CSI3600 – System Administration

Prof. Fredericks

Management

Outline

- Administration challenges
- Management
  - Workstations
  - Servers
  - Services
- File permissions
- Common Bash commands

Administration Challenges

- What are some challenges to being a system administrator?
  - **You are the reason things don't work, but you are also the solution**
  - Knowledge of software and hardware
  - Be able to communicate effectively
    - Written and verbally
  - Balance requirements
    - Needs of organization
      - Short-term vs. long-term (which fire is more important)
      - User vs. organization
  - **24x7 availability**

What is management?

- Management is…
  - Configuring systems (desktop, servers, etc.)
  - Providing reliability (system uptime)
  - Providing services
    - Email
    - File server

- IT infrastructure
  - Keeps the business running

Workstation Management

- Three key objectives
  1. Provide default and/or necessary applications
     - Initial system state
  2. Manage and deploy software updates
  3. Configure network
     - Access to email / file servers
     - Access to internal network (intranet) or restrict Internet access

Developer Management

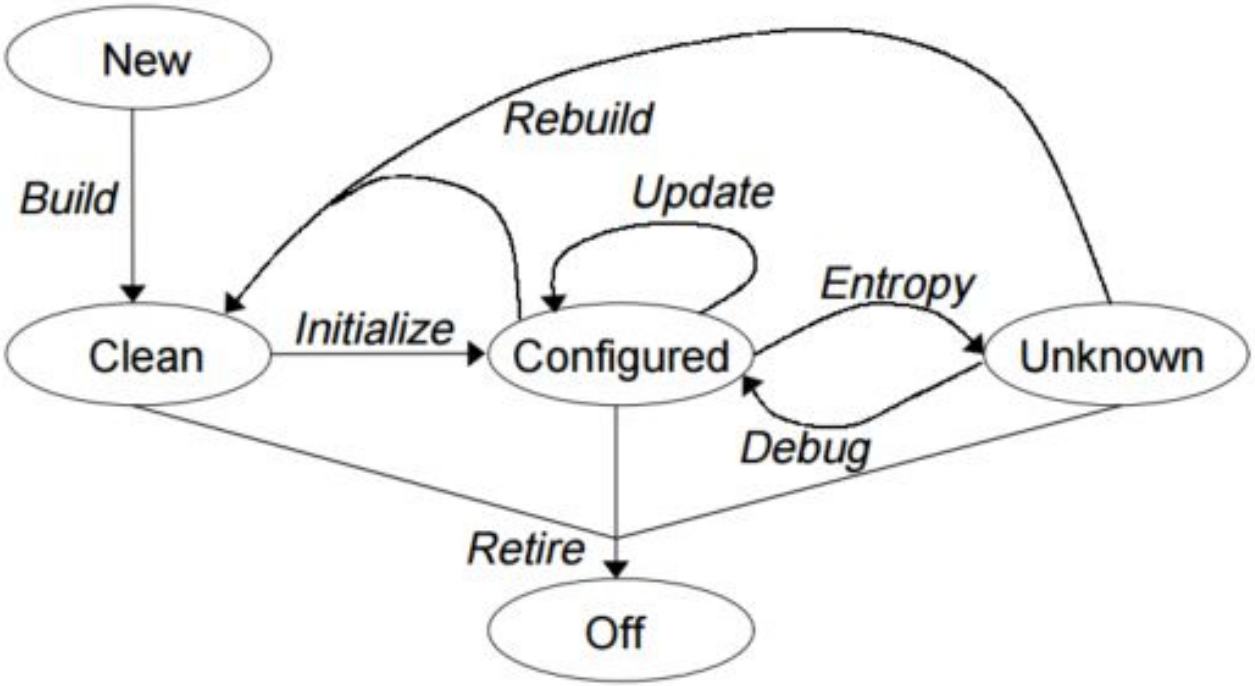
- You may be in charge of administering systems for software developers
- Provide services for:
  - Configuration management
    - Requirements, versioning, etc.
  - Developer tools
    - IDEs, bug trackers, etc.
  - Software releases
  - Quality testing

Handling Multiple Workstations

- Provide initial configuration state (image)
  - Pre-configured default software and applications
- Provide necessary updates to all systems
- Appropriate network configuration

