

An Introduction to Operating Systems and Linux

By Ibrahim Fares

Headlines

- ▶ Introduction to OS
- ▶ Selecting a proper OS
- ▶ A Tour About OS'S :
 - Proprietary OS'S
 - Free-Open Source OS'S

Operating System...

- ▶ For the ordinary user , the Operating System (OS) is the user interface (frontend)



Operating System...

- ▶ What the end user sees is just the Operating System front end user interface (Desktop)
- ▶ Behind the scene there is the KERNEL



- ▶ The Kernel runs a lot of functionality in the background and performs a lot of tasks for the user and the developer, and abstracts a lot of HW details



FUNCTIONS OF OPERATING SYSTEM

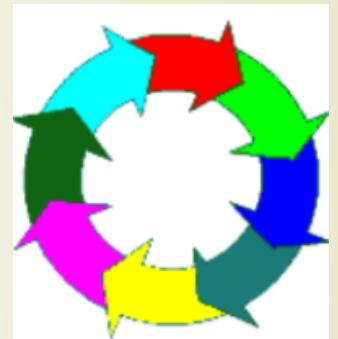
User Management



- ▶ The kernel supports multiple users using it
- ▶ Each user has his own resources and privileges
- ▶ There is the super-user (root/admin) that have full control on the system
- ▶ User privileges decides his rights on his owned resources and other users resources
- ▶ The kernel manages user privileges and protects each user resources from unauthorized access by other users
- ▶ The kernel manages also the system environment for each user

Process Management

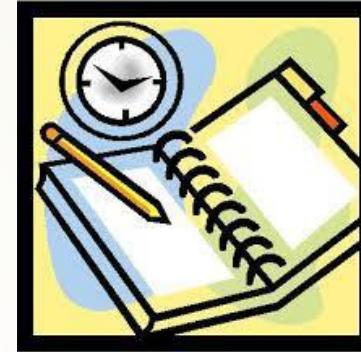
- ▶ The kernel enables the system to run multiple applications (processes) at the same time
- ▶ Each Application thinks that it owns the whole system
- ▶ Each process may even have multiple threads that run simultaneously
- ▶ The kernel has a **SCHEDULER** that provides time slots to each process/thread in a pre-determined order to enable multi-tasking
- ▶ The order of time slot assignment and the time slot size is based on the **SCHEDULING ALGORITHM**



Process Management

► Process Scheduling Algorithm

- There are multiple scheduling techniques :
 - ✓ Round robin
 - ✓ Priority Based
- Processes may block on events such as hardware job completion, other process will take the processor cycles



Process Management Preemption

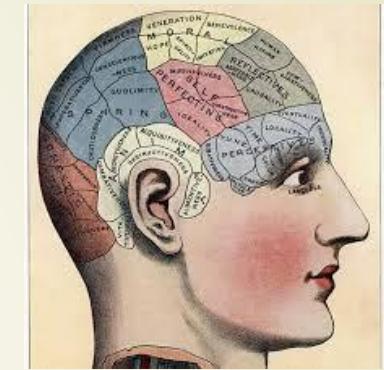
► The scheduler can be:

- Pre-emptive
 - ✓ A higher priority task can interrupt the low priority task before it completes its job
- Non-preemptive
 - ✓ Once a process takes control, it has to finish its job before it releases control



Memory Management

- When the processor is running multiple processes, each process will need to have its own memory
 - The kernel works with the Processor to provide two functionalities:
 - Memory Protection (MPU)
 - Protects each process memory from being corrupted by the other processes
 - Memory Management (MMU)
 - Each process will think that it owns all the system memory



File-System Management



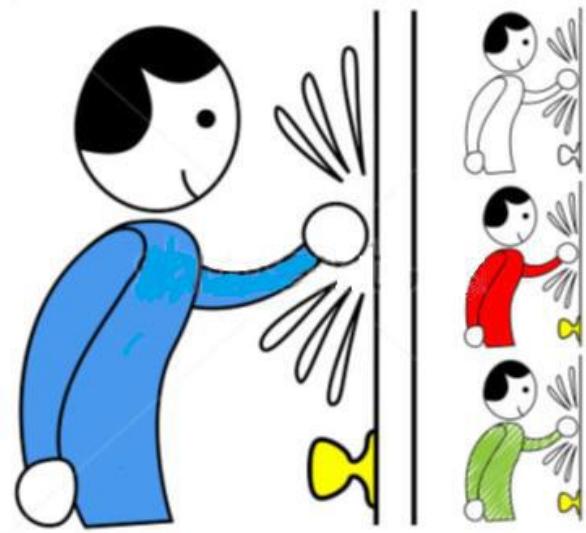
- ▶ The kernel performs jobs related to handling of storage devices
- ▶ It gives the applications the feel of files and directories and hide the hardware details behind all of that

Power Management



- The kernel handles power related functionality such as:
 - Managing sleep modes of the processor
 - Adjusting the processor clock rate based on its load

I/O and Interrupt Handling



- ▶ Processor is connected to several I/O devices
- ▶ Those devices may use Interrupts to get the processor/kernel attention
- ▶ Kernel handles the interrupts received from Input/Output devices

Timers and Time Related Tasks

- ▶ The Processor is interrupted per time tick
- ▶ Kernel uses this interrupt to keep track of time
- ▶ Tick duration defines the time resolution for the kernel/system
- ▶ Kernel is responsible to keep track of time:
 - Handle timers
 - Handle time of day clock
 - Handle deferred jobs



Networking and Communication



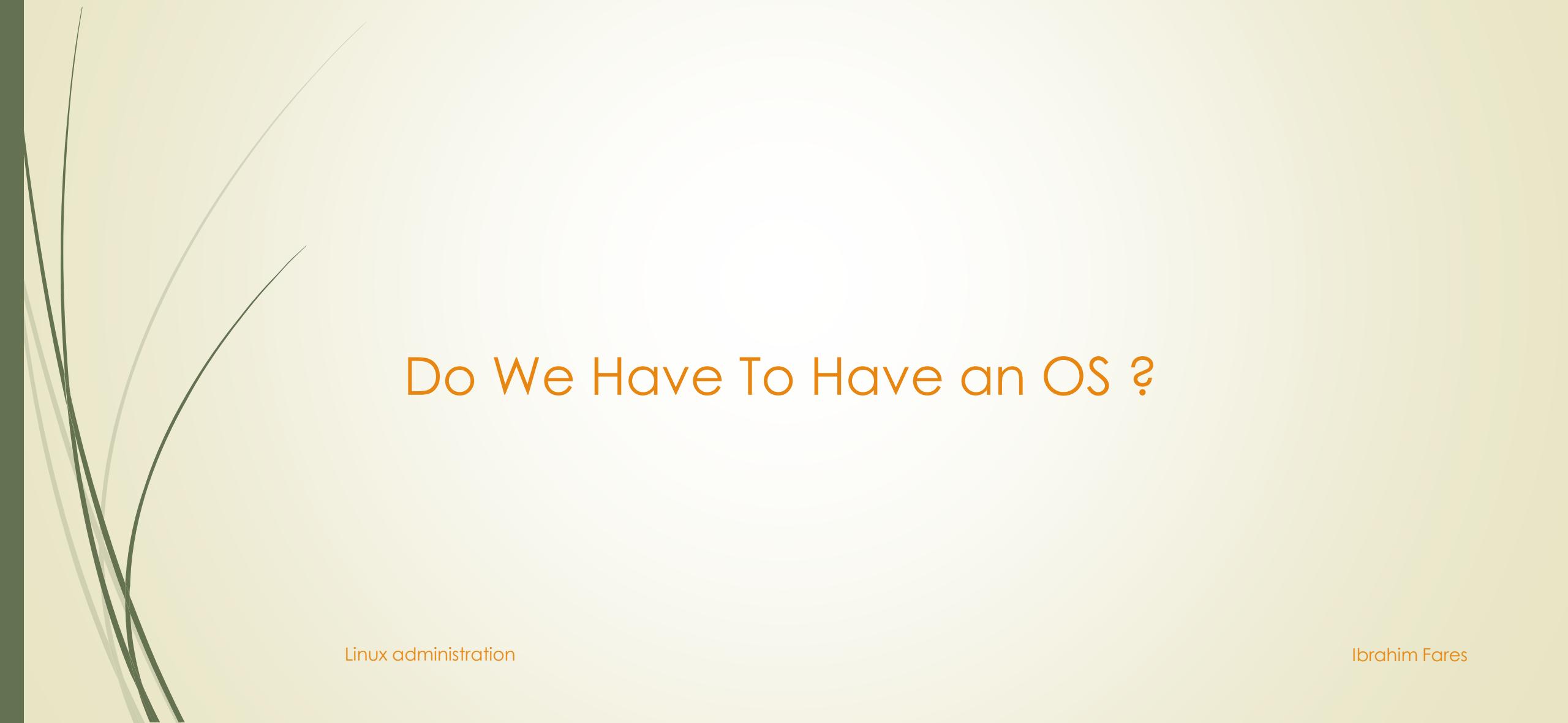
- The kernel is responsible to enable communication via:
 - Wired / wireless networking
 - USB/SDIO/Serial/...
 - Other communication means

Summary



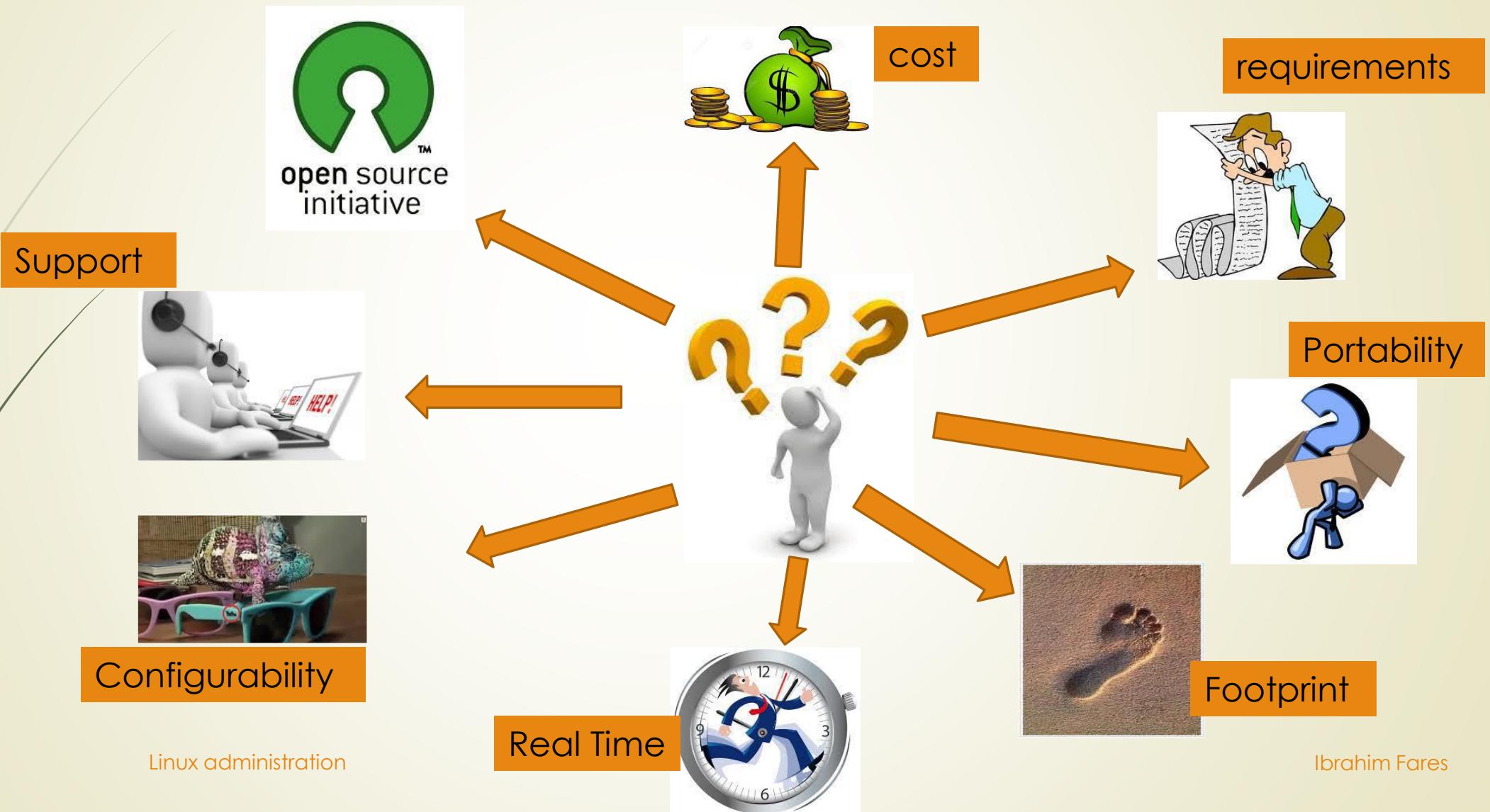


Task-1



Do We Have To Have an OS ?

How We Select an OS





A Tour About OS'S

1-Proprietary RTOS'S

Windows CE

- ▶ Developed by Microsoft
- ▶ Has recently been replaced by Windows Embedded Compact
- ▶ Has the mix of the windows desktop and embedded devices requirements
- ▶ Suitable for embedded devices that require also a good user interface
- ▶ Medium to Large HW requirements:
 - Footprint of OS around 200-350KB
 - May run in under a 1MB of memory
- ▶ True RTOS, has a deterministic behavior
- ▶ Cost is around \$3 per device license + Initial Cost for tools



Application Examples



Linux administration

Ibrahim Fares

Nucleus

NUCLEUS

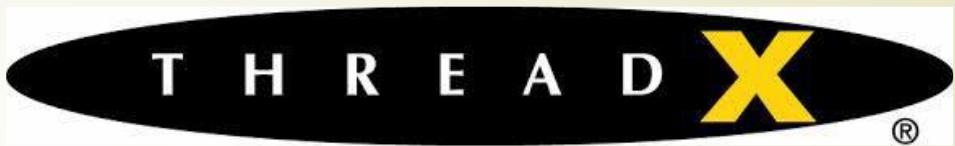
- ▶ Developed by Mentor Graphics
- ▶ Portable to different devices (Microcontrollers and Micro - Processors)
- ▶ Foot print as small as 2KB
- ▶ Support unique power management features
 - Power/clock gating
 - Deep sleep modes
- ▶ Cost is about
 - \$13K for the OS license per implementation
 - \$3K per seat for the tools

NUCLEUS Application Examples

- Very widely used



ThreadX



- ▶ Developed by Express Logic, Inc. by the same the author of Nucleus
- ▶ Small footprint (can go as low as 2KB on Microcontrollers)
- ▶ True RTOS, Fast and deterministic behavior
- ▶ Fast boot time (This is very important for devices that can't afford a long boot time)
- ▶ Flexible licensing models



Application Example

- ▶ Hewlett-Packard (HP) use ThreadX in all of their printers,
 - Inkjets
 - LaserJets
 - All-in-One



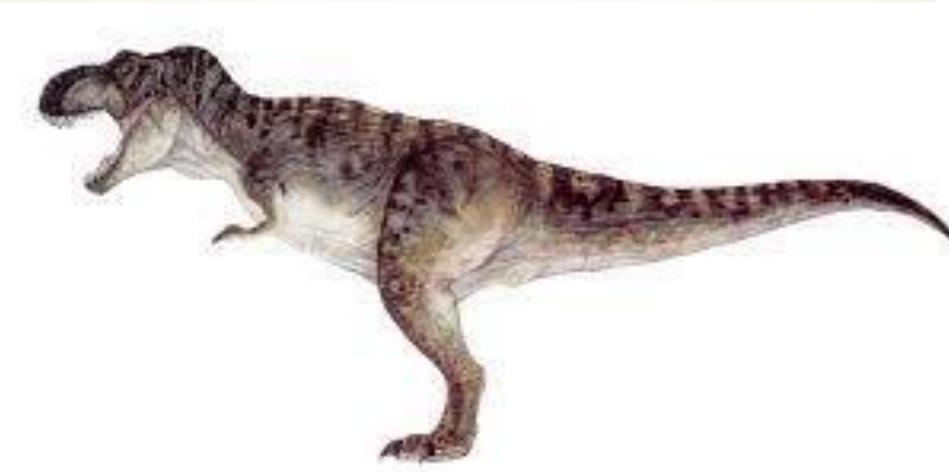


Application Example

- Broadcom use ThreadX in a lot of its Chips for cell phones in their communication processors



VxWorks



VxWorks®



WIND RIVER

- ▶ Developed by Wind River Systems, 1987
- ▶ Legacy RTOS. Used to be the top OS for Embedded Platforms
- ▶ Not used recently for small system, but still considered by heavy Institutions (such as Aerospace and Defense)
- ▶ Expensive licensing cost + Royalty cost
- ▶ Other OS's with their latest development are catching up



Application Example



KUKA



Linux administration

Ibrahim Fares



Application Example



Linux administration

Ibrahim Fares



A Tour About OS'S

2- Free-Open Source OS'S

Is Free Really Free ??

