache/1.3/mod_autoindex.so
LoadModule dir_module /usr/lib/apache/1.3/mod_dir.so
LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so
LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so
LoadModule access_module /usr/lib/apache/1.3/mod_access.so
LoadModule auth_module /usr/lib/apache/1.3/mod_auth.so
LoadModule expires_module /usr/lib/apache/1.3/mod_expires.so
LoadModule unique_id_module /usr/lib/apache/1.3/mod_unique_id.so
LoadModule setenvif_module /usr/lib/apache/1.3/mod_setenvif.so
ExtendedStatus On

Port 80
User www-data
Group www-data
ServerAdmin swtam9@ie.cuhk.edu.hk
DocumentRoot /var/www
```

Apache Configration File

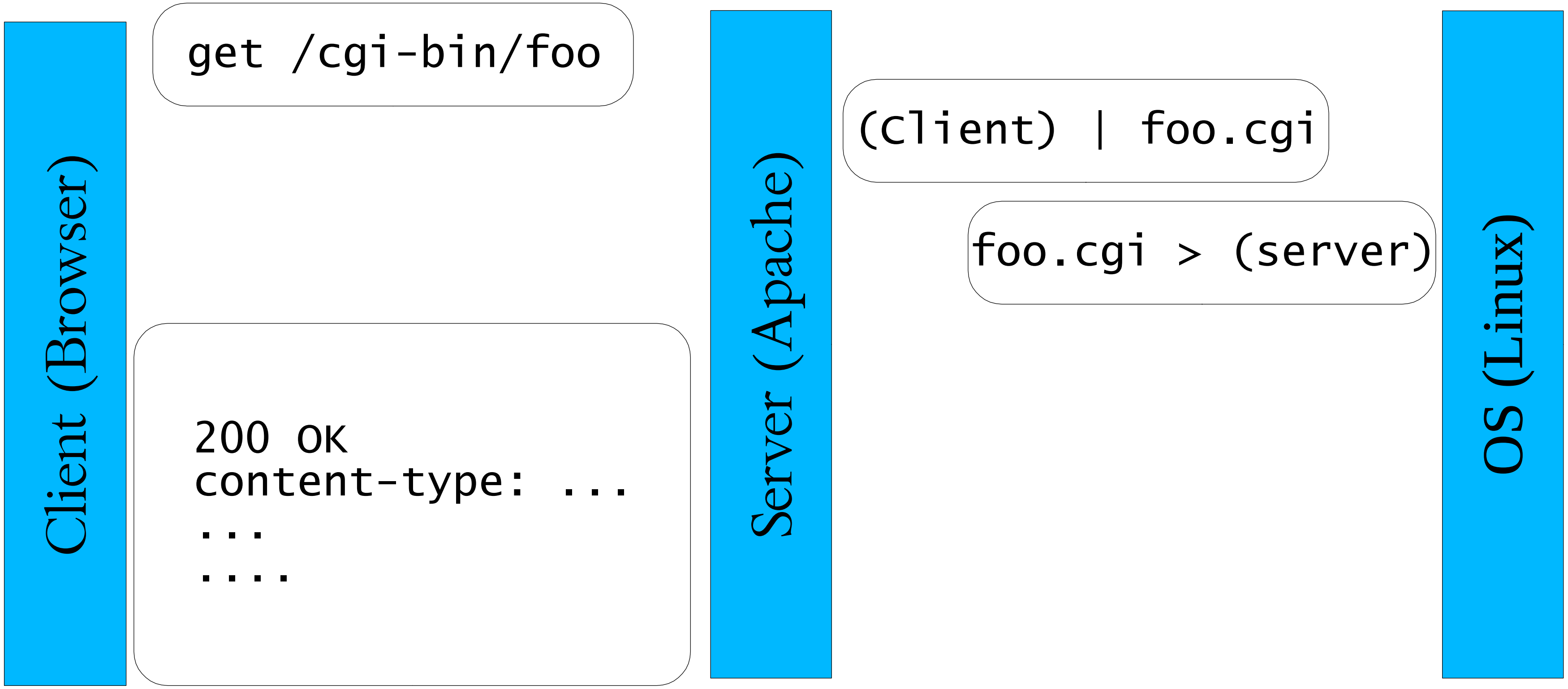
```
<Directory />
    Options SymLinksIfOwnerMatch
    AllowOverride None
</Directory>