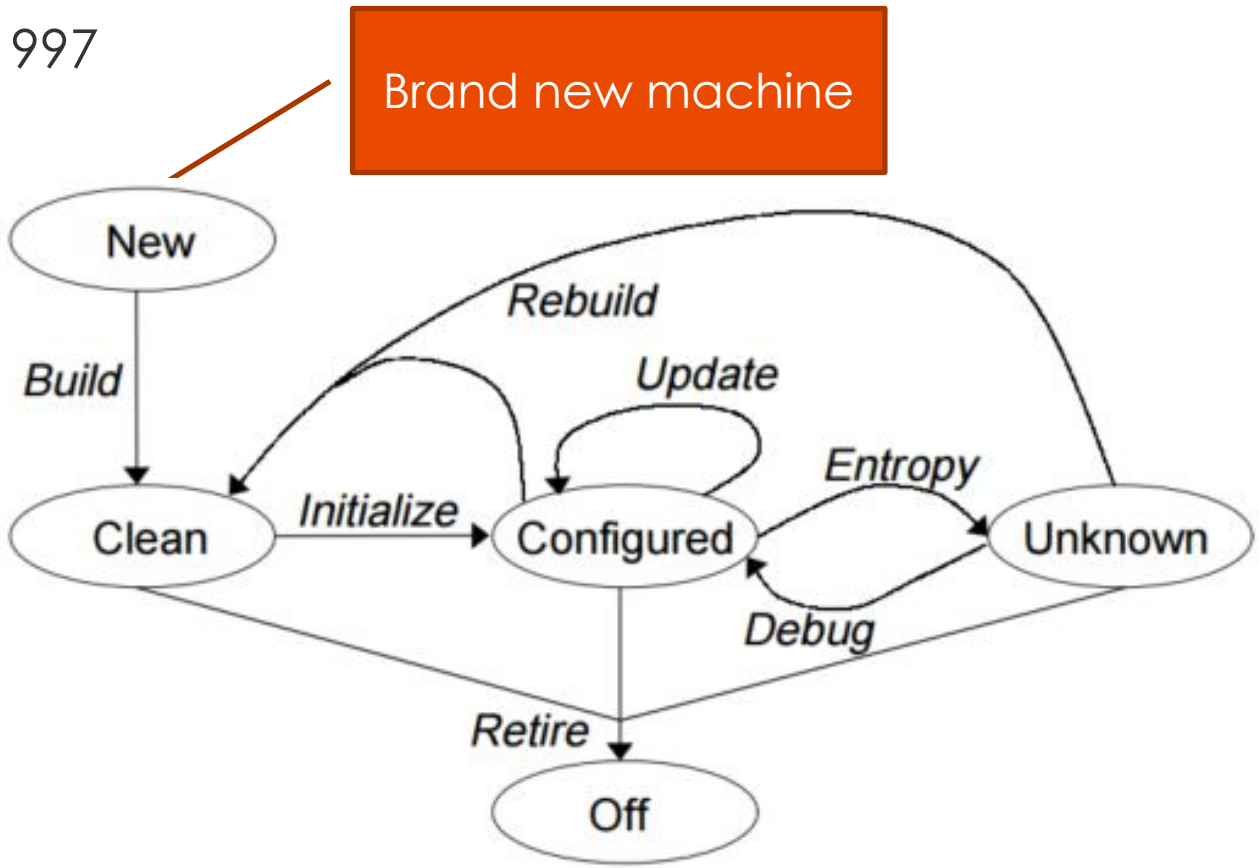
Machine Lifecycle

- Evard 1997



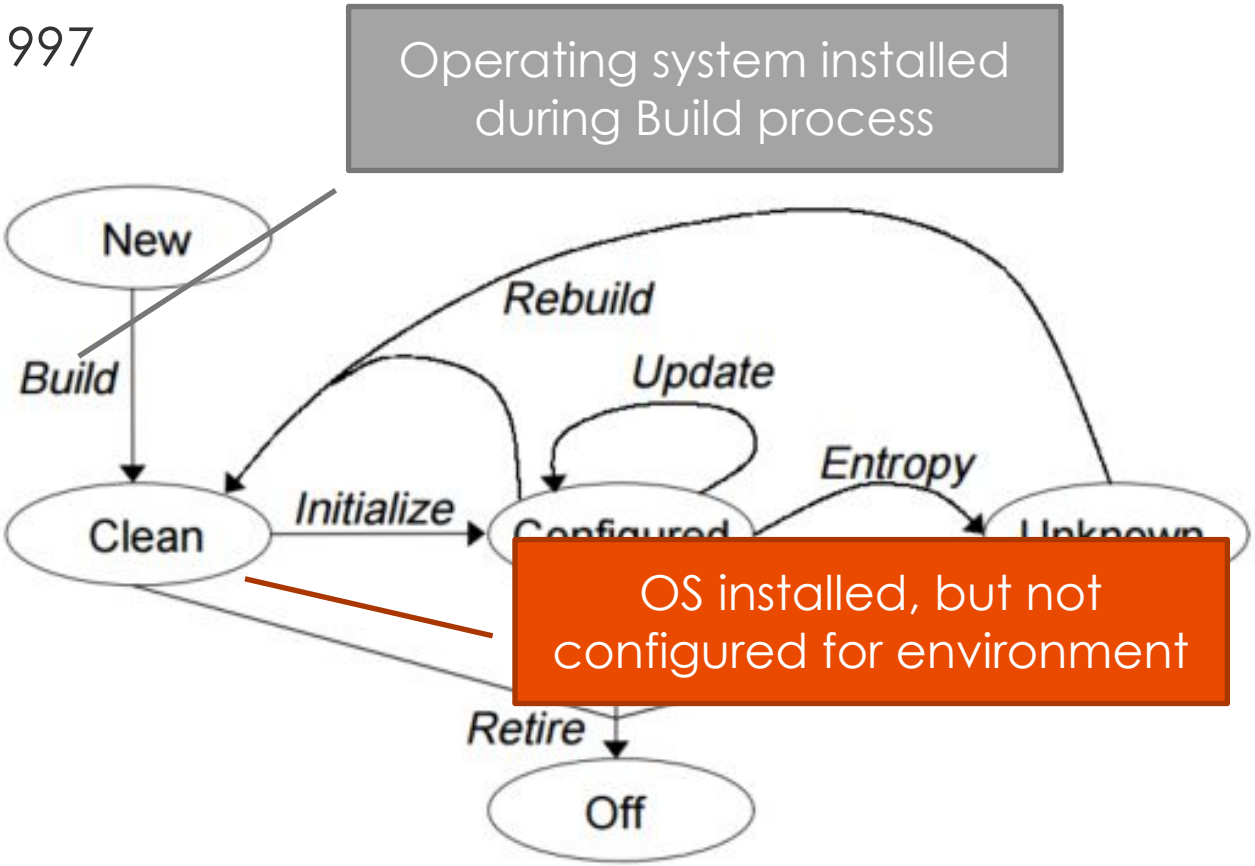
Machine Lifecycle

- Evard 1997



Machine Lifecycle

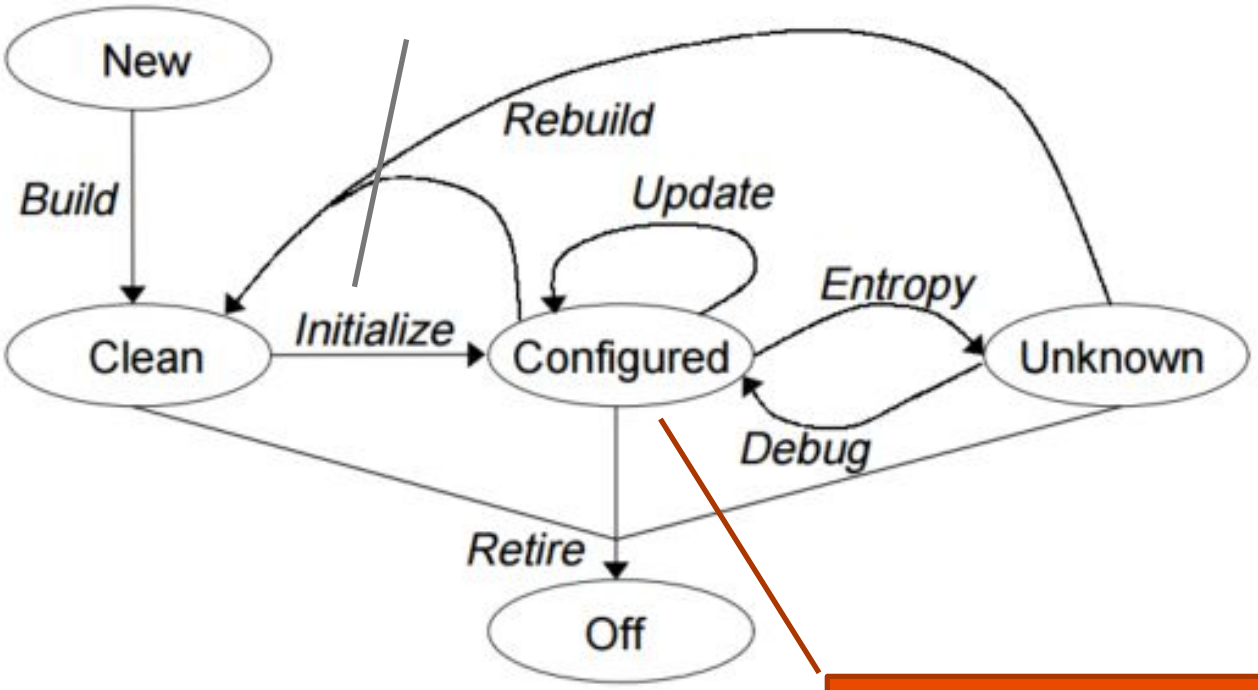
- Evard 1997



Machine Lifecycle

- Evard 1997

Modify OS to support env
[patches, updates, changes]

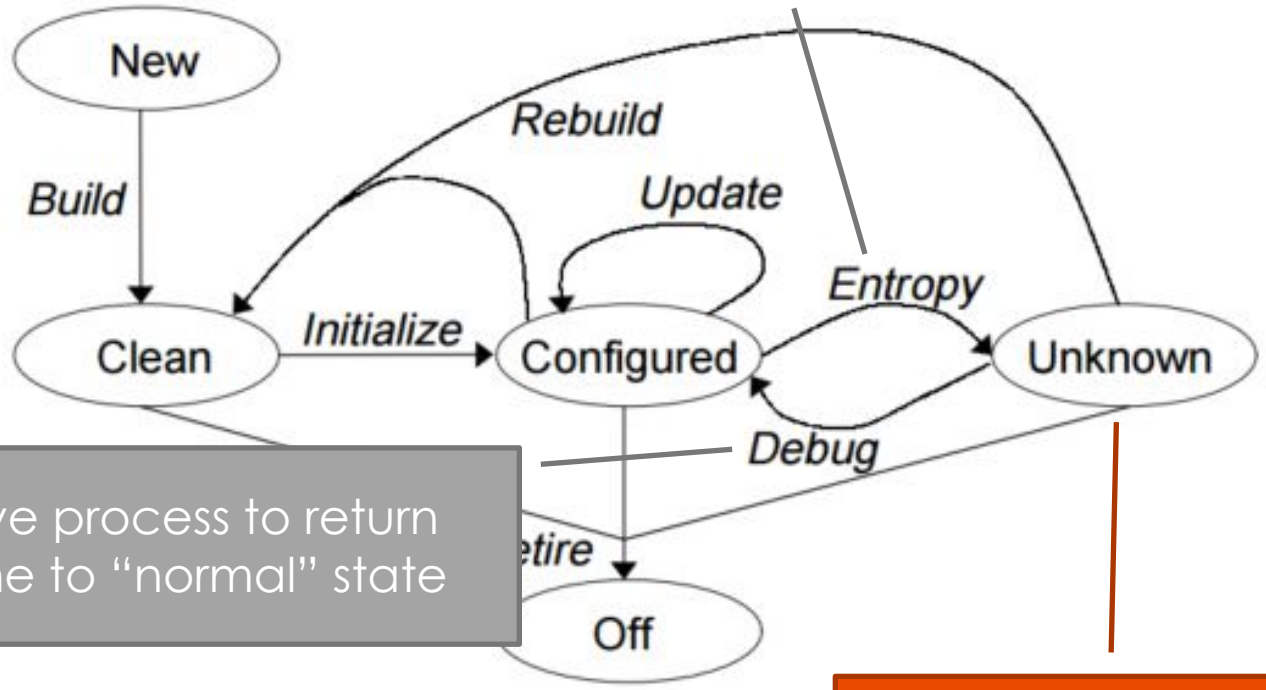


Configured correctly for environment

Machine Lifecycle

- Evard 1997

Gradual change (undisciplined changes, environment changes, etc.)