- ▶ Free for freedom and not for free drink
- ▶ Free means freedom to run, study, copy, modify, distribute both the original code and the modified versions
- ▶ Free/Open source OS's include:
 - The OS itself (the kernel)
 - Development tools: Compilers, Debuggers, toolchains, ...
 - Support libraries: C library, XML parsers, Security, command line utilities
- ▶ Keep in mind the licensing risks (GPL, LGPL, ...)
 - Licensing terms
 - Might mean you can not have a proprietary system
 - Might mean you need to give up your source code

Advantages of Open Source OS

- ▶ Cost perspective
 - Kernel, development tools, associated libraries and support tools, applications, ...
- ▶ Big community help and support
- ▶ Ability to understand and control details of operation
- ▶ Availability of online resources and documents
- ▶ Fast release cycles
- ▶ Ability to influence the future of the OS

Linux (www.kernel.org)

- ▶ Most popular Open Source OS
- ▶ Big Community Support
- ▶ Distributions
 - Build your own distribution
 - Redhat
 - Suse
 - Ubuntu
 - And many others
- ▶ GPL v2.0 License
- ▶ Require 32bit processors with MMU
- ▶ Highly configurable, RAM and Storage requirements according to configurations



eCos (<http://ecos.sourceforge.org>)

- ▶ Suitable for smaller embedded systems
- ▶ Small footprint, with low requirement on resources
- ▶ Modified GPL License to give developer more rights to use it without need to GPL his code
- ▶ One of the most popular open source RTOS in the world today
- ▶ Highly configurable (eCos stands for the "Embedded Configurable Operating System")
- ▶ Support multi-tasking with priority-based real-time scheduler
- ▶ True RTOS, Highly deterministic behavior
- ▶ Highly portable to most of 32 bits processors





ecos Application Example

- Sony, Playstation 3 Gaming Console for WiFi support





ecos Application Example

- ▶ Samsung, latest LCD HDTVs (32-70 inch) to support multi-media playback via USB2 from cameras, mp3 players and flash drives





ecos Application Example

- ▶ Eye-Fi Wireless SD Cards that use WiFi to transfer images between camera and computer



Nuttx



- ▶ First released by Gregory Nutt in 2007 with BSD license
- ▶ True RTOS (deterministic)
- ▶ Small footprint
- ▶ Supports from simple 8 bit processors (such as Zilog Z80) to 32 bit processors (such as ARM-9, x86)
- ▶ Supports C & C++ development
- ▶ TCP/IP Stack and USB support
- ▶ Support multitasking with FIFO and round-robin scheduling
- ▶ Support pre-emption
- ▶ Highly configurable
- ▶ Used in MP-3 players, Auto-pilot, ...



Application Example





Introduction to Linux

By Ibrahim Fares



Linux/Unix

HISTORICAL BACKGROUND

UNIX



Bell Laboratories



- ▶ It started in Bell Labs with a terminated project for the Multics Multi-user operating system
- ▶ Dennis Ritchie and Ken Thomson started to work on Unix
- ▶ 1969: First Implementation of Unix, done on PDP-7 using assembly
- ▶ 1971: Ported Unix to PDP-11 using assembly
- ▶ 1972: Creation of “C” programming language to facilitate the porting
- ▶ 1973: Complete rewrite of UNIX into “C” (which led to high portability)
- ▶ First Public version, Unix System V6
- ▶ 1977: Unix System III
- ▶ 1982 Unix System V (by AT&T)





UC Berkeley

- ▶ BSD (Berkeley Software Distribution)
- ▶ 1979: 3BSD
- ▶ The Series 4BSD (4.0, 4.1, 4.2, 4.3 BSD)
- ▶ 1993: 4.4 BSD
- ▶ Dragonfly BSD, Free BSD, Net BSD, Open BSD



Linux administration



Ibrahim Fares

Commercial Unix

- ▶ Other companies built their variants
 - Digital → Tru64
 - HP → HP-UX
 - IBM → AIX
 - Sequent → DYNIX/ptx
 - SGI → IRIX
 - Sun Microsystems → Solaris



The Unix Fever....

- Unix has widely spread in universities and industry because,
 - Written in C (portable to different architectures)
 - Simplicity and elegant design:
 - Less number of System calls (compared to other OSs)
 - Almost everything is treated as a file
 - Fast Process creation using fork-exec concepts
 - Strong Inter-Process Communication (IPC) support
 - Powerful, Robust, and Stable



GNU GNU Not Unix



Linux administration



gzip



Ibrahim Fares



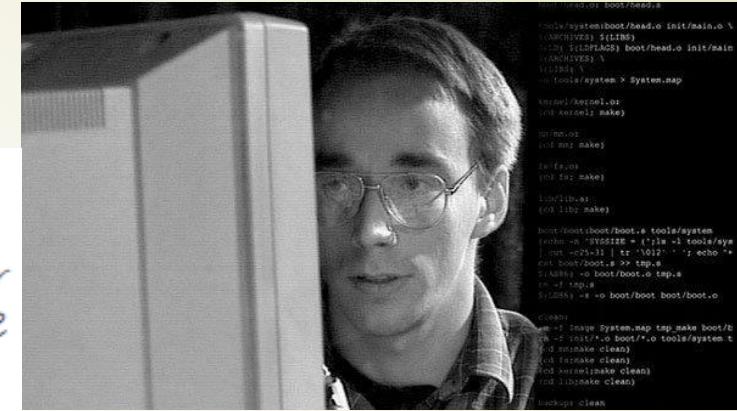
- ▶ Richard Stallman is a software freedom activist and a computer programmer
- ▶ Believing in free software, he formed the Free Software Foundation and started the GNU project in 1983
- ▶ The target with the GNU project was to create a Unix-like operating system along with all the eco-system based on free software
- ▶ He is still active in advocating for free software and campaigning against software patents and digital right management (DRM)



- ▶ Created the GPL license (GNU Public License)
- ▶ Stallman pioneered the concept of copyleft,
 - If you modify a free software (GPL) then your modifications will have to be also GPL
- ▶ He started by building the eco-system but lacked the kernel

Then Comes Linux

"Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix. it probably never will support anything other than AT-harddisks, as that's all I have :-(. "



- ▶ Developed by Linus Torvalds (University of Helsinki) on an i-386 platform
- ▶ First public introduction in 1991
- ▶ Unix-like operating system kernel together with GNU Software and tools
- ▶ Open Source, received contribution by many
- ▶ Under the GNU GPL 2.0 License

Examples for Linux Platforms

- AMD x86-64
- ARM
- Compaq ALPHA
- CRIS
- DEC VAX
- H8/300
- Hitachi SuperH
- HP PA-RISC
- IBM 3/390
- Intel IA-64
- MIPS
- Motorola 68000
- FreeScale PowerPC
- Sparc, Ultra-Sparc
- V850

And the list keeps growing

System Requirement

- ▶ Linux was initially built on x86 architecture, but now it and tools and libraries associated with it support a large range of architectures
- ▶ The mainstream Linux requires at least 32 bit processors with support of MMU (Memory Management Unit), some special configurations can support MMU-less architectures
- ▶ Linux is highly configurable, developer can adjust its configuration based on the need and available hardware
- ▶ Required RAM depends on the configuration, but 8MB is a typical minimum requirement, and 32MB is a typical average requirement
- ▶ Required Storage also relies on configurations, but 4MB is a typical minimum for small systems.



18.04

Year

Release month

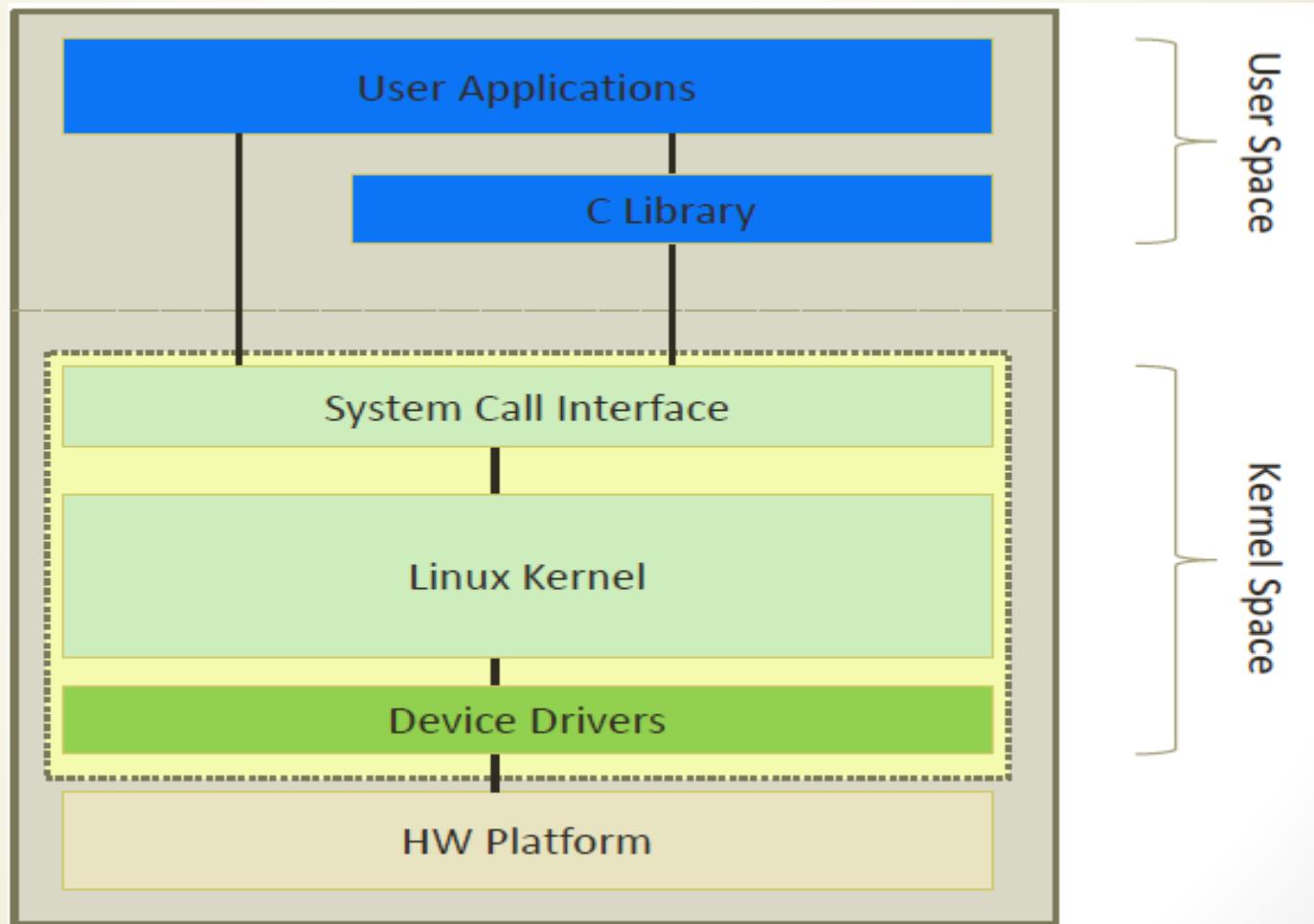
A large black number '18.04' is centered. Two orange arrows point upwards from the bottom of the '18' to the word 'Year' and the bottom of the '04' to the word 'Release month'.

Linux Distributions

- ▶ You can build your own distribution (not an easy job, but sometimes it is worth it)
- ▶ If interested check the Linux From Scratch Project (<http://www.linuxfromscratch.org/>)
- ▶ Otherwise, you can use readily available distributions
- ▶ There are a lot of distributions available, most popular now is the Ubuntu

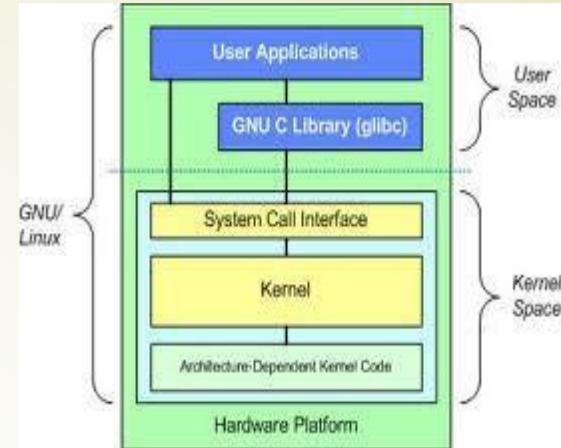


Linux Model



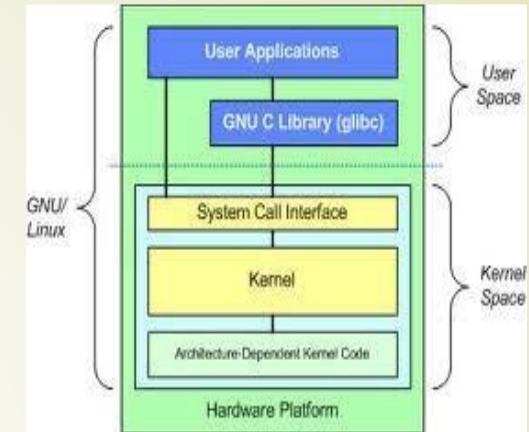
User Space Apps

- ▶ These are the typical applications
- ▶ Uses the “C” library in a lot of functionality
- ▶ Linux uses the MMU capability to protect apps from corrupting each other memory
- ▶ User apps can not access the hardware (screen, I/O, files, ...) directly
- ▶ Must use “System Calls” into the kernel which perform the task on behalf of the user application



Kernel Space

- ▶ Have direct access on all system resources
- ▶ Perform tasks on behalf of the user app in System Calls
- ▶ Dangerous ... can easily crash the system
- ▶ No access to “C” library
- ▶ Handles Interrupts and mission critical jobs
- ▶ Connects with hardware via device drivers





► So, What We will use !