<Directory /var/www/>
    Options Indexes Includes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>

<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>

<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml index.cgi
</IfModule>
```

Apache Configuration File

```
AccessFileName .htaccess
UseCanonicalName On
TypesConfig /etc/mime.types
DefaultType text/plain
CustomLog /var/log/apache/access.log combined
ServerSignature On
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

<Directory /usr/lib/cgi-bin/>
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_perl.c>
  Alias /perl/ /var/www/perl/
  <Location /perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
  </Location>
</IfModule>
```

Common Gateway Interface

- CGI = A means to do dynamic content
- Principle:



```
get /cgi-bin/foo
```

```
(client) | foo.cgi
```

```
foo.cgi > (server)
```

```
200 OK
content-type: ...
...
....
```

Common Gateway Interface

- User Input = Environmental variables
- Standard output = Web output

Common Gateway Interface

- How to enable CGI in Apache?
  - Put the scripts in some script directory, e.g. /cgi-bin/*
  - Enable Apache to process CGIs by add directives to the configuration file
    - Pointing out the scripts directory (option ExecCGI)
    - Load the CGI modules (mod_cgi.so)

Web Authentication

- You may want to authenticate a user before he can access your web
- Using the file .htaccess to control the access
  - Filename specified in config file
  - The file contains directives that overrides those in httpd.conf

Web Authentication

- Example .htaccess:

```
AuthType Basic
AuthName "Authorized users only"
AuthUserFile /home/adrian/public_html/passwords
Require valid-user
```

- Create password file

```
# htpasswd -c /home/adrian/public_html/passwords adrian
New password: (password here)
Re-type new password: (password here)
Adding password for user adrian
```

PHP

- A very fast, robust scripting for dynamic content
- Faster and more reliable than CGI
- Low loading
- Integrated into Apache through modules
  - Loads mod_php.so
  - Modifies some directive in httpd.conf for identifying PHP scripts from HTML files

PHP Programming

- Please consult any PHP book (very easy)

Apache + SSL

- SSL = Secure Socket Layer
- An encrypted channel for web content transfer
- You needs the SSL libraries and modules

Apache + SSL

- Configuration:
  - Load SSL module (mod_ssl.so)
  - Configure Apache to tell how, when and where to use SSL

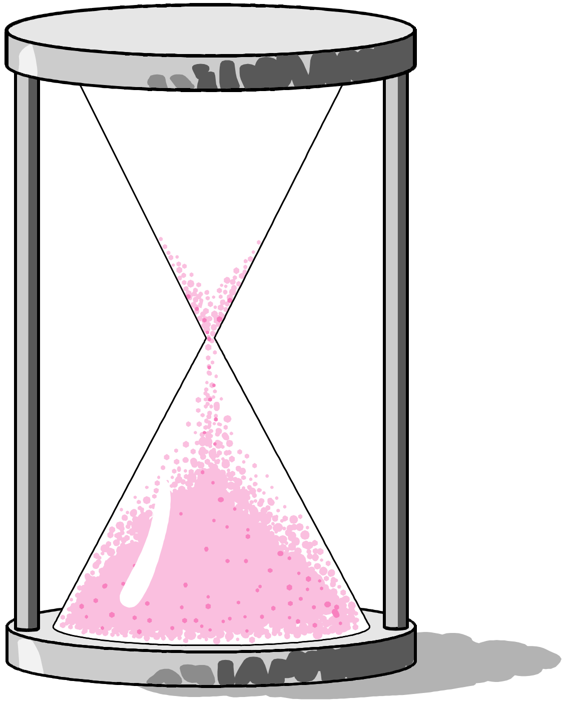
Log files

- Located at /var/log/httpd/*
- Log for:
  - Access
  - Error
  - Secure access
  - Program status
  - etc.

More information

- Main portal of Apache: http://www.apache.org/

Conclusion



Conclusion

- Learning Linux = Learning *nix
- Learning Linux = Read tons of documents
- Learning Linux = Learn to search things on Internet
- Learning Linux = Fun
- Learning Linux = Get addict

Thank you very much