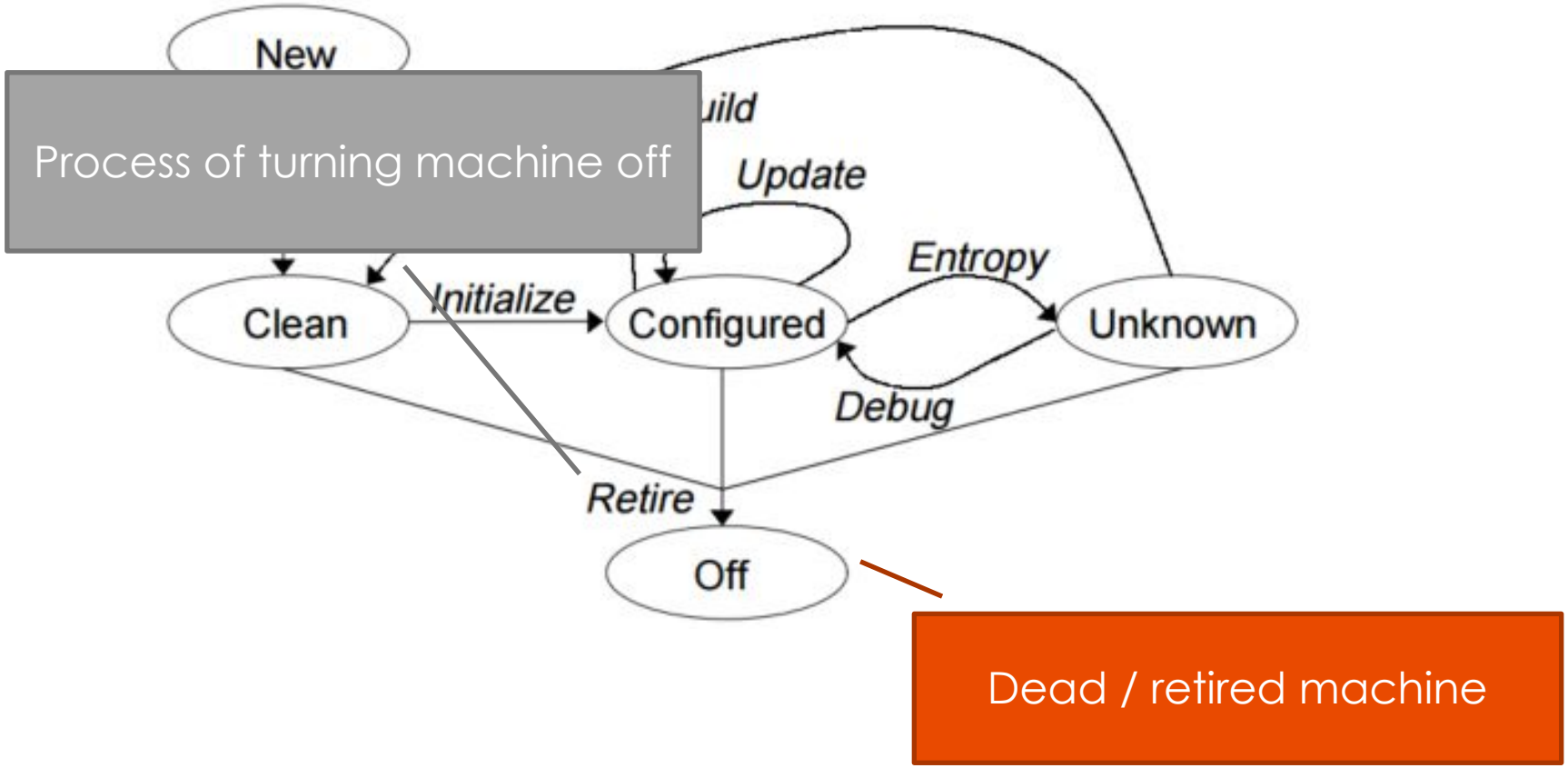


Intensive process to return machine to "normal" state

Misconfigured or out of date

Machine Lifecycle

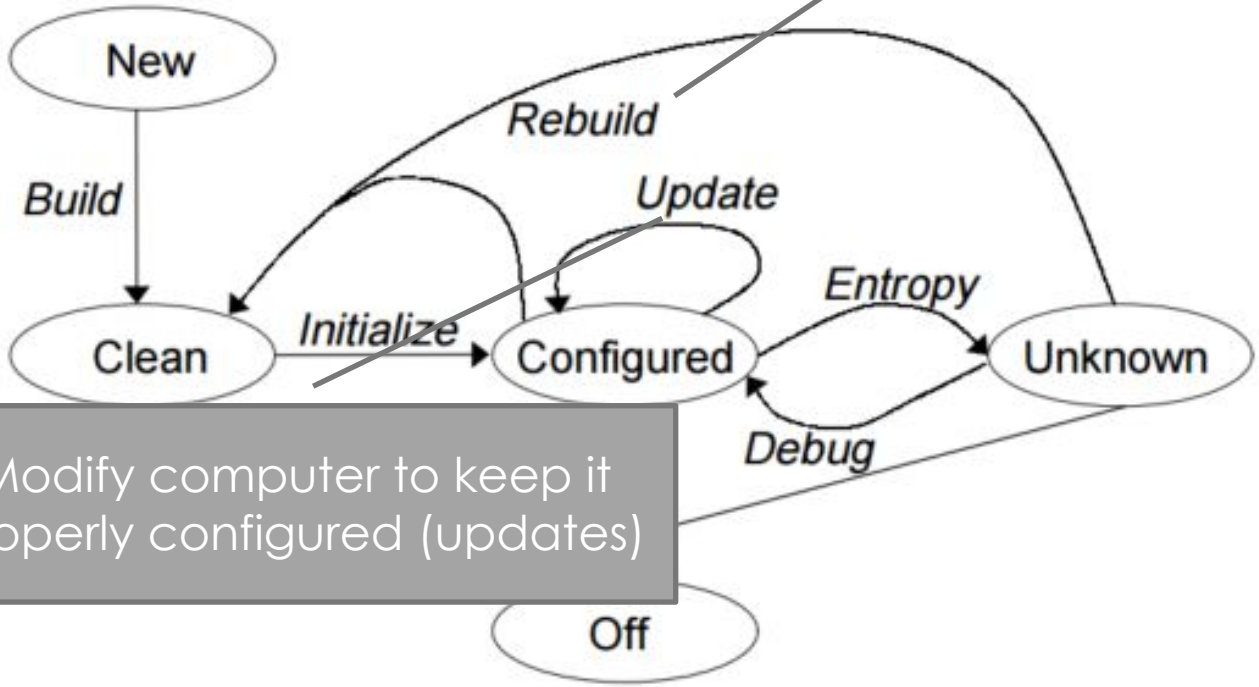
- Evard 1997



Machine Lifecycle

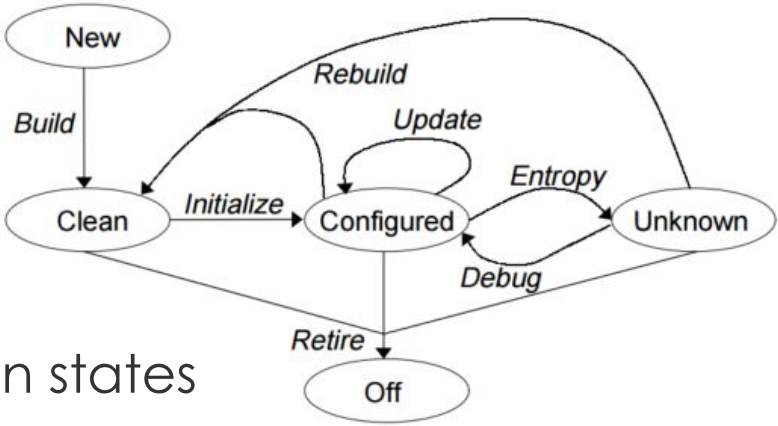
Drastic change that simple update can't handle