Ubuntu



Installing Ubuntu

Linux administration

Ibrahim Fares

Choices for Ubuntu Installation

1. Booting from Ubuntu DVD
2. Installing Ubuntu alone on the machine
3. Installing Ubuntu along with Windows in a dual boot configuration
4. Installing Ubuntu inside a virtual machine
5. Other choices....

Choices for Ubuntu Installation

1. Booting from Ubuntu DVD
2. Installing Ubuntu alone on the machine
3. Installing Ubuntu along with Windows in a dual boot configuration
4. Installing Ubuntu inside a virtual machine
5. Other choices....

Virtualization

What ??



Virtualization

Why ??

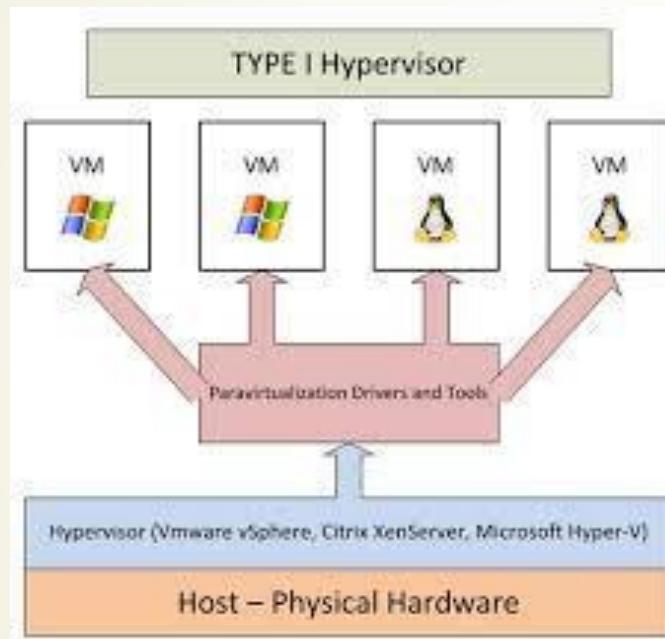
- ▶ Virtualization is used for :
 - Running Multiple Operating Systems simultaneously
 - Easier Software Installations
 - Testing and Disaster Recovery
 - Infrastructure Consolidation

Virtualization

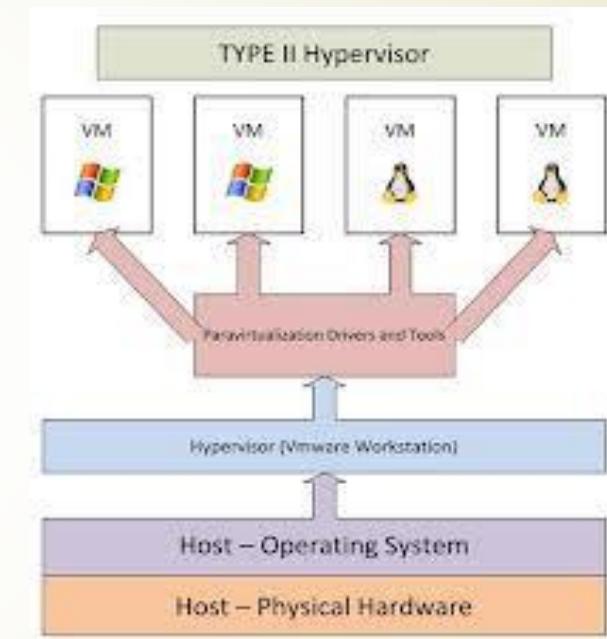
How ?

- ▶ We can achieve virtualization via the use of a Hyper-visors
- ▶ Hyper-Visor is a software that can accommodate multiple virtual machines. Each VM can have its own OS
- ▶ There are two types of Hyper-visors,
 - Type One (Bare-Metal Hyper-Visor)
 - Type Two (Use of host OS)

Virtualization How ?



Hyper-Visor Acts as a light host OS



Hyper-Visor Acts as an Application



Linux For Embedded Systems

For Arabs

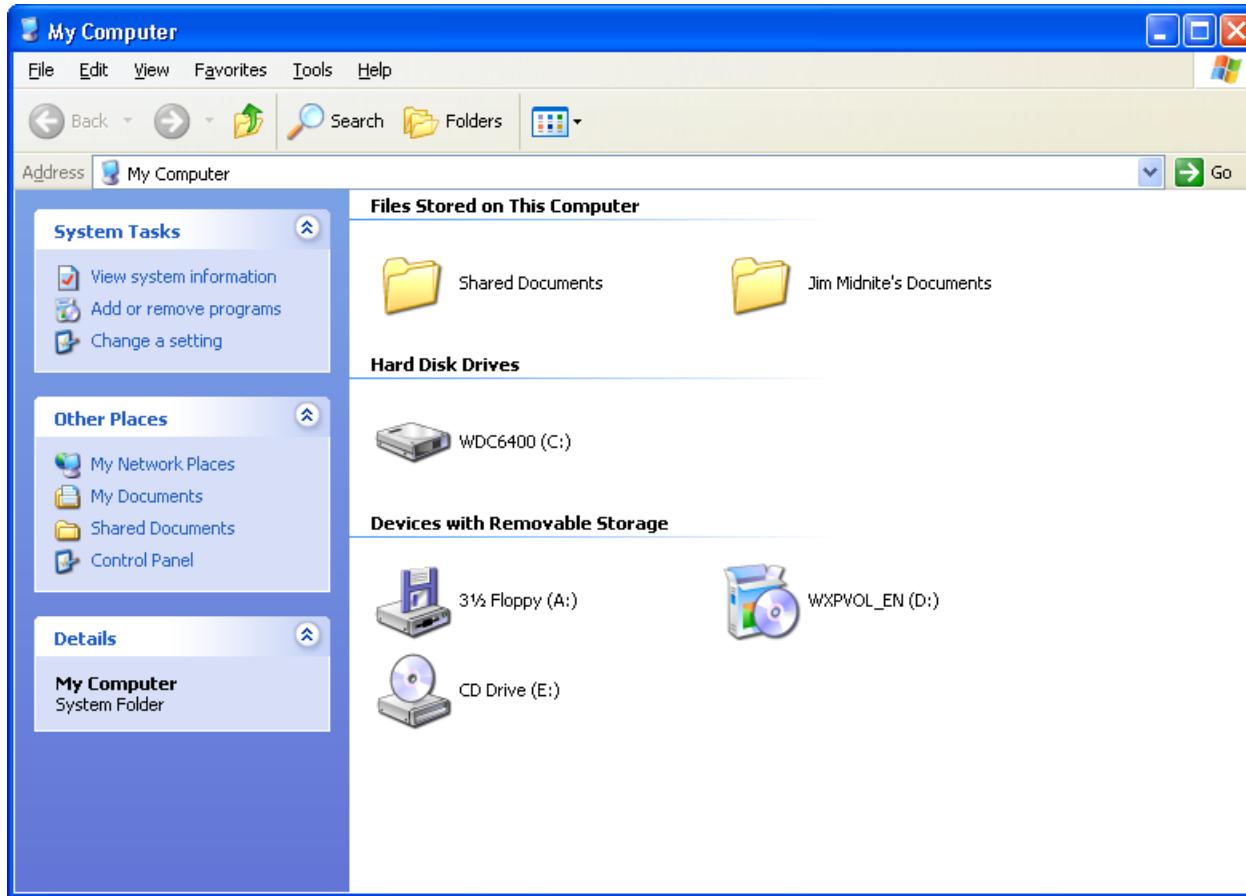
Course 102: Understanding Linux

Ahmed ElArabawy

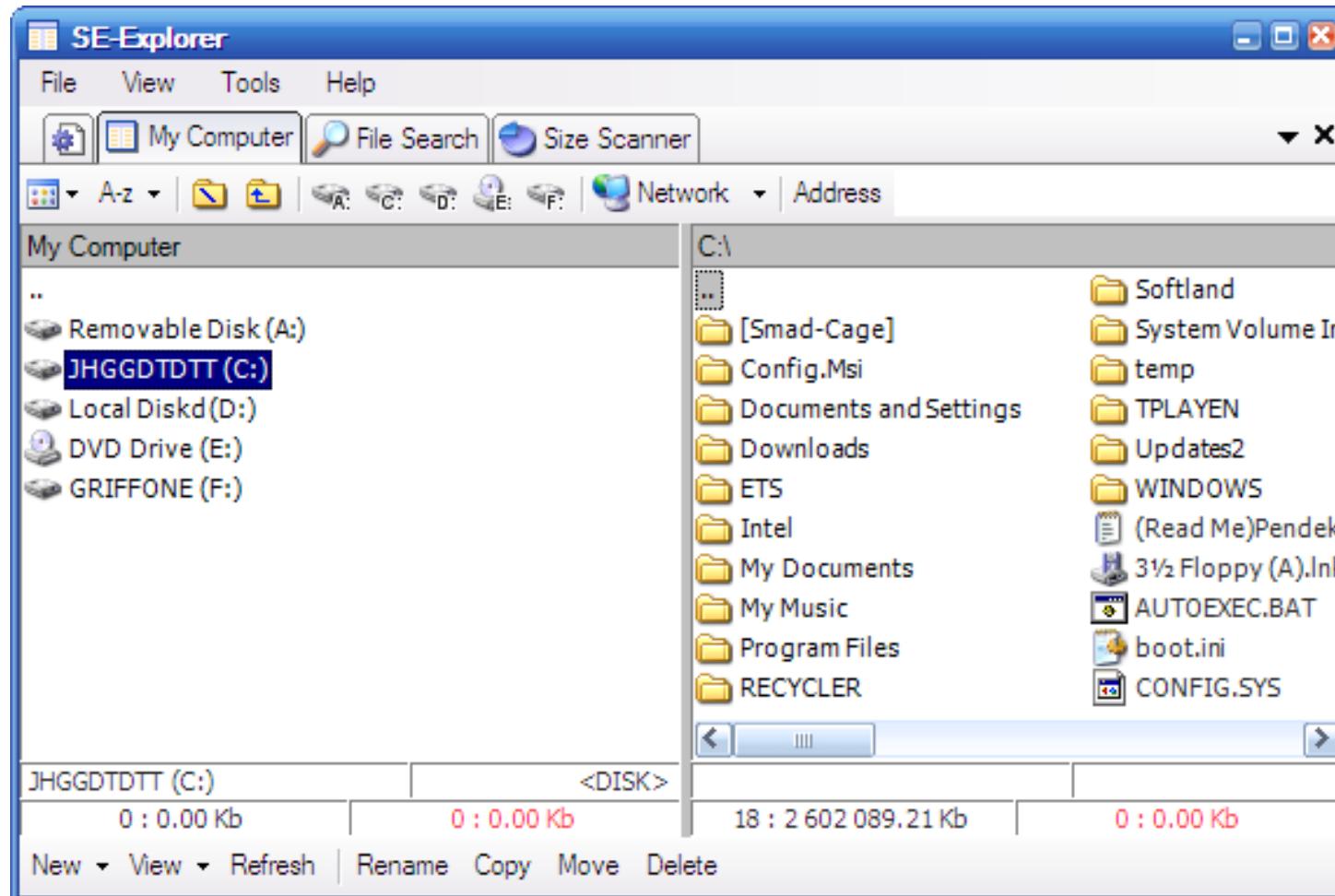


Lecture 2: Unwrapping Linux

File System Layout (Windows)



File System Layout (Windows)



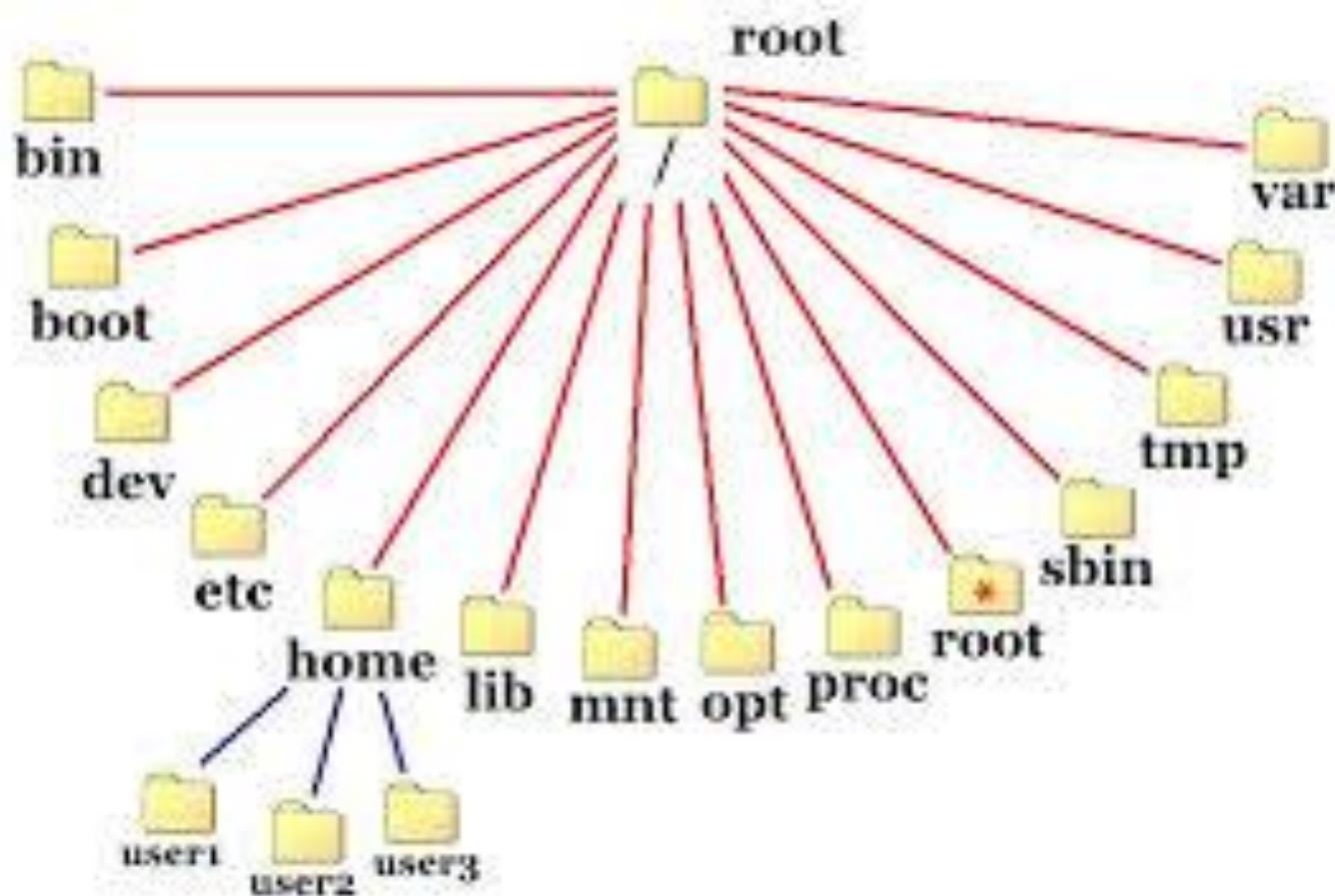
File System Layout (Linux)