- Evard 1997



Modify computer to keep it properly configured (updates)

Machine Lifecycle



- Need to plan for all transitions between states
- Maximize useful time
  - Computer only usable in **Configured** state
- Fast / efficient recovery
  - Automate tasks
    - Manual can be slow and error-prone
  - Minimize entropy
    - Restrict root privileges
    - Control changes made by 3rd party applications
- Rebuild/retire means moving data/applications

Automation

- Automating tasks saves time and can reduce mistakes
  - Full automation: all configuration prompts answered (no user input)
  - Partial automation: requires user input
  - Documentation = consistency
- Examples:
  - Kickstart (Red Hat)
  - JumpStart (Solaris)
- Cloning/ghosting can be used to create a disk image and deploy across a large network
  - CloneZilla : http://clonezilla.org/
  - FOGProject: https://www.fogproject.org/

Vendor

- Save time
- Choose vendor known for **reliability**
  - Some cut corners by using consumer-grade parts rather than server-grade
- Maintenance service / Availability of parts
- Speak with other admins for their recommendations
  - (System Administrator's Guild) www.sage.org and (League of Professional System Administrators) www.lopsa.org
- ***Homogeneous environments*** : can use same vendor
- ***Heterogeneous environments*** : not limited to one vendor

Server Management

- Number of users varies wildly
- Must be:
  - Reliable
  - Always up (uptime)
  - Extremely secure
  - Alive for a long time
- More expensive than workstation, but well worth the cost in the users it serves

Servers

- Differently configured than desktops (OS)
- Typically reside within data center
  - Automated backup
  - Maintenance contracts
  - Environmental controls
  - Remote access
- Higher price point
- Higher quality hardware

Server Hardware

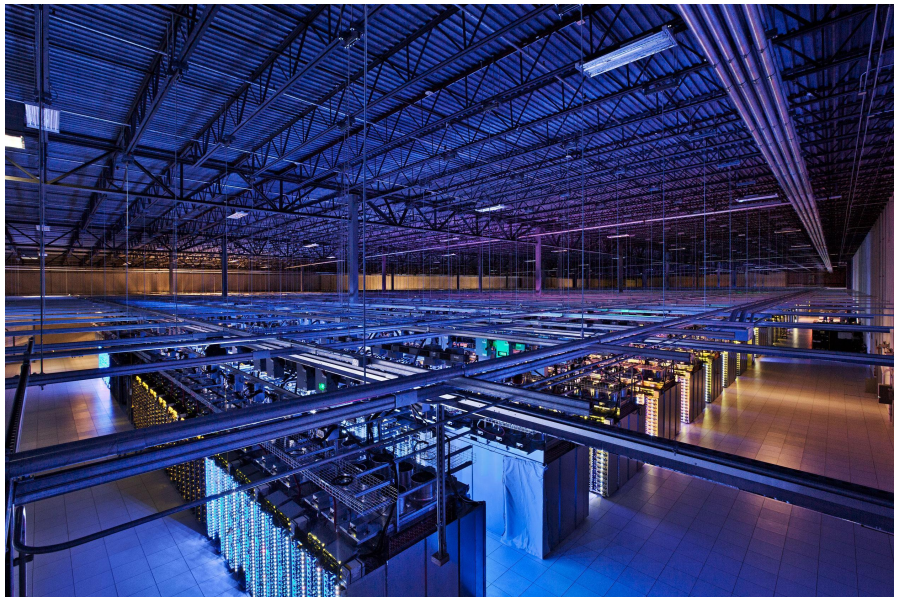
- Use server hardware over conventional hardware
  - Physical space for components
  - CPU performance
  - Disk and network I/O
  - Upgrade options
  - Rack mounts
    - Easily add an additional server blade
- Vendors can save you time!

Data Backups

- Servers often host critical data
  - Backups ensure data is not lost
  - Clients not backed up as often
- Separate server administration network may be worthwhile
  - Offload costly (performance) backup procedures
  - Alternate access if main network is down
  - Requires additional hardware (cables, NICs, etc.)

Data Center

- Why use a data center?
  - Controlled power
  - Controlled temperatures
  - Physical security
  - Fire protection
  - High-performance network capabilities

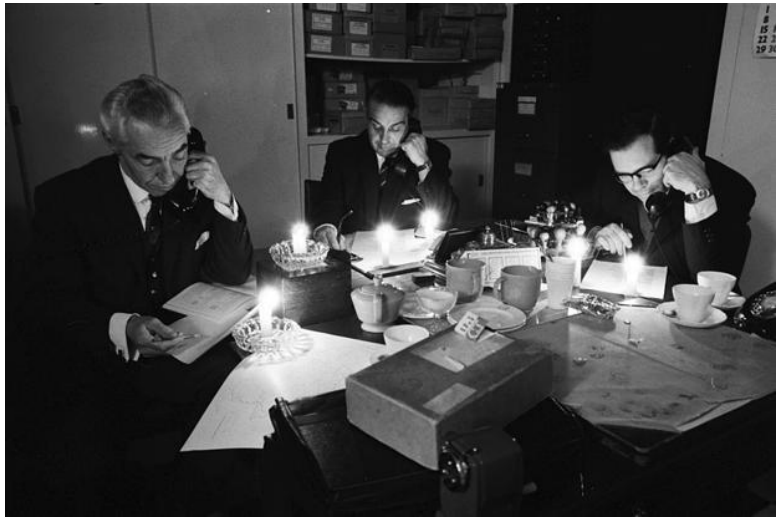


Disk Mirroring

- Disk. drives. fail. often.
- Mirror disks and store offsite
- RAID (redundant array of inexpensive disks / redundant array of independent disks) provides data integrity
  - Virtualization of multiple drives into a single drive with redundancy
  - Numerous levels of RAID
    - ***Striping***: Segment data across different hard drives
    - ***Mirroring***: Replicate data across different hard drives
    - ***Parity***: Data used to check and rebuild damage
  - RAID software widely available
  - RAID still requires backup however

Power Redundancy

- Power. supplies. fail. often.
- Multiple power supplies in case of critical failure
- Separate power cords for each redundant supply
- Separate sources for each power supply (separate UPS)



Hot Swapping

- Replace part while system active
- New components easily added
  - No downtime
- Failed components removed and replaced
  - No outages
- Hot swap increases cost
  - Saves time cost
- Things to check:
  - OS support?
  - Are parts actually hot-swappable?
  - Service interruption?

Server Alternatives

- Server appliances
  - Dedicated to a particular task
  - E.g., file server, router, email server, etc.
- Web services
  - Cloud service (Amazon EC2, Google Cloud)

- Multiple workstations
  - Inexpensive in terms of hardware
  - Expensive in terms of time/maintenance
  - Redundancy to counter reliability problems

Service Management

- Services impose structure on environment
- Provide necessary functionality
- Services include:
  - DNS
  - Email
  - Files
  - Authentication
  - …
- Requires:
  - Reliability
  - Hardware / software
  - Scalability
  - Support / maintenance / monitoring

Designing a Service

- What do you need when you create and deploy a service?
- Customer requirements
  - Reason for service
  - Service-level agreement (SLA)
  - Trial run (demo, limited release, etc)
- Operational requirements
  - Service dependencies
    - Integration?
  - Administration
  - Scalability
  - Impact to network/performance
  - Budget

Service-level Agreements (SLA)

- Contract between service provider and end user
  - Defines service expected from provider

- SLA
  - Description of service
  - Reliability (uptime)
  - Responsiveness (punctuality)
  - Monitoring / reporting (roles, methods, etc.)
  - Consequences for failing to meet SLA
  - Escape clauses / constraints

  - (Can also be used in research!)

https://www.paloaltonetworks.com/resources/learning-center/what-is-a-service-level-agreement-sla.html

Example Of A Service Level Agreement Contract, Continued

SERVICE LEVEL AGREEMENT

Contract Date: Expiration Date:

Agreement Number:

Division:

Location:

Project:

Peak Times:

This document with attachments specifies the agreement between the above named business unit and the Data Processing Center (DPC) for shared computing services. This agreement consists of the following sections:

Section I:	Services To Be Provided
Section II:	Expected Service Requirements
Section III:	Service Assumptions
Section IV:	Costs
Section V:	Contract Maintenance

Service Considerations

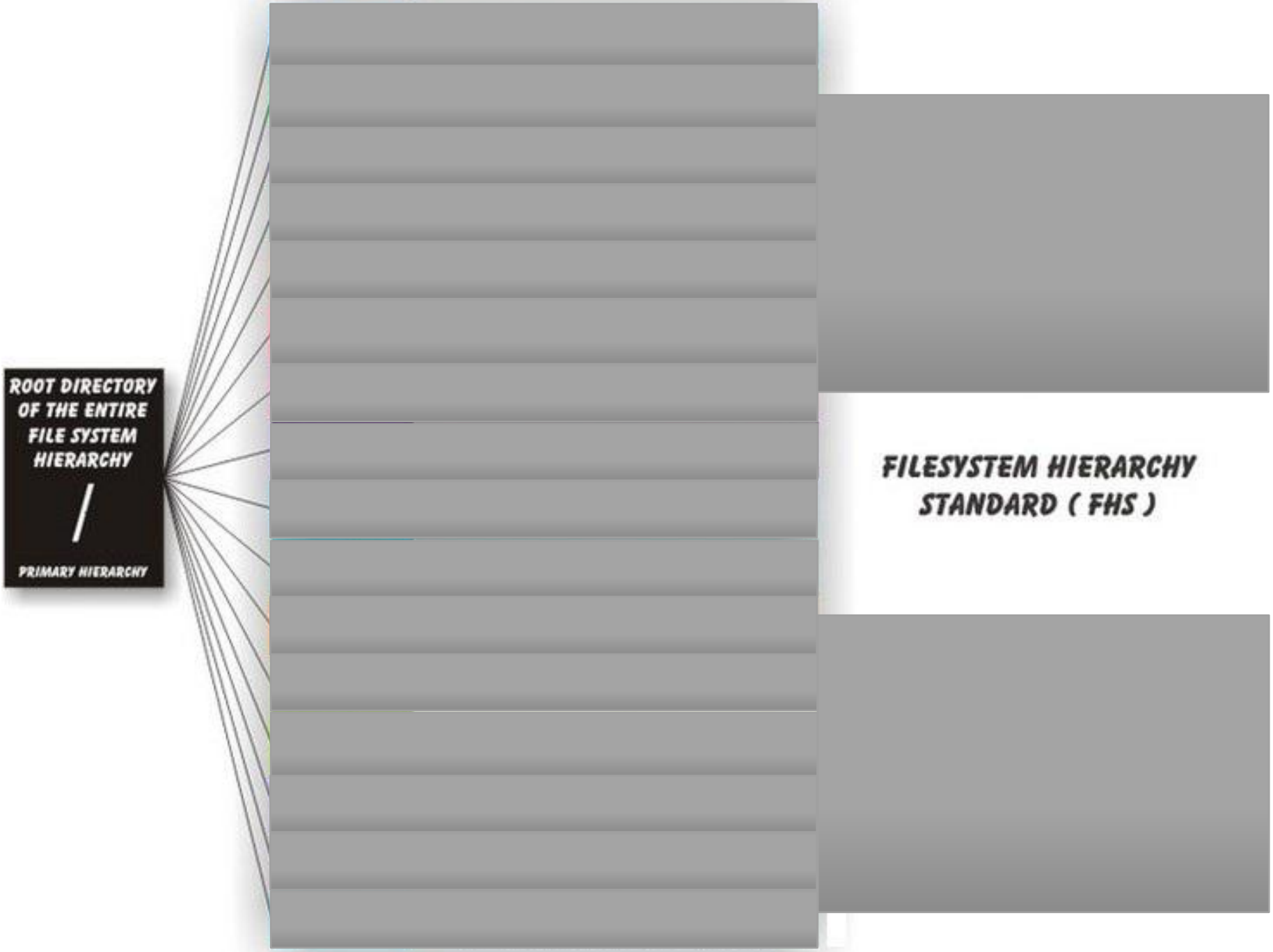
- Reduce single points of failure
  - Common power supplies
  - Network switch
  - Other service dependencies
    - Authentication, DNS, etc.
- Reliability
  - Use reliable hardware
  - Exploit redundancy
  - KISS principle
- Restrict access
  - Customers don't need physical access