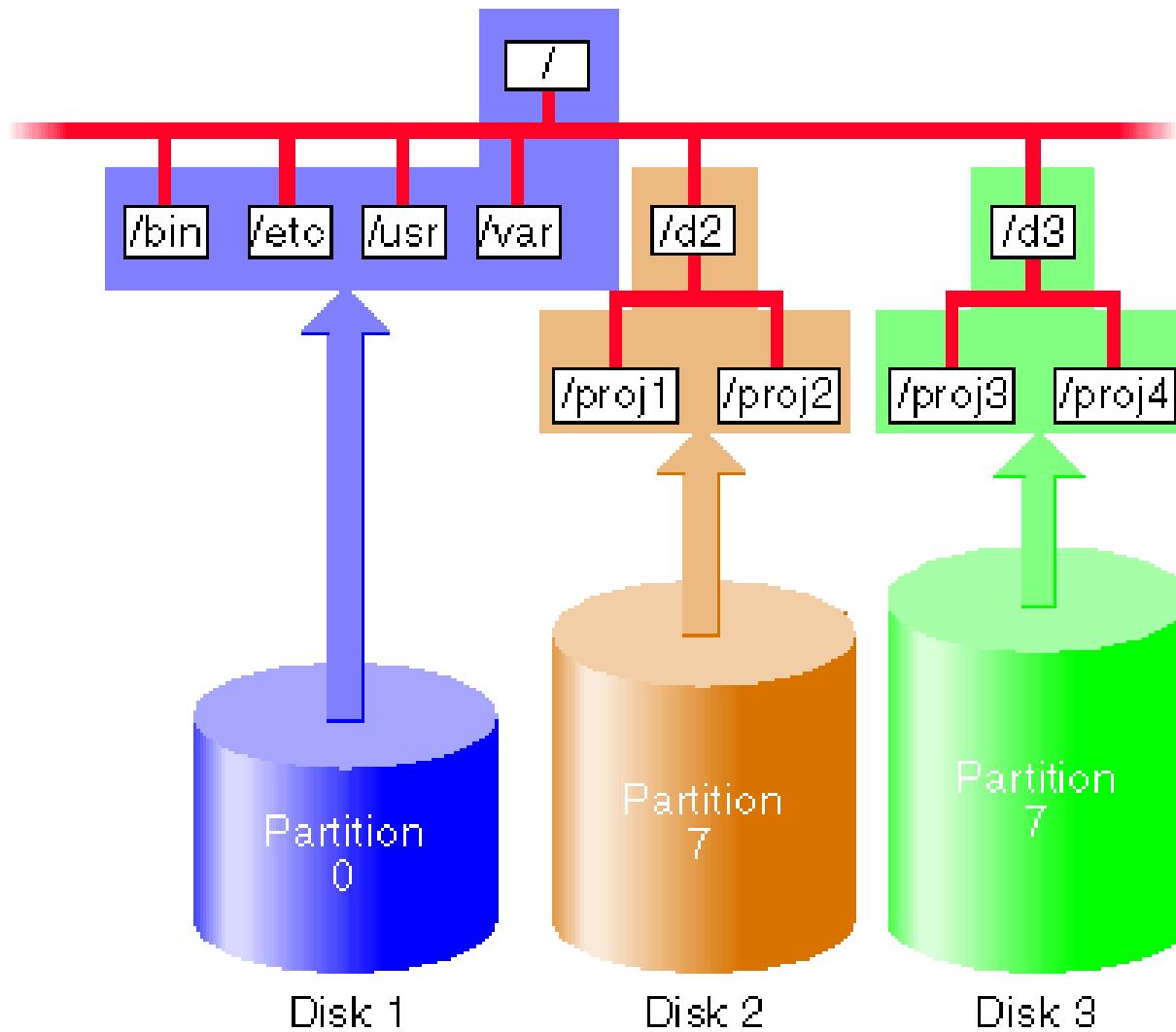
The screenshot shows a file manager window with the following details:

- Left sidebar:** A tree view of the file system under the "tei" directory. The visible branches include Home, File System, dtds, and tei. The "tei" branch is expanded, showing experimental, old, p4x, xml, ulm, etc., and a large number of sub-directories and files under tei.
- Right pane title:** "dtbs tei p4x".
- Content area:** A grid of icons representing various XML-related files and directories. The files include celt2.dtd, catalog, celtdoc, zip, dtd.zip, isoamsb.ent, isoamsc.ent, isoamsc2.ent, isolat1.ent, p3x.dtd, teiback2.dtd, teicorp2.dtd, teidsgis.dtd, and several others ending in .ent or .xsd. The "celt2.dtd" file is highlighted with a red border.
- Bottom status bar:** "celt2.dtd selected (196.0 kB), Free space: 63.3 GB".

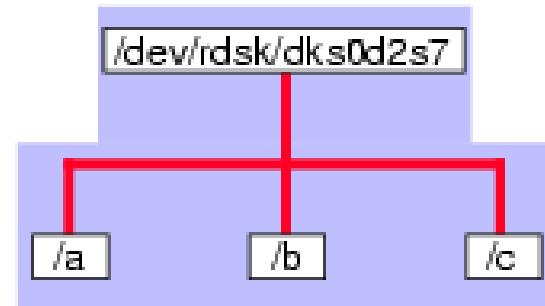
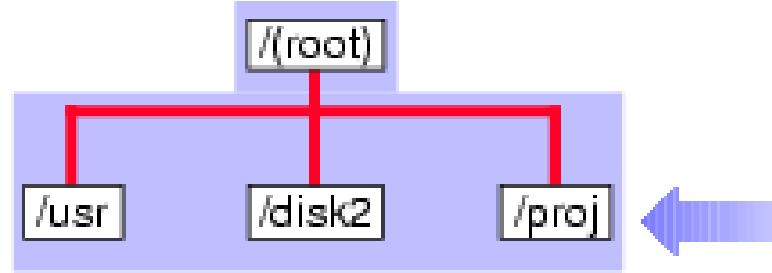
File System Layout (Linux)



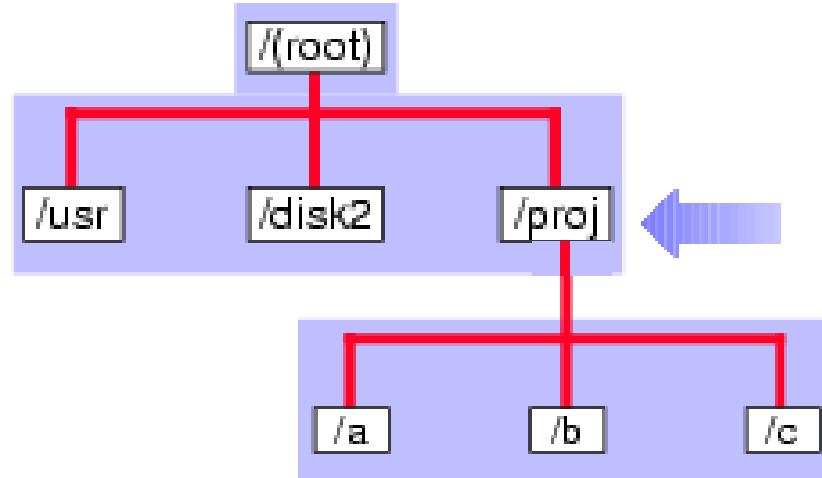
Mounting Devices



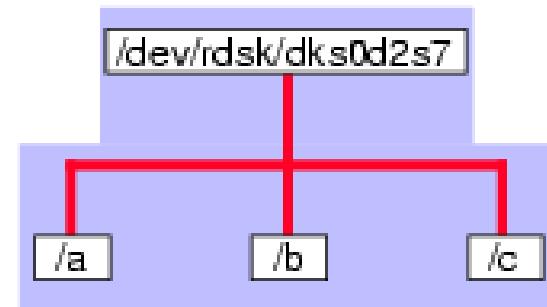
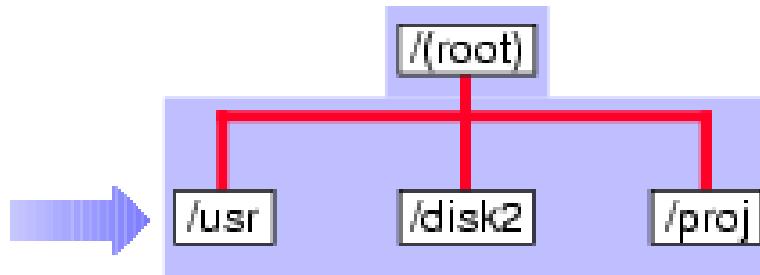
Choosing The Mounting Point



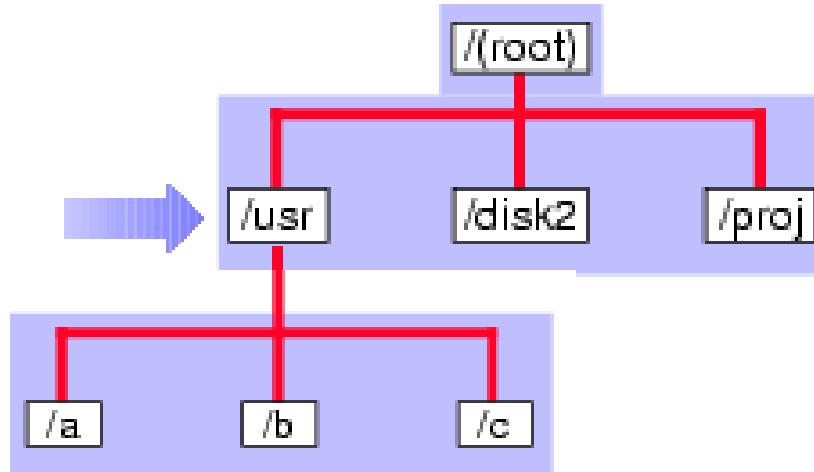
Choosing The Mounting Point



Choosing The Mounting Point



Choosing The Mounting Point



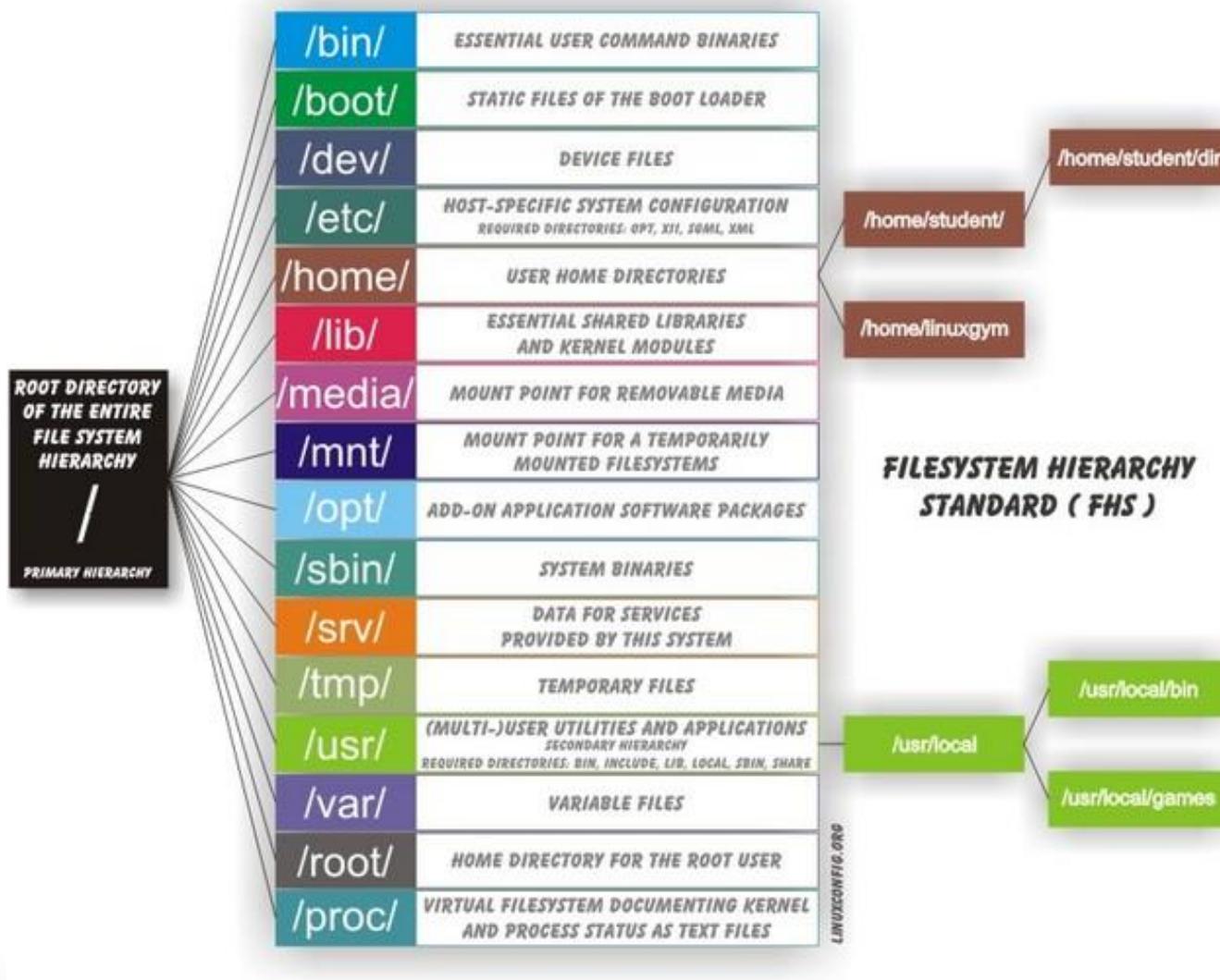


Linux File-System Layout

A major difference Between Windows and Linux in file structure

- In Windows,
 - Each partition of drive will have its own separate tree.. (C, D, E,...)
 - Plugging a flash or a portable hard-disk results in a new tree
- In Linux,
 - We have a single tree (unified file-system)
 - The head of the tree is called the root ‘/’
 - Plugging a device will add a branch (or sub-tree) to the file-system ... at the selected mounting point

Linux File-System Hierarchy



Linux File-System Hierarchy

Directory	Role
/	Root Directory, the head of the tree
/home/	Contains the home directories of users /home/aelarabawy (my home directory) Each user can put his files under his home directory
/root/	The home directory for the super user (admin)
/usr/	Files (programs, libraries, documentation, etc..) used by all users in the system Examples: /usr/bin/ binaries used by all users binaries included with the distribution /usr/local/ files that don't come with the distribution and installed by the user /usr/sbin/ system files that come with the distribution /usr/share/ shared data by programs in /usr/bin/



Linux File-System Hierarchy

Directory	Role
/boot/	<p>Boot loader, Linux Kernel, Startup files</p> <p>Examples:</p> <p><code>/boot/vmlinuz/</code> Linux kernel image <code>/boot/grub/grub.conf</code> Bootloader configuration file <code>/boot/initrd/</code> Startup Files used in booting</p>
/bin/	Common programs, shared by the system, the system administrator and the users
/sbin/	System binaries Programs for use by the system and the system admin
/lib/	Shared libraries
/opt/	Optional software; Extra and third party software

Linux File-System Hierarchy

Directory	Role
/dev/	<p>Device nodes; not a regular file</p> <p>Each device in the system is represented by a file, so to write to the device, we write to this file, to read from a device, we read from this file</p>
/etc/	<p>System configuration files (similar to those in the Control Panel in Windows)</p> <p>Examples:</p> <ul style="list-style-type: none"> /etc/fstab contains a list of storage devices and their associated mount point /etc/passwd contains list of user accounts
/var/	<p>Storage for variable files and temporary files created by users, such as log files, the mail queue, the print spooler area, space for temporary storage of files downloaded from the Internet, or to keep an image of a CD before burning it.</p> <p>Examples:</p> <ul style="list-style-type: none"> /var/log log files /var/log/messages system log file

Linux File-System Hierarchy

Directory	Role
/net/	Standard mount point for entire remote file systems
/media/	contains the mount points of removable media devices such as CD-ROMs and USBs (that are mounted automatically)
/mnt/	Standard mount point for external file systems, e.g. a CD-ROM or a digital camera.
/proc/	Does not contain stored files, it is just the front of some devices feedback (reading from it, is like asking a driver to provide information) Used to read information about system resources.
/sys/	Same as /proc/
/tmp/	Temporary space for use by the system, cleaned upon reboot, so don't use this for saving any work!
/lost+found/	For Every disk partition, contains files that were saved during failures are here.