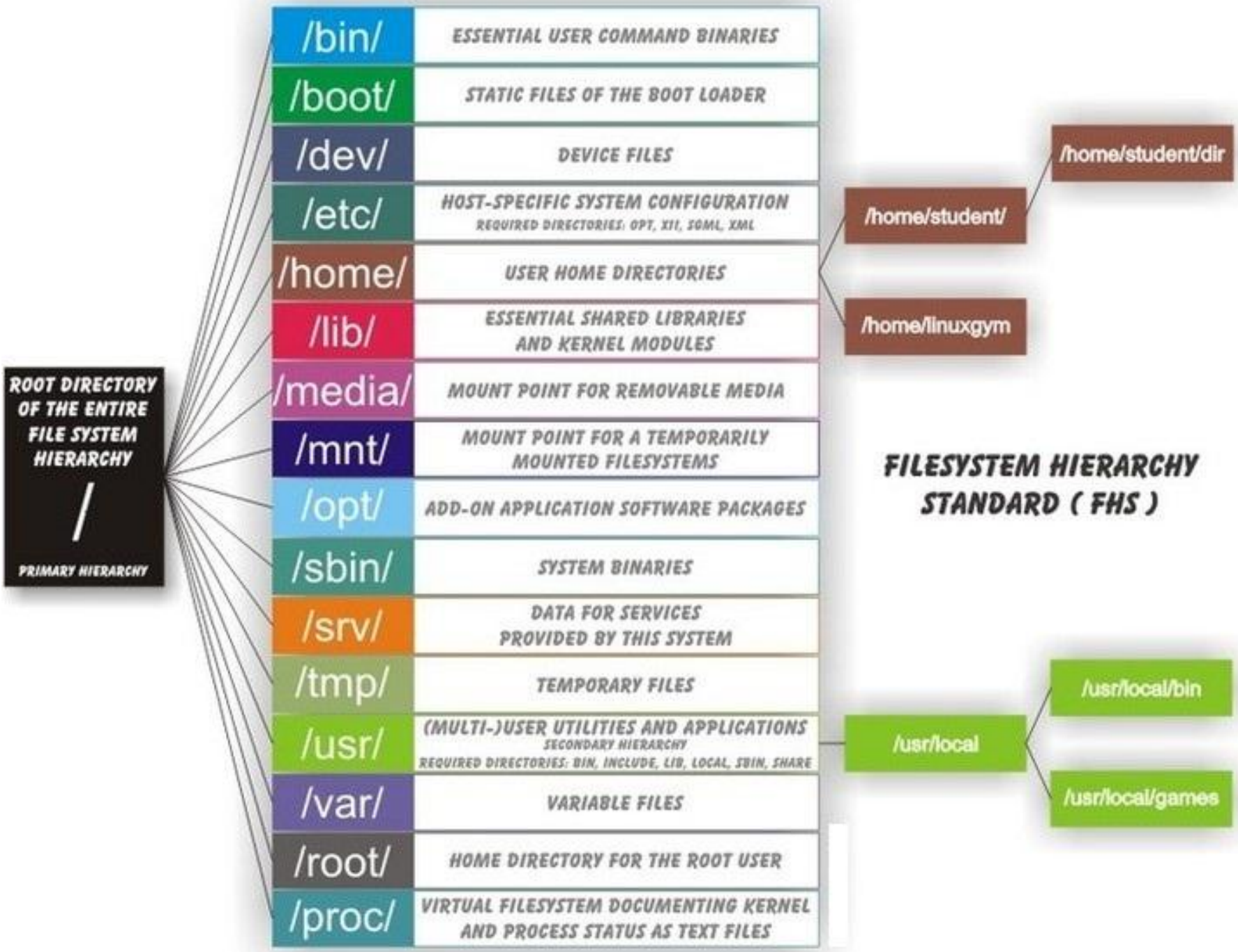
Service Considerations

- Performance
  - Consider scalability problem
    - Proper estimation of capacity over time
    - Split across multiple machines if necessary
- Monitoring
  - Ensure helpdesk or IT staff is automatically alerted in case of failure
  - Keep an eye on performance and capacity
  - Ensure customers don't see the problems before your staff does!
- Ensure documentation is available!
  - Staff fully trained

Filesystem Hierarchy Standard (FHS)

- Linux directory structure typically based on FHS
  - Defines standard directory structure and contents for Unix/Unix-like systems
  - Official site: http://www.pathname.com/fhs


ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY
/
PRIMARY HIERARCHY
FILESYSTEM HIERARCHY STANDARD (FHS)


ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY
/
PRIMARY HIERARCHY
/bin/
ESSENTIAL USER COMMAND BINARIES
/boot/
STATIC FILES OF THE BOOT LOADER
/dev/
DEVICE FILES
/etc/
HOST-SPECIFIC SYSTEM CONFIGURATION
REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/
USER HOME DIRECTORIES
/lib/
ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/
MOUNT POINT FOR REMOVABLE MEDIA
/mnt/
MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/
ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/
SYSTEM BINARIES
/srv/
DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/
TEMPORARY FILES
/usr/
(MULTI-)USER UTILITIES AND APPLICATIONS
SECONDARY HIERARCHY
REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/
VARIABLE FILES
/root/
HOME DIRECTORY FOR THE ROOT USER
/proc/
VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES
/home/student/
/home/student/dir
/home/linuxgym
/usr/local
/usr/local/bin
/usr/local/games
FILESYSTEM HIERARCHY STANDARD (FHS)

Directory	Description
/bin	Contains binary commands for use by all users
/boot	Contains the Linux kernel and files used by the boot loader
/dev	Contains device files
/etc	Contains system-specific configuration files
/home	Is the default location for user home directories
/lib	Contains shared program libraries (used by the commands in /bin and /sbin) as well as kernel modules
/media	A directory that contains subdirectories used for accessing (mounting) filesystems on removable media devices such as floppy disks, DVDs, and USB flash drives
/mnt	An empty directory used for temporarily accessing filesystems on removable media devices
/opt	Stores additional software programs
/proc	Contains process and kernel information
/root	Is the root user's home directory
/sbin	Contains system binary commands (used for administration)
/tmp	Holds temporary files created by programs
/usr	Contains most system commands and utilities—contains the following directories: /usr/bin—User binary commands /usr/games—Educational programs and games /usr/include—C program header files /usr/lib—Libraries/usr/local—Local programs /usr/sbin—System binary commands /usr/share—Files that are architecture independent /usr/src—Source code /usr/X11R6—The X Window system (sometimes replaced by /etc/X11)
/usr/local	Is the location for most additional programs
/var	Contains log files and spools

In-Class Assignment / Break (10-15)

- I've mentioned that we'll have a course project
  - Again, running a dedicated server for a particular **service**
- **Jot down 2-3 possible topics for your project that you would like to do**
  - Again, this should be something interesting to you!
  - If you are thinking of working on a team, this is a good opportunity to form that team!
    - (I am also OK if you go to 3 people, but it must merit 3 people)
  - **Mention what the service is and what it is intended to do**

Bash

- Common uses of CLI
  - Managing files / directories
  - Search
  - File linking
  - Managing permissions

Managing Files and Directories

- **mkdir** command: creates new directories
  - Arguments specify directory's absolute or relative pathname
- **mv** command: moves files
  - Minimum of two arguments:
    - Source file/directory (may specify multiple sources)
    - Target file/directory
  - Pathnames can be absolute or relative
    - For multiple files, can use wildcards in pathname
  - Also used to rename files or directories
  - E.g., **mv** *file1 file2*

Managing Files and Directories

- **cp** command: copies files
  - Same arguments as the **mv** command
  - Also used to make copies of files
- To copy a directory full of files, you must tell the **cp** command that the copy will be recursive
  - Recursive copy command copies the directory and all subdirectories and contents
  - Recursive search includes all subdirectories in a directory and their contents
  - Use **–r** option
  - E.g., **cp –r** *original_dir copied_dir*

Managing Files and Directories

- If the target is a file/directory that exists
  - Both the **mv** and **cp** commands **DO NOT** warn the user that the target file will be overwritten and will ask whether to continue (in Scientific)
    - (Fedora Linux does warn the user)

Search

- Different commands for searching
  - **locate**
  - Receives full or partial filename as argument
  - Looks in a premade indexed database of all files on system
    - To update the database use the **updatedb** command
      - Usually updated once a day with cron
    - Faster than **find**
  - Lots of information returned

Search

- **find**
  - Recursively search for files starting from a specified directory
  - Slower than locate command, but more versatile
  - Format: **find <start directory> -criteria <what to find>**
    - e.g., **find / -name project**
  - If using wildcard metacharacters, ensure that they are interpreted by the find command
    - Place wildcards in quotation marks
  - To reduce search time, specify subdirectory to be searched

Search - Find

Criteria	Description
`-amin -x`	Searches for files that were accessed less than *x* minutes ago
`-amin +x`	Searches for files that were accessed more than *x* minutes ago
`-atime -x`	Searches for files that were accessed less than *x* days ago
`-atime +x`	Searches for files that were accessed more than *x* days ago
`-empty`	Searches for empty files or directories
`-fstype x`	Searches for files if they are on a certain filesystem *x* (where n could be ext2, ext3, and so on)
`-group x`	Searches for files that are owned by a certain group or GID (*x*)
`-inum x`	Searches for files that have an inode number of *x*
`-mmin -x`	Searches for files that were modified less than *x* minutes ago
`-mmin +x`	Searches for files that were modified more than x minutes ago
`-mtime -x`	Searches for files that were modified less than *x* days ago
`-mtime +x`	Searches for files that were modified more than *x* days ago
`-name x`	Searches for a certain filename *x* (*x* can contain wildcards)
`-regex x`	Searches for certain filenames using regular expressions instead of wildcard metacharacters
`-size -x`	Searches for files with a size less than *x*
`-size x`	Searches for files with a size of *x*
`-size +x`	Searches for files with a size greater than *x*
`-type x`	Searches for files of type *x* where *x* is: • b for block files • c for character files • d for directory files • p for named pipes • f for regular files • l for symbolic links (shortcuts) • s for sockets
`-user x`	Searches for files owned by a certain user or UID (*x*)

Search

- PATH environment variable
  - Lists directories where executable files are
  - Run an executable without specifying full path
    - e.g., **find** vs. **/bin/find**
  - **echo $PATH** → prints contents of PATH variable

```
[fredericks@SciLinux6 ~]$ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbi
n:/sbin:/home/fredericks/bin
```

- **which**
  - Search for executable along PATH

```
[fredericks@SciLinux6 ~]$ which find
/bin/find
```

Search (Inside Files)

- **grep**
  - **grep** (global regular expression print) command: displays lines in a text file that match common regular expressions
  - E.g., **grep <pattern> <file>**
  - **egrep** command: displays lines in a text file that match extended regular expressions
    - Can be written as **grep** –E
  - **fgrep** command: does not interpret any regular expressions
    - Returns results much faster than **egrep**
    - Can be written as **grep** -F

grep

- Requires two arguments
  1. Text to search for (can use regular expressions)
  2. Files in which to search
- **grep** is case sensitive
  - For case-insensitive search, use –i option
- **grep** matches patterns of text, ignoring division into words
  - To search only for occurrences of a word, surround it by space characters

Linking

- Two types of links
  - Symbolic link (soft link / symlink)
    - Pointer to file
    - Shortcut
    - **ln –s** *file_1 symbolic_link_file*
  - Hard link
    - Share data
    - **ln** *file_1 copied_file*

Linking Example

- **ls** *–la*

```
[fredericks@SciLinux6 CIT348_Examples]$ ls -la
total 12
drwxr-xr-x. 2 fredericks wheel      4096 Sep  3 12:14 .
drwxr-xr-x. 3 fredericks fredericks 4096 Sep  3 12:14 ..
-rw-r--r--. 1 fredericks wheel        24 Sep  3 12:14 source_file.txt
```

- **ln** *–s source_file.txt symlink_file*

```
[fredericks@SciLinux6 CIT348_Examples]$ ln -s source_file.txt symlink_file
[fredericks@SciLinux6 CIT348_Examples]$ ls -la
total 12
drwxr-xr-x. 2 fredericks wheel      4096 Sep  3 12:17 .
drwxr-xr-x. 3 fredericks fredericks 4096 Sep  3 12:14 ..
-rw-r--r--. 1 fredericks wheel        24 Sep  3 12:14 source_file.txt
lrwxrwxrwx. 1 fredericks wheel        15 Sep  3 12:17 symlink_file -> source_file.txt
```

Linking Example

- **ln** *source_file.txt copied_file.txt*

```
[fredericks@SciLinux6 CIT348_Examples]$ ln source_file.txt copied_file.txt
[fredericks@SciLinux6 CIT348_Examples]$ ls -la
total 16
drwxr-xr-x. 2 fredericks wheel      4096 Sep  3 12:17 .
drwxr-xr-x. 3 fredericks fredericks 4096 Sep  3 12:14 ..
-rw-r--r--. 2 fredericks wheel        24 Sep  3 12:14 copied_file.txt
-rw-r--r--. 2 fredericks wheel        24 Sep  3 12:14 source_file.txt
lrwxrwxrwx. 1 fredericks wheel        15 Sep  3 12:17 symlink_file -> source_file.txt
```

File Permissions

- Permissions are tracked based on user ID (UID)
  - Numerical value assigned to account
  - Permissions can be assigned to users or groups

- Users revisited
  - Users stored in /etc/passwd file
    - Passwords stored in /etc/shadow
    - Services have their own user accounts

File / Directory Ownership

- Primary group : user's default group
- When you create a file or directory, user and default group become owners
  - Anybody in user's group will have access to that file

- **whoami** – current user name
- **groups** – list primary group and group memberships
- **touch** – create an empty file

Ownership

- **chown** – change ownership of directory/file
  - Arguments: owner and which file/directory to change
  - -R option makes changes recursive

- **chgrp** – change group ownership of directory/file
  - Same arguments as above

- Permissions represented as inode data structure

- Three sections:
  - User permissions : owner
  - Group permissions : group-wide
  - Other permissions : system-wide

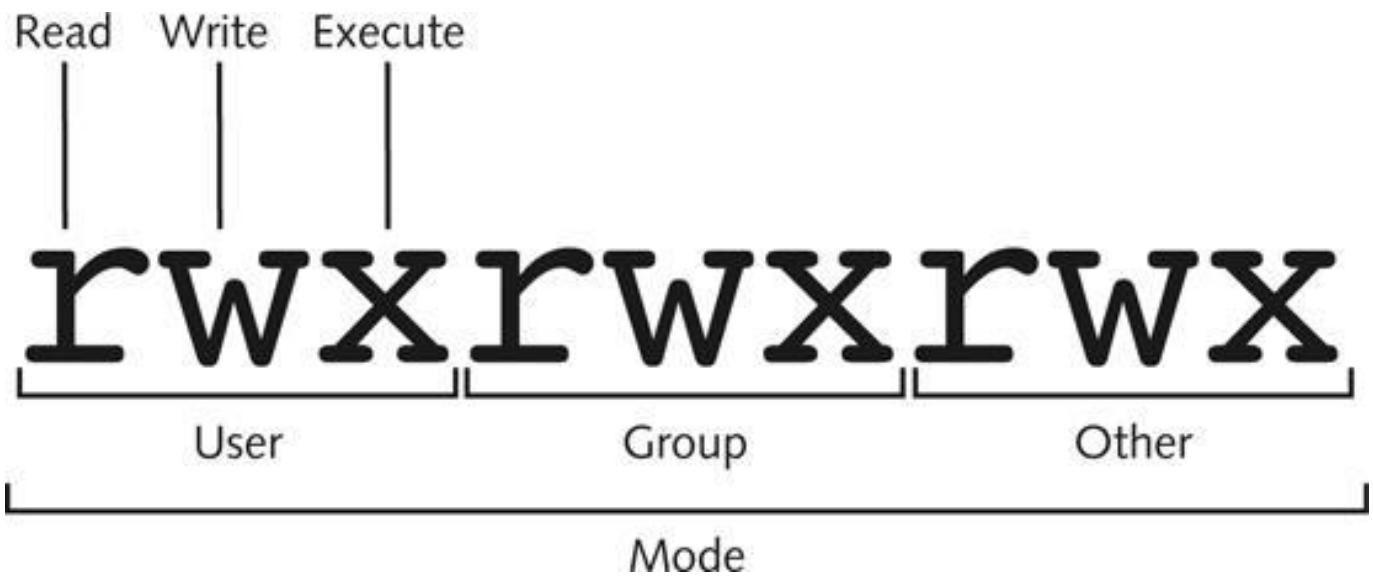
Permissions

Read	:	r	:	4
Write	:	w	:	2
Execute	:	x	:	1
No permission	:	-	:	0

octal

- Can be applied to three classes of users
  - Owner
  - Group
  - Other (Everyone)

Mode




```
[fredericks@SciLinux6 ~]$ ls -l
total 44
drwxr-xr-x. 2 fredericks fredericks 4096 Jul 20 11:59 Desktop
drwxr-xr-x. 3 fredericks fredericks 4096 Sep  9 14:39 Documents
drwxr-xr-x. 2 fredericks fredericks 4096 Sep 14 09:48 Downloads
```

Mode

- User (Owner)
  - Owner of a file or directory
- Group
  - Users with ability to change permissions on a file or directory
- Other
  - Refers to all users on system
- Permissions are not additive
  - The system assigns the first set of permissions that are matched in the mode order: user, group, other
- Linux permission should **not** be assigned to other only