Linux 4 Embedded Systems

<http://Linux4EmbeddedSystems.com>



Linux For Embedded Systems

For Arabs

Course 102: Understanding Linux

Ahmed ElArabawy

Lecture 3: Basic Concepts and Commands





Files & Filenames

- Filenames can use any characters

`abc??##.a1 sta*.pn`

But it is not wise to use special characters in filenames

spaces are accepted but not recommended, use dashes or underscores instead

`test results.txt` => `test_results.txt` or `test-results.txt`

- File names are case sensitive

`Test_results.doc` `test_results.doc`

But avoid creating files with same names with difference in case

- File names starting with a dot are hidden files

`.bashrc` `.profile`

- No concept of file extension, the dot is just another character

`file.txt.old` `.bashrc` `file.doc.mod` `.profile.old` `results.yesterday` `results`

Still it is a good idea to stick to the popular extensions for clarity

Paths

- Directory names separated by slashes '/'

/usr/src/shared/files/myfile.txt

- Can be absolute or relative

- Absolute: Does not depend on where you are

/home/aelarabawy/abcd

~/abcd

~ means ***/home/aelarabawy*** (home directory of current user)

~salah means ***/home/salah*** (home directory of the user salah)

- Relative: Depends where you are in the tree

./my-project/progress-reports

../..//my-project/progress-reports

. means Current Directory

.. Means Parent Directory



Commands

- General form of Commands,
\$ <command> [options] [arguments]

- Commands can be issued on their own

\$ ls

\$ pwd

\$ cd

- The Options are normally optional (by definition)

The usually start with ‘-’ or ‘--’

‘-’ used with the short name for the option

\$ ls -a

‘--’ used with the long name for the option

\$ ls --all

We normally use the short name in CLI and the long name in scripts (for clarity)



Commands

- If we want to use multiple options Sometimes we join options or keep them separate

\$ ls -a -R

\$ ls -aR

\$ ls --all --recursive

- Sometimes a '--' without a name afterwards will mean end of options (even a following hyphen will not be considered a new option)
- Arguments are those info passed to the command such as file names and paths

\$ rm -rf ./project-data



Basic Commands

Command	Effect
\$ ls	To list files
\$ tree	To list a tree of files
\$ pwd	To show the current directory
\$ cd	To move around in the tree
\$ mkdir	To Create directories
\$ cp	To copy files and directories
\$ mv	To move or rename files and directories
\$ rm	To delete files and directories
\$ clear	To clear the screen

Listing Files (ls Command)



\$ ls (List Current Directory)

```
mandar@YourOwnLinux: ~
mandar@YourOwnLinux:~$ ls
a.c                               EvenOdd.sh~
Add.sh~                            examples.desktop
a.out                             Fact.sh~
ascii                            helloWorld.java
backup_helloWorld.java           helloWorld.sh~
cat_demo.doc                      helloWorld.sh.java
Desktop                           Host.sh
Div.sh~                           ManU.txt
Documents                         ManU.txt~
Downloads                         Mul.sh~
dropbox.deb                        Music
Pictures                          Prime.sh~
Public                            script
Sub.sh~                           Table.sh~
Templates                         utorrent-server-3.0-25053.tar.gz
Videos
```

Listing Files (ls Command)



\$ ls -a (List all files, including hidden files)

```
mandar@YourOwnLinux:~$ ls -a
.                                .gt5-diffs
..                               .gt5.html
.a.c                            .gtk-bookmarks
Add.sh~                         .gvfs
a.out                           helloWorld.java
ascii                           helloWorld.sh~
backup_helloWorld.java          helloWorld.sh.java
.bash_history                    helloWorld.sh.java~
.bash_logout                     Host.sh
.bashrc                          .ICEauthority
.bluefish                        .local
.cache                           ManU.txt
cat_demo.doc                     ManU.txt~
.config                          .mission-control
 dbus                            .mozilla
Desktop                          Mul.sh~
Div.sh~                         Music
.dmrc                            Pictures
_dnacache                       Prime.sh~
Documents                        .profile
Downloads                        Public
dropbox.deb                      .pulse
EvenOdd.sh~                     .pulse-cookie
examples.desktop                 script
Fact.sh~                        .shotwell
.fontconfig                      Sub.sh~
```

Listing Files (ls Command)



\$ ls -l (List with long format)

A screenshot of a terminal window titled "robb@ubuntu: ~". The window displays the output of the "ls -l" command, listing the contents of the user's home directory. The output shows ten entries: Desktop, Documents, Downloads, examples.desktop, hw02, Music, Pictures, Public, Templates, and Videos. Each entry includes file permissions (e.g., drwxr-xr-x), file size (e.g., 4096), modification date (e.g., 2011-10-04 22:25), and name (e.g., Desktop).

```
robb@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 robb robb 4096 2011-10-04 22:25 Desktop
drwxr-xr-x 3 robb robb 4096 2011-05-14 15:57 Documents
drwxr-xr-x 3 robb robb 4096 2011-10-03 19:34 Downloads
-rw-r--r-- 1 robb robb 179 2011-05-12 09:11 examples.desktop
drwxr-xr-x 4 robb robb 4096 2011-09-29 23:37 hw02
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Music
drwxr-xr-x 3 robb robb 4096 2011-10-04 22:23 Pictures
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Public
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Templates
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Videos
robb@ubuntu:~$
```

Listing Files (ls Command)



\$ ls -l (List with long format)

The terminal window shows the following output:

```
robb@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 robb robb 4096 2011-10-04 22:25 Desktop
drwxr-xr-x 3 robb robb 4096 2011-05-14 15:57 Documents
drwxr-xr-x 3 robb robb 4096 2011-10-03 19:34 Downloads
-rw-r--r-- 1 robb robb 179 2011-05-12 09:11 examples.desktop
drwxr-xr-x 4 robb robb 4096 2011-09-29 23:37 hw02
```

Below the terminal, a diagram illustrates the fields of a long listing entry:

File Type	# of Hard Links	File size	Last Modify Time	File name
-rwxr-x---	1	0	Oct 31 11:06	test
User	Group	User	Group	
Permissions		Owners		

Annotations explain the fields:

- Permissions:** Points to the first part of the file type column, showing "-rwxr-x---".
- # of Hard Links:** Points to the "# of Hard Links" column, showing "1".
- File size:** Points to the "File size" column, showing "0".
- Last Modify Time:** Points to the "Last Modify Time" column, showing "Oct 31 11:06".
- File name:** Points to the "File name" column, showing "test".
- User:** Points to the "User" row under the permissions column.
- Group:** Points to the "Group" row under the permissions column.
- Owners:** Points to the "User" and "Group" rows under the "# of Hard Links" column.

Legend for file types:

- (file)
- d (directory)
- l (symbolic link)
- b (block special file)
- c (character special file)
- p (named pipe special file)
- s (local socket special file)

Legend for permissions:

- r read
- w write
- x execute

Listing Files (ls Command)



\$ ls [options] [<dir or file> ..]

\$ ls (list current directory)

\$ ls -a (all: show hidden files)

\$ ls -l (long: show file details)

\$ ls -t (timestamp: sort based on timestamp)

\$ ls -S (Size: sort based on file size)

\$ ls -r (reverse: make the sort in reverse order)

\$ ls -d (directories: Only show directories)

\$ ls -R (Recursive: list files inside subdirectories)

\$ ls <dir> (List the contents of the mentioned directory)

\$ ls <file> (List the mentioned)

\$ ls <dir or file> <dir or file> <dir or file> (list selected dirs or files)

Or a Mix of These Options

Displaying the Directory Tree (tree Command)



\$ tree (List tree from Current Directory)

```
bashitout@mike-desktop:~/projects$ tree
.
└── awesomesite
    ├── awesomesite
    │   ├── __init__.py
    │   ├── settings.py
    │   ├── urls.py
    │   └── wsgi.py
    └── manage.py
```

```
2 directories, 5 files
```

```
bashitout@mike-desktop:~/projects$ █
```

Displaying the Directory Tree (tree Command)



\$ tree -a (List tree from Current Directory)

```
bashitout@mike-desktop:~/projects$ tree -a
:
└── awesomesite
    ├── awesomesite
    │   ├── __init__.py
    │   ├── settings.py
    │   ├── urls.py
    │   └── wsgi.py
    └── .git
        ├── branches
        ├── config
        ├── description
        ├── HEAD
        └── hooks
            ├── applypatch-msg.sample
            ├── commit-msg.sample
            ├── post-update.sample
            ├── pre-applypatch.sample
            ├── pre-commit.sample
            ├── prepare-commit-msg.sample
            ├── pre-rebase.sample
            └── update.sample
        ├── info
        │   └── exclude
        ├── objects
        │   ├── info
        │   └── pack
        ├── refs
        │   └── heads
        └── tags
    └── manage.py

12 directories, 17 files
bashitout@mike-desktop:~/projects$
```

Displaying the Directory Tree (tree Command)



\$ tree [options] [<dir or file> ..]

\$ tree (display the full tree starting from current dir)

\$ tree -d (only show directories)

\$ tree -a (show all files; including hidden ones)

\$ tree <dir> (show the tree starting from a different point)

Or a Mix of These Options

Print Working Directory (pwd Command)



\$ **pwd**

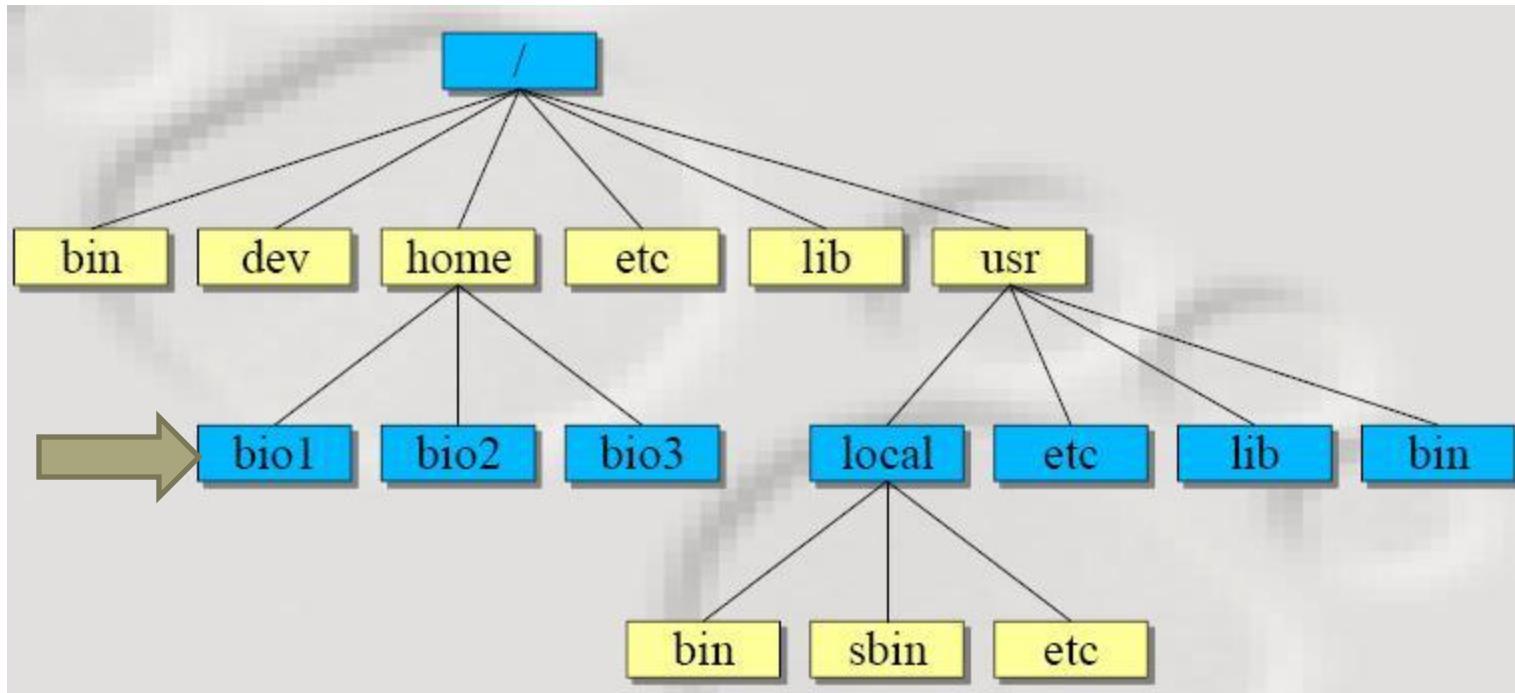
\$ pwd (Display Current Directory)

```
vivek@wks01:~/scripts$ pwd
/home/vivek/scripts
vivek@wks01:~/scripts$ ls -l
total 32
-rwxr-xr-x 1 vivek vivek    545 Sep 28  2008 conditions.sh
-rw-r--r-- 1 vivek vivek  1720 Sep 28  2008 gmail_parser.py
-rwxr-xr-x 1 vivek vivek     56 Dec  1  2007 hddmonit.sh
drwxr-xr-x 2 vivek vivek  4096 Mar 10  01:24 php
-rwxr-xr-x 1 vivek vivek 10283 Jul 20  2012 pogodynka.sh
drwxr-xr-x 2 vivek vivek  4096 Mar 10  01:24 python
vivek@wks01:~/scripts$ cd python
vivek@wks01:~/scripts/python$ pwd
/home/vivek/scripts/python
vivek@wks01:~/scripts/python$ █
```

Moving Around (cd Command)



\$ cd [destination]



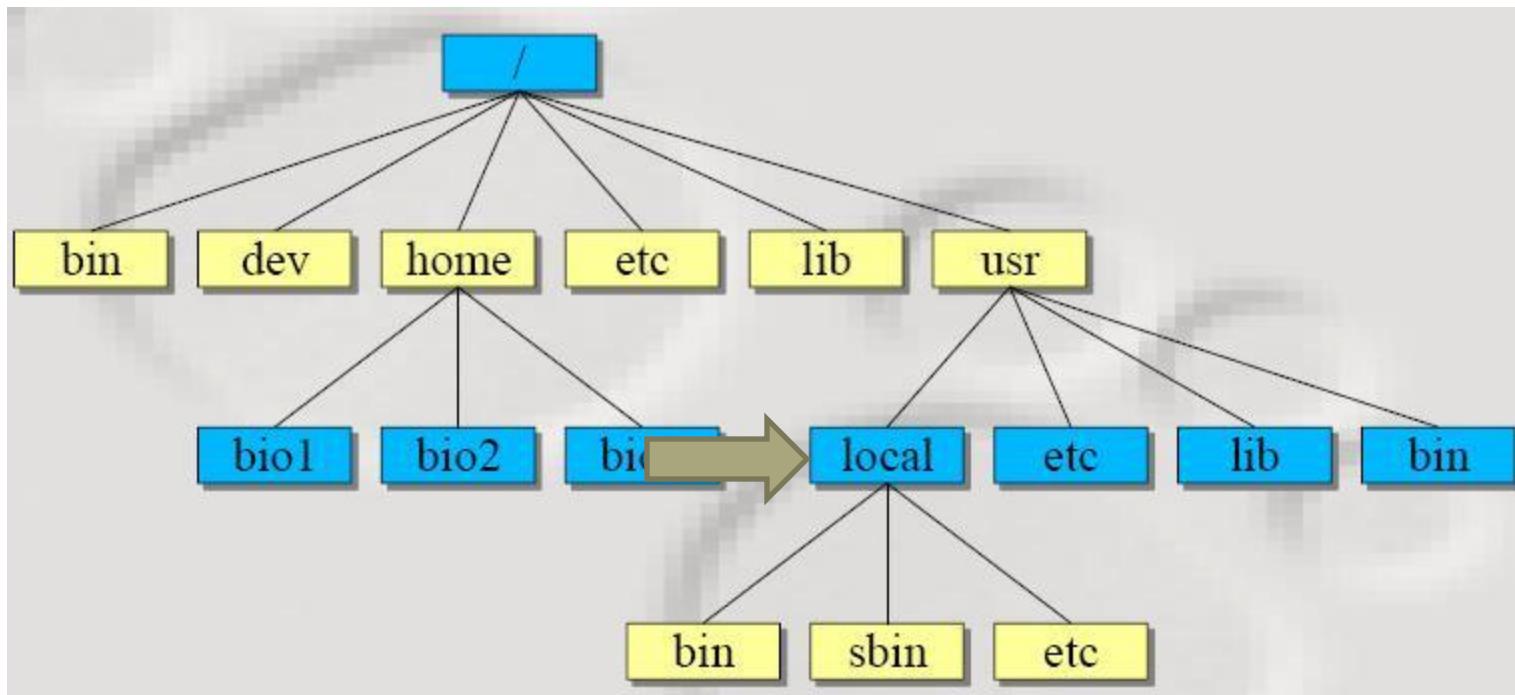
\$ pwd

/home/bio1/

Moving Around (cd Command)



\$ cd [destination]

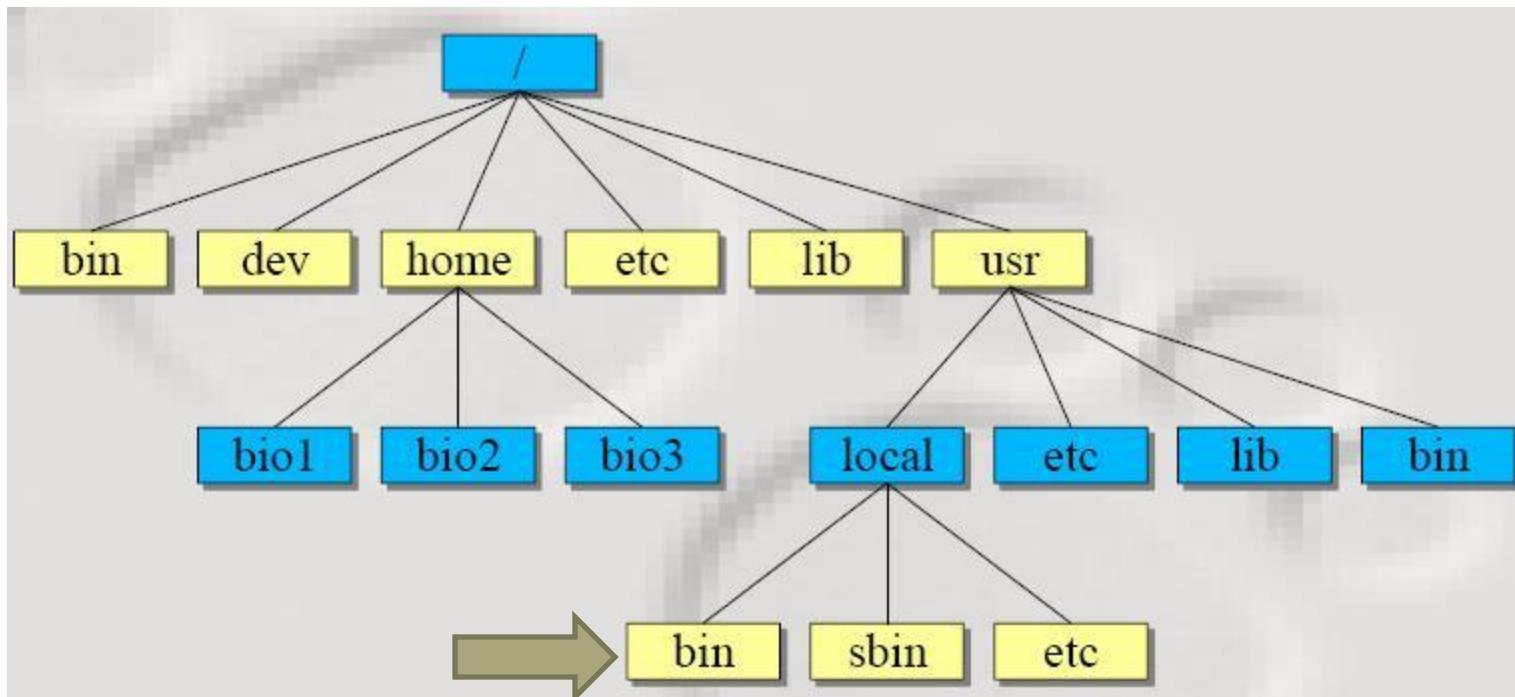


\$ cd /usr/local

Moving Around (cd Command)



\$ cd [destination]



\$ cd /usr/local/bin

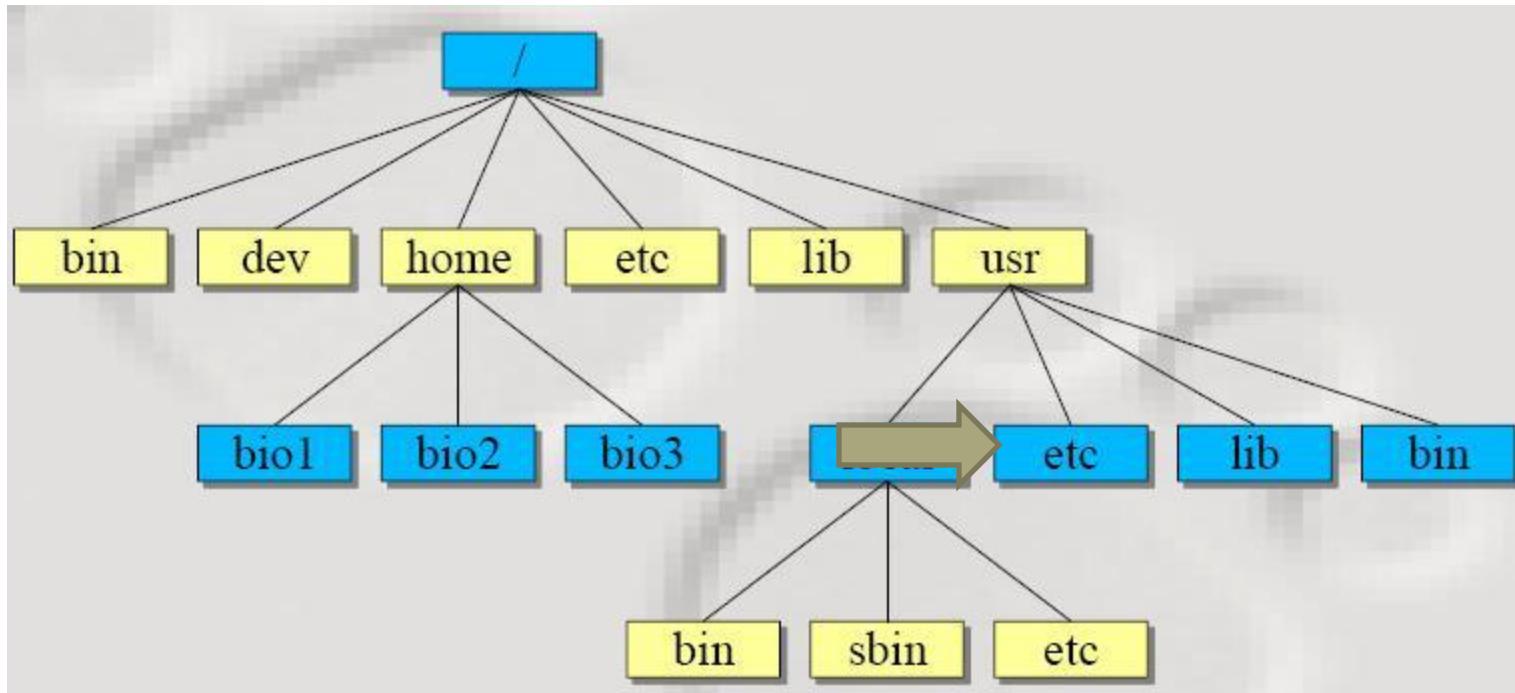
\$ cd ./bin

\$ cd bin

Moving Around (cd Command)



\$ cd [destination]



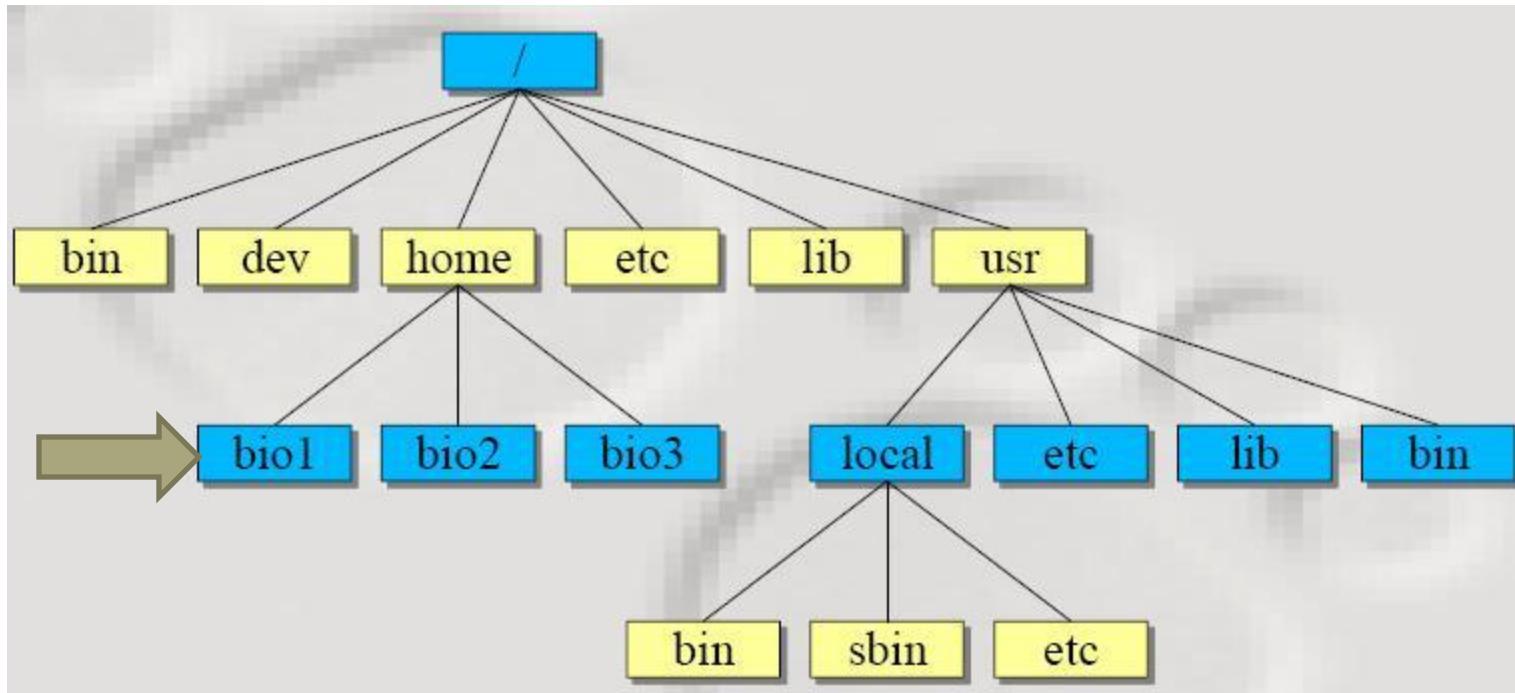
\$ cd /usr/etc

\$ cd ../../etc

Moving Around (cd Command)



\$ cd [destination]



\$ cd ~

\$ cd

Moving Around (cd Command)



\$ cd [destination]

\$ cd /etc/network (absolute path)

\$ cd ../project/ (relative path)

\$ cd ./project (relative path)

\$ cd project (relative path, same as **./project**)

\$ cd ~ (go to my home directory **/home/aelarabawy/**)

\$ cd ~user_name (go to **/home/user_name**)

\$ cd (same as **cd ~**)

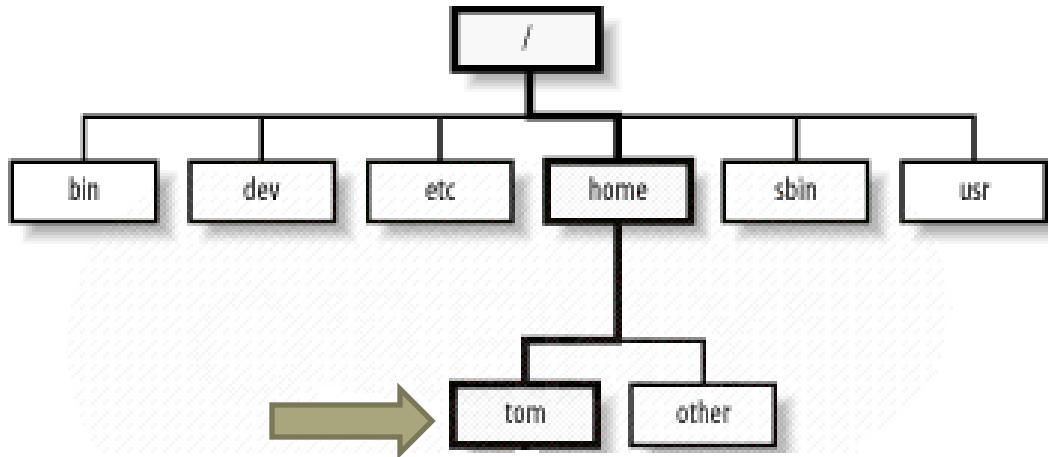
\$ cd .. (go to parent directory)

\$ cd - (go to previous directory)

Making New Directories (mkdir Command)



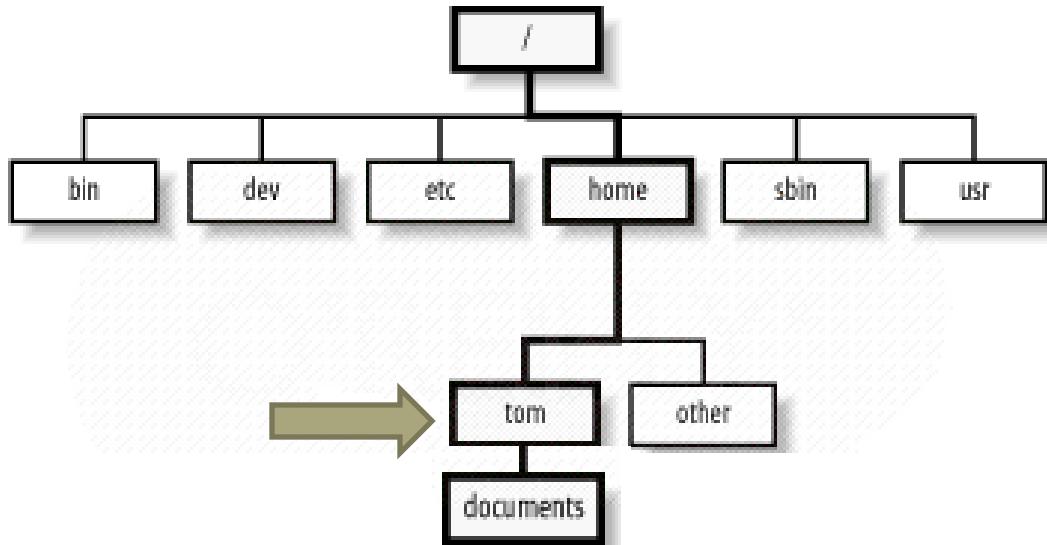
\$ mkdir <new directory name with path>



Making New Directories (mkdir Command)