Permission Actions

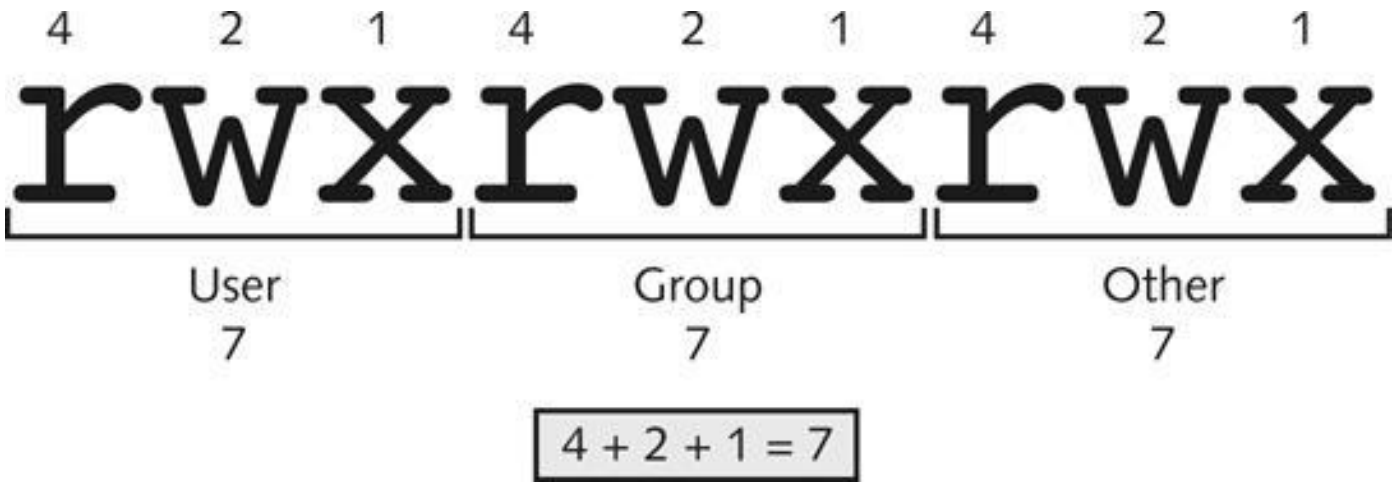
Permission	Definition for Files	Definition for Directories
Read	Allows a user to open and read the contents of a file	Allows a user to list the contents of the directory (if she has also been given execute permission)
Write	Allows a user to open, read, and edit the contents of a file	Allows a user to add or remove files to and from the directory (if she has also been given execute permission)
Execute	Allows a user to execute the file in memory (if it is a program file or script)	Allows a user to enter the directory and work with directory contents

chmod

- Change mode (permissions) on file or directory
- Two (minimum) arguments:
  - Criteria (updating permissions)
  - Filenames

- **chmod** 755 *file1.txt*
- **chmod** u+x *file2.txt*

Permissions



Decoding Permissions

- **chmod** *755 file1.txt*

Mode (One Section Only)	Corresponding Number
rwx	4 + 2 + 1 = **7**
rw-	4 + 2 = **6**
r-x	4 + 1 = **5**
r--	**4**
-wx	2 + 1 = **3**
-w-	**2**
--x	**1**
---	**0**

Updating Permissions

- **chmod** u+x *file2.txt*

Category	Operation	Permission
u (user)	+ (adds a permission)	r (read)
g (group)	- (removes a permission)	w (write)
o (other)	= (makes a permission equal to)	x (execute)
a (all categories)		

Default Permissions

- New files given **rw-rw-rw-** by default
- umask – defines default permissions
  - Value subtracted from octal value of mode
  - Masks out permissions you don't want to provide
  - If a file is 755, and a umask of 022 is used, the result is:
    - 733 => rwx-wx-wx

umask

- umask 022

666

	New Files	New Directories
Permissions assigned by system	rw-rw-rw-	rwxrwxrwx
- umask	0 2 2	0 2 2
= resulting permissions	rw-r--r--	rwxr-xr-x

777

644

755

Special Permissions

- Sticky bit
  - Disallow others from renaming/deleting files in directory
  - Only owner (and root) can rename/delete
  - Indicated with 't' in mode
    - e.g., drwx rwx rwx**t**
    - **chmod** +t *directory*
- Why?
  - /tmp – accessible by all users
  - Sticky bit prevents other users from deleting working files in /tmp

Special Permissions

- SUID (set user ID)
  - If set on a file, user who executes the file becomes owner of the file during execution
    - Basically, user ***inherits*** owner's permissions
    - e.g., `passwd` command
      - Why?

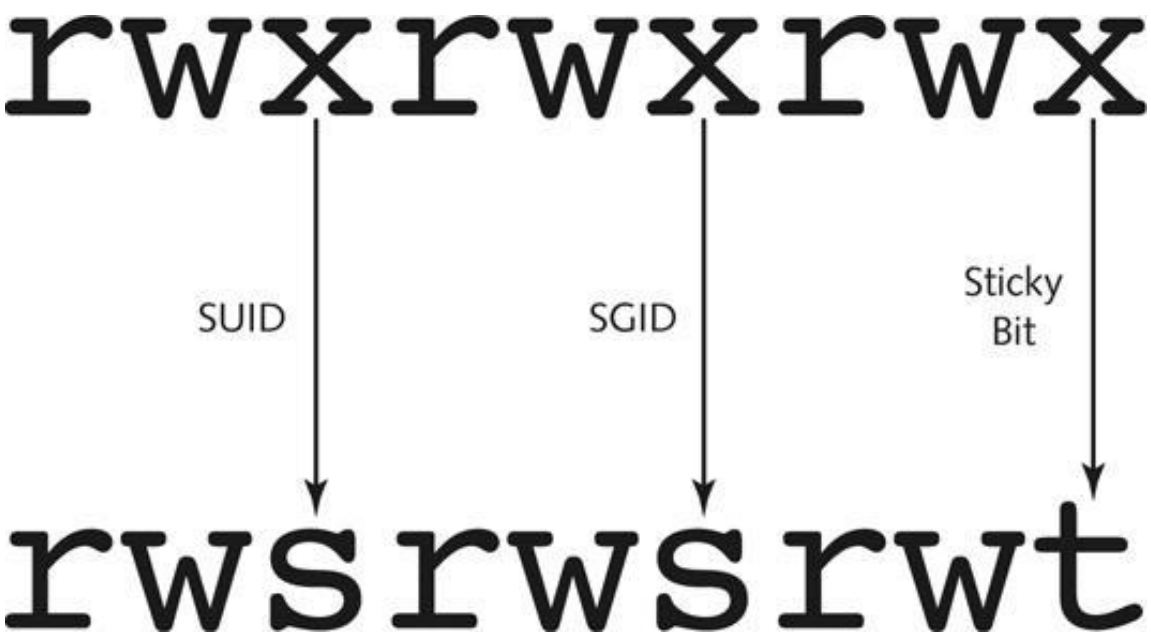
  - **chmod** *u+s file1*
    - rw**S**rwxrwx → no execute permission
    - **chmod** *u+x file1*
    - rw**s**rwxrwx

Special Permissions

- SGID (set group ID)
  - Applicable to files and directories
  - If set on a file, user who executes the file becomes member of the file's group during execution
  - If a user creates a file in a directory with SGID set, the file's group owner is set to be the directory's group owner and not the user's primary group
  - (Same as SUID but for a group)

  - **chmod** g+s *file.txt*
    - rwxrw**S**rwx

Special Permissions



Common Settings

- Home directory
  - 700 (only user has access)
  - rwx --- ---
- Web server (Apache)
  - 755 (common) / 750 (single user development)
  - rwx r-x r-x / rwx r-x ---
  - Create group if developing on a team (775)
    - If developing a website, its best to work on a dev server and simply push to the production server
- /etc/passwd
  - 644
  - rw- r– r--

vSphere Demo

- Access Linux desktop via web interface
  - As with SSH, must be on OU VPN

https://vcenter6.secs.oakland.edu/

vSphere

- vCenter Inventory Lists
  - Virtual Machines
    - <your machine>
      - Right click → Open Console

OS Installation

- By default, your system boots to a working copy of Linux
- CTO has provided you with a mounted copy of the Scientific Linux ISO
  - In case if “something bad” happens, you can reinstall

THE END

- Homework #1 due at 11:55pm **Friday**
- Start getting familiar with the command line!
  - Make sure you can connect to the VM using a CLI
    - (And make sure you can connect from off-campus using the VPN!)