\$ mkdir <new directory name with path>



\$ mkdir documents

\$ mkdir ./documents

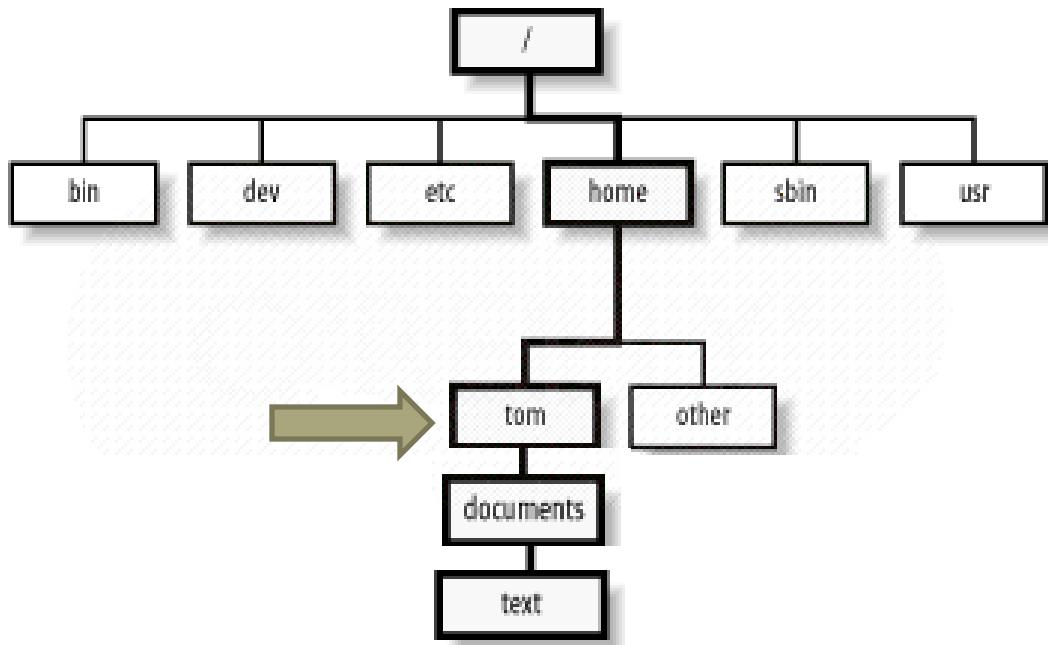
\$ mkdir /home/tom/documents

\$ mkdir ~/documents

Making New Directories (mkdir Command)



\$ mkdir <new directory name with path>

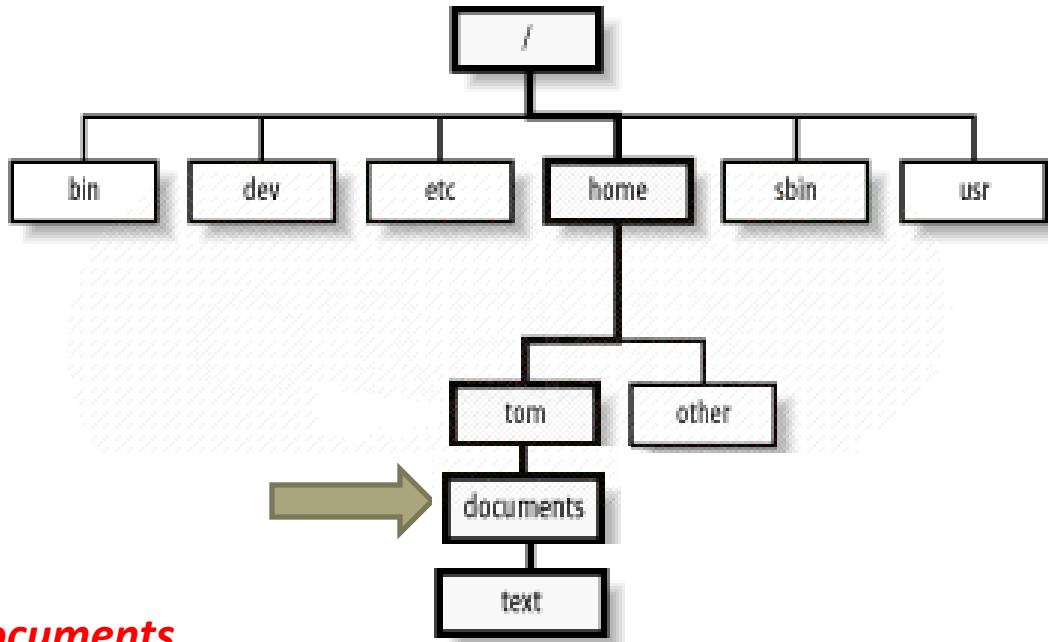


\$ *mkdir documents/text*

Making New Directories (mkdir Command)



\$ mkdir <new directory name with path>

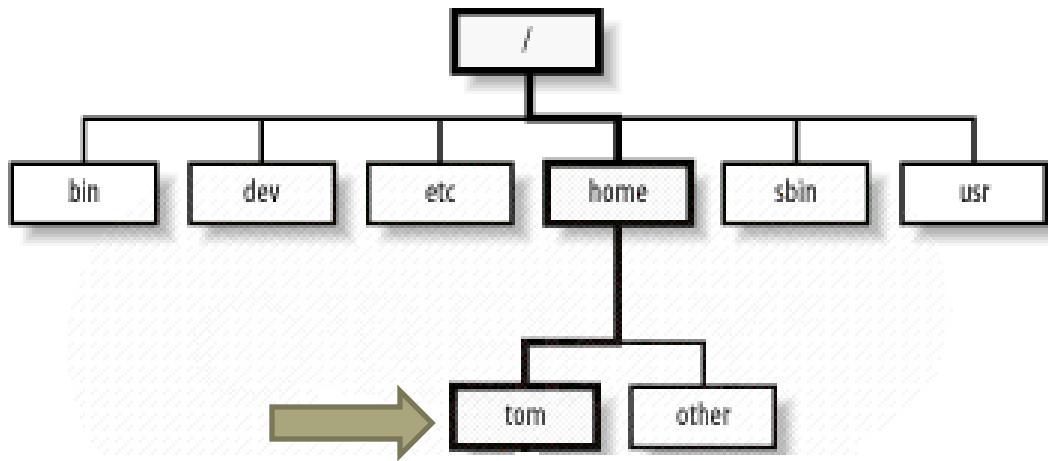


\$ cd documents
\$ mkdir text

Making New Directories (mkdir Command)



\$ mkdir <new directory name with path>



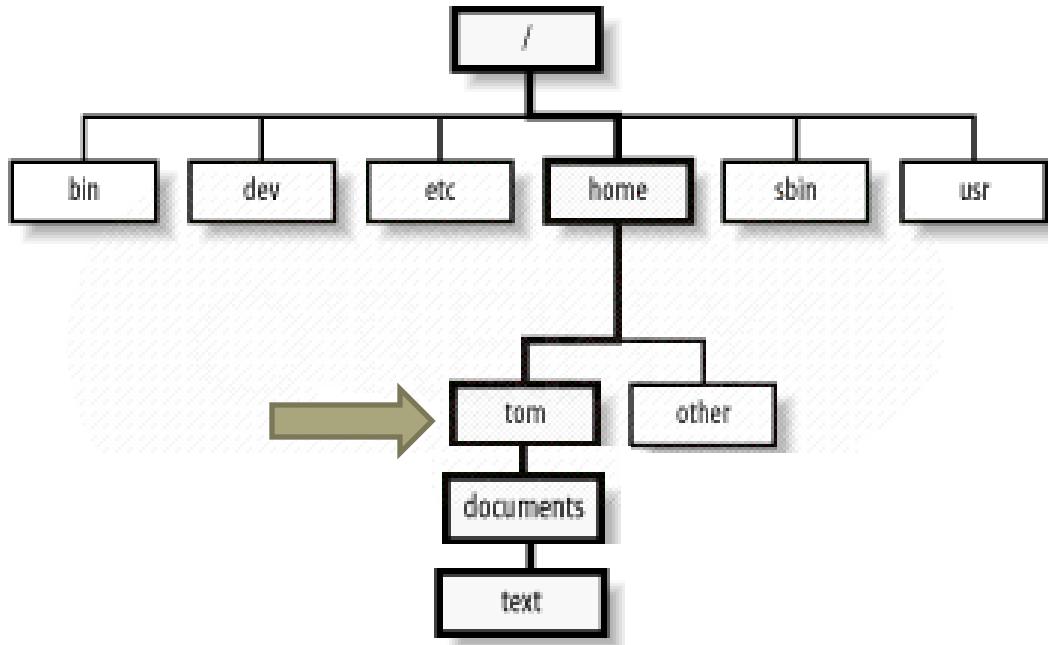
\$ mkdir documents/text



Making New Directories (mkdir Command)



\$ mkdir <new directory name with path>



\$ *mkdir -p documents/text*

Making New Directories (mkdir Command)



\$ mkdir <new directory name with path>

\$ mkdir project1 (create a new directory from current location)

\$ mkdir project1 project2 (create 2 directories)

\$ mkdir /home/aelarabawy/lectures (absolute path)

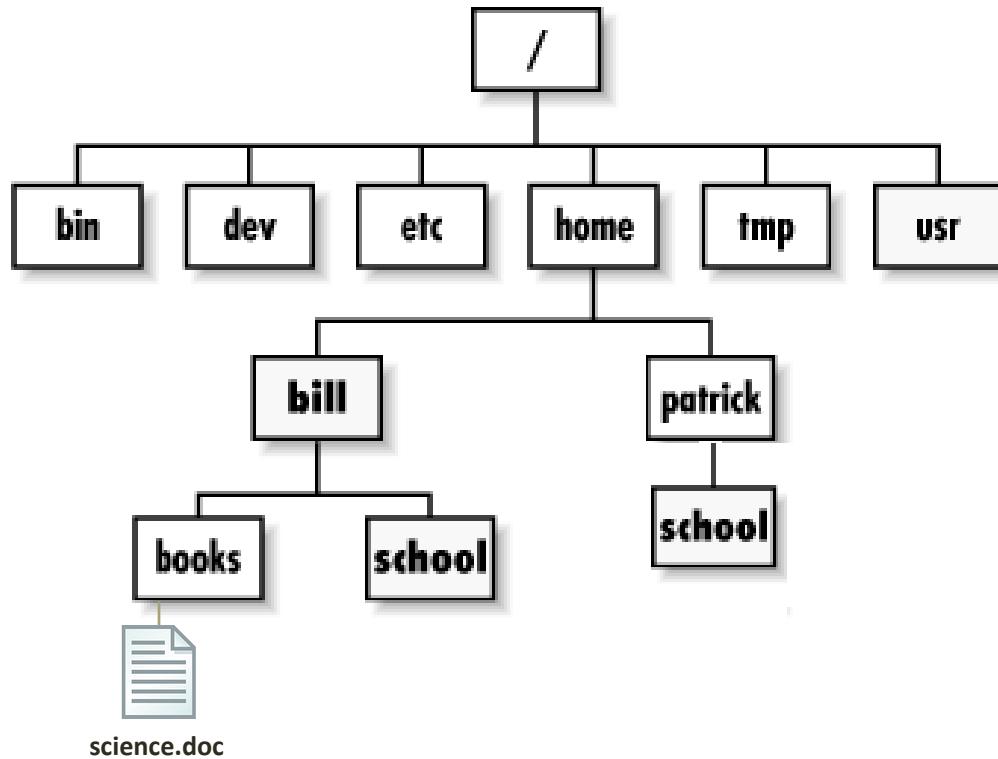
\$ mkdir ../projects/project1 (relative path)

\$ mkdir -p ../projects/project1 (create intermediate folders if needed)

Copying Files & Directories (cp Commands)



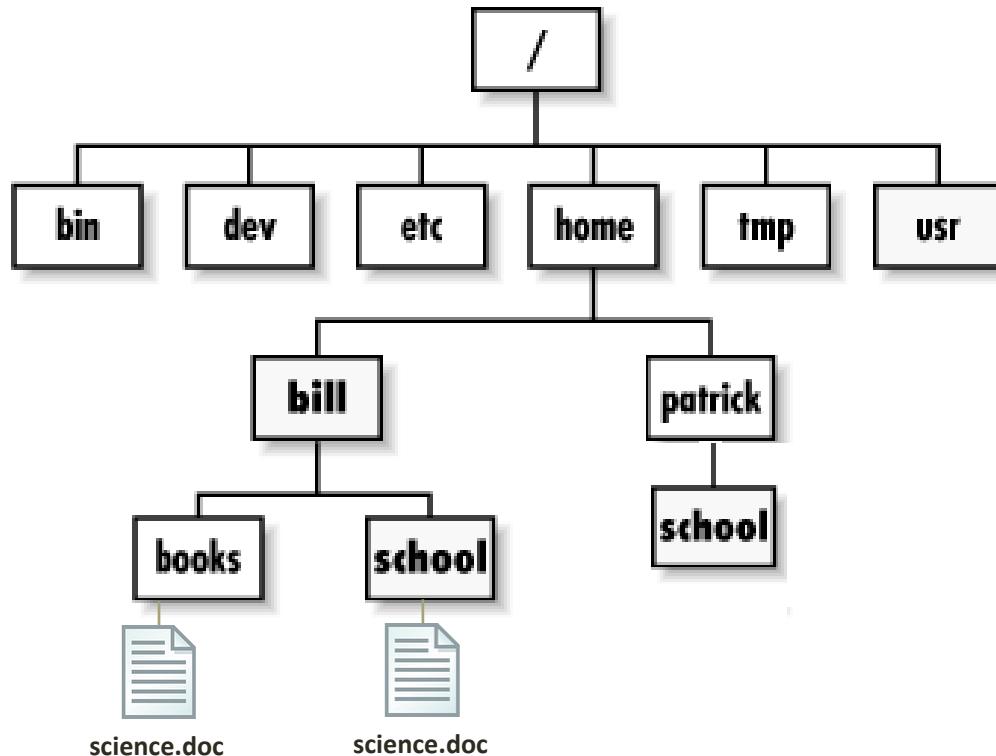
\$ cp <existing file or dir> <new filename & destination>



Copying Files & Directories (cp Commands)



\$ cp <existing file or dir> <new filename & destination>

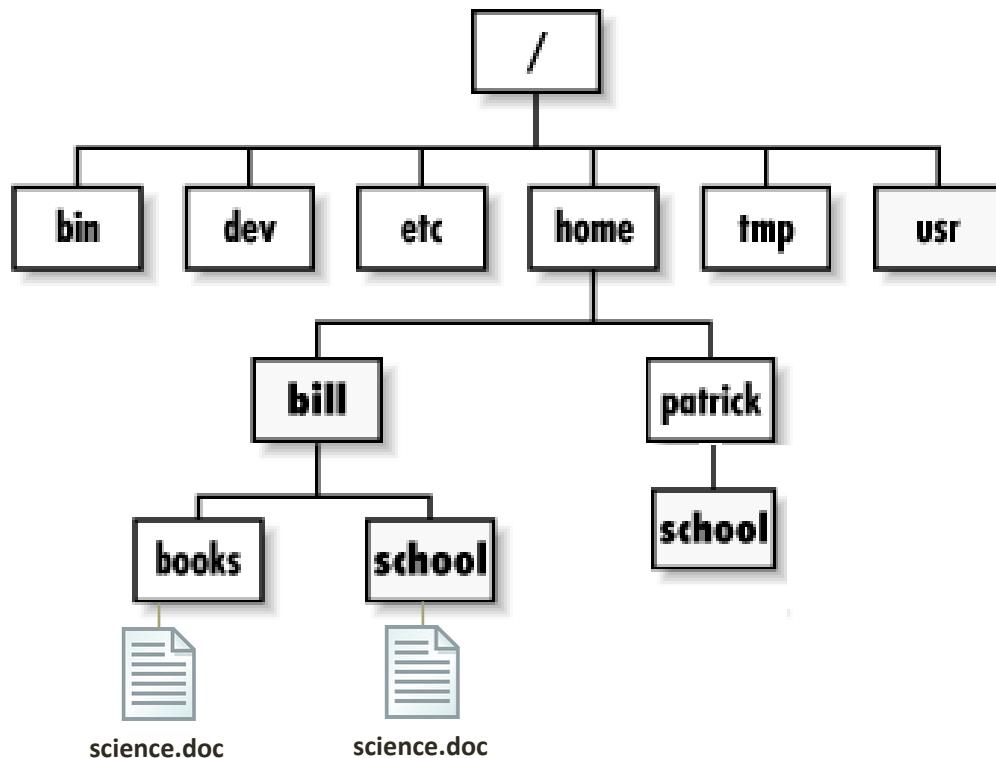


```
$ cp ~/books/science.doc ~/school/science.doc  
$ cp ~/books/science.doc ~/school/
```

Copying Files & Directories (cp Commands)



\$ cp <existing file or dir> <new filename & destination>



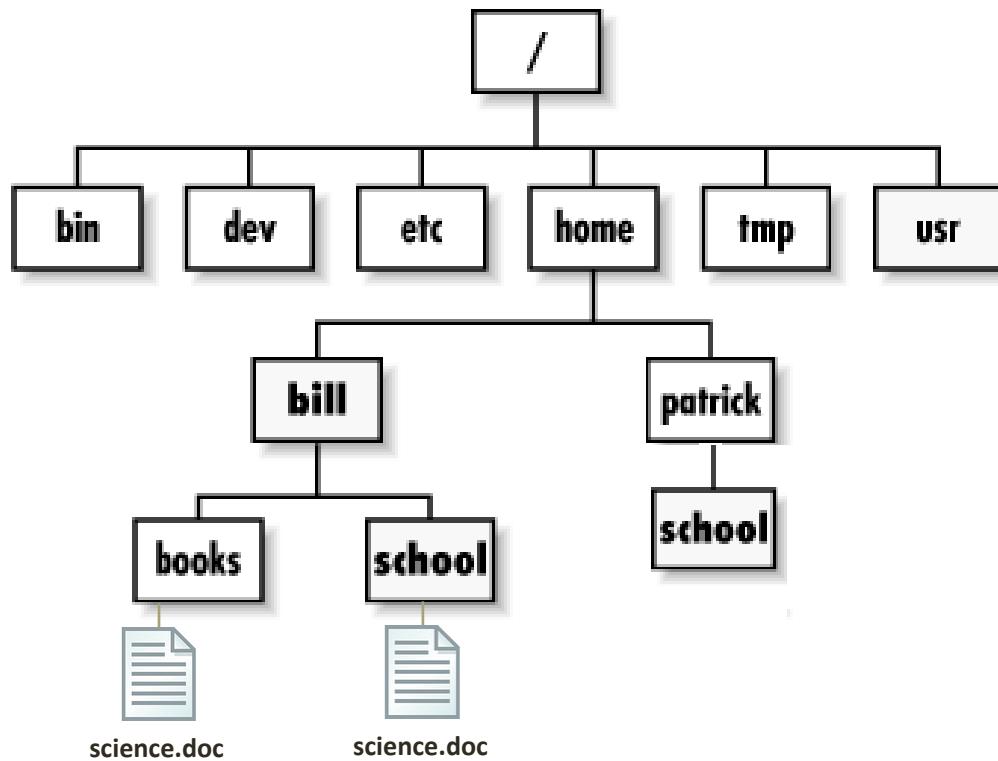
\$ cp /home/bill/school /home/patrick/



Copying Files & Directories (cp Commands)



\$ cp <existing file or dir> <new filename & destination>

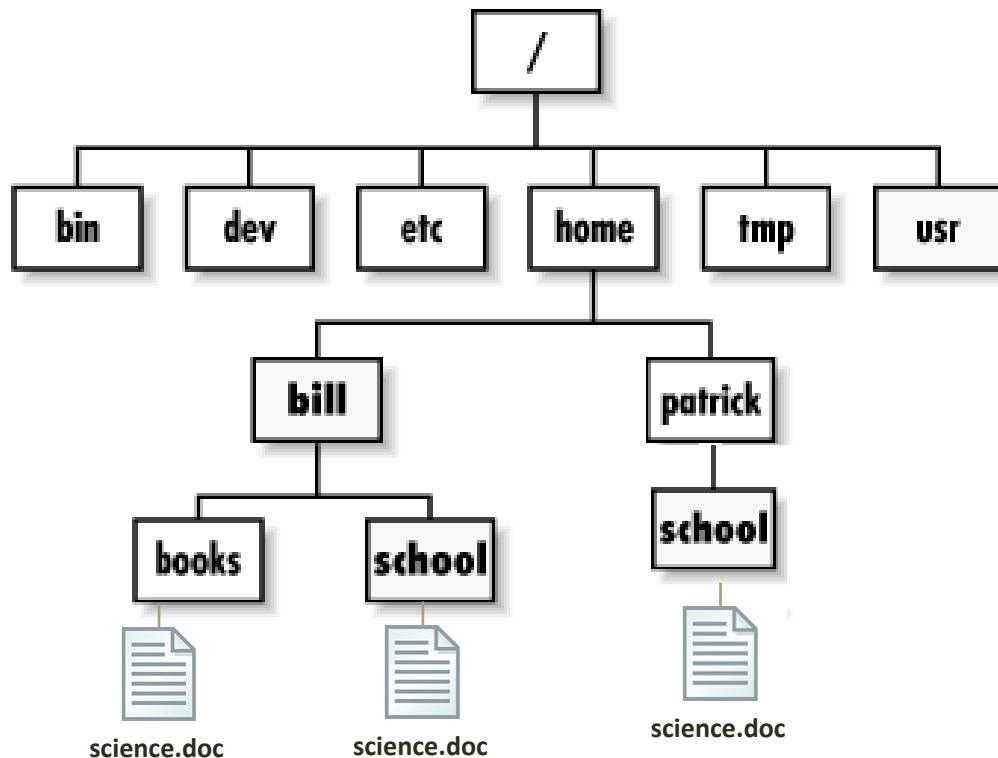


\$ sudo cp /home/bill/school /home/patrick/ **X**

Copying Files & Directories (cp Commands)



\$ cp <existing file or dir> <new filename & destination>



\$ sudo cp -r /home/bill/school /home/patrick

Copying Files & Directories (cp Commands)



\$ cp <existing file or dir> <new destination>

\$ cp file1 file2 (copy file1 to a new file file2 same location)

\$ cp file1 .../projects/ (copy file1 to the new location)

\$ cp -r folder1 .../projects/ (copy the folder with its contents)

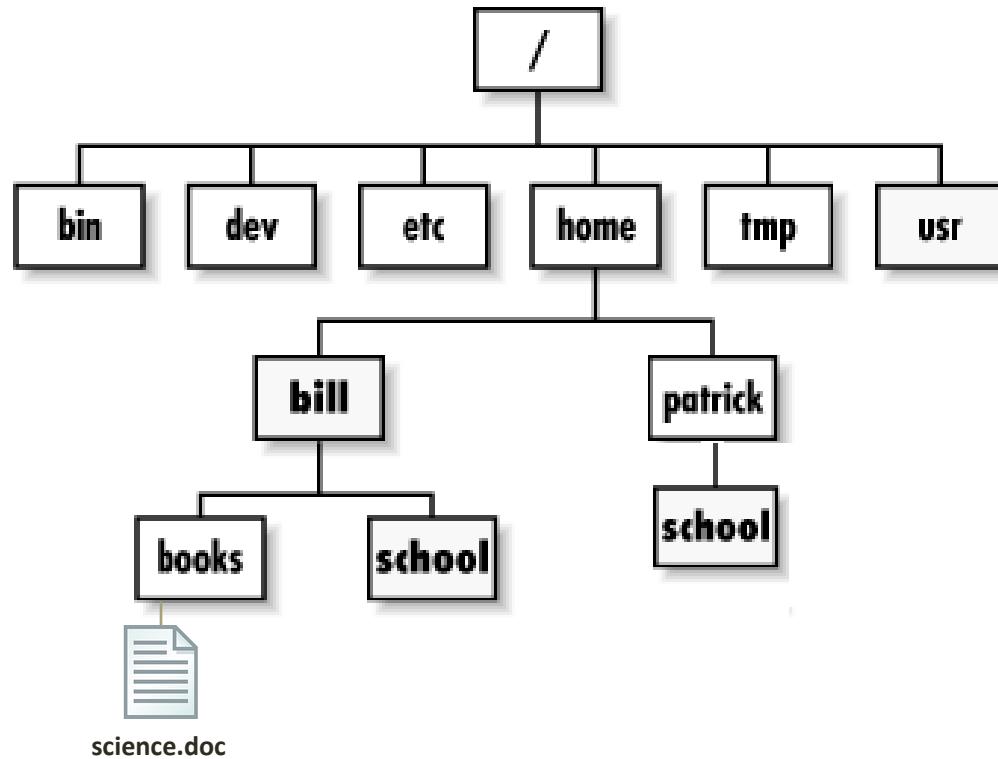
\$ cp -r folder1 .../projects/folder2 (copy the folder with new name)

\$ cp /etc/passwd . (copy the file ...to here)

Moving/Renaming Files & Directories (mv Command)



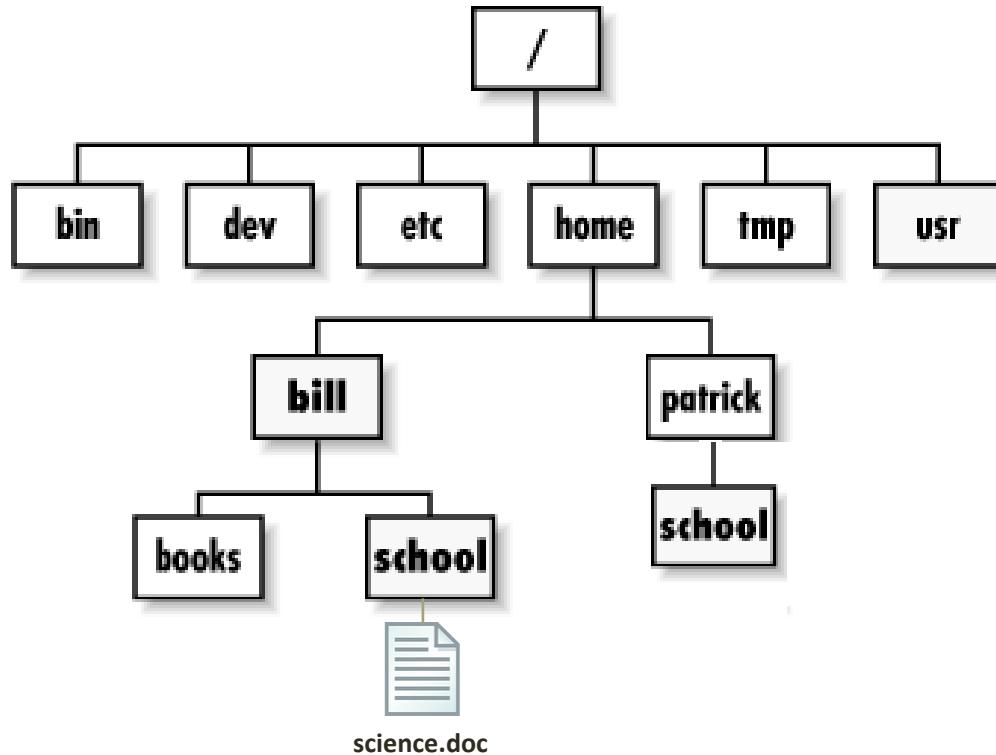
```
$ mv <existing file or dir> <new filename & destination>
```



Moving/Renaming Files & Directories (mv Command)



\$ mv <existing file or dir> <new filename & destination>

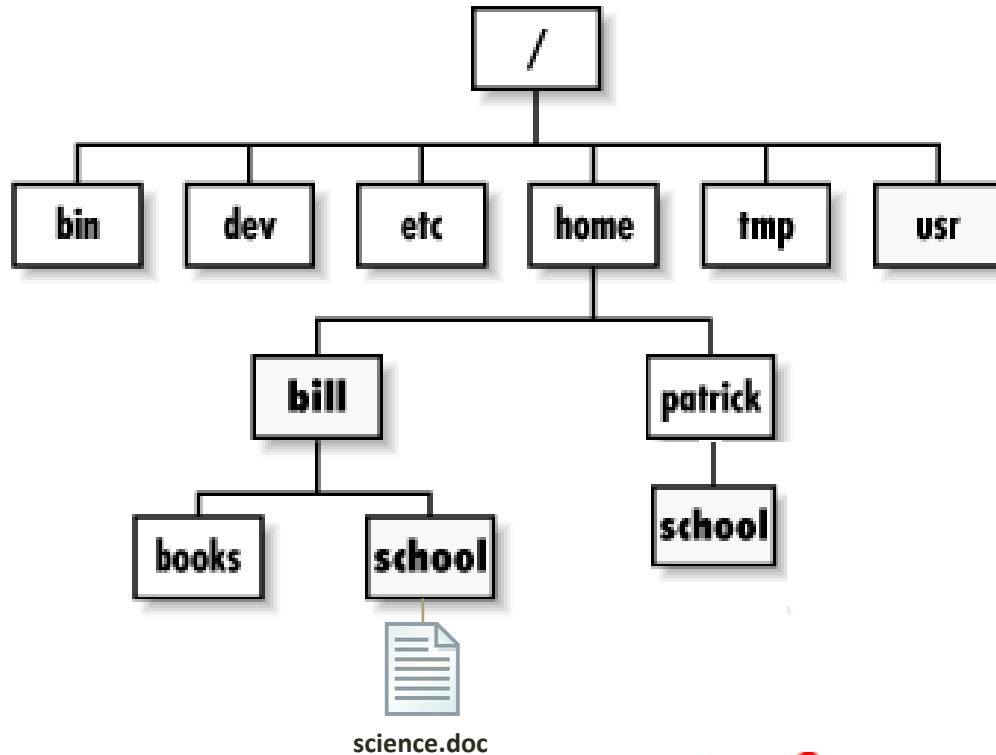


\$ mv ~/books/science.doc ~/school/science.doc
\$ mv ~/books/science.doc ~/school/

Moving/Renaming Files & Directories (mv Command)



\$ mv <existing file or dir> <new filename & destination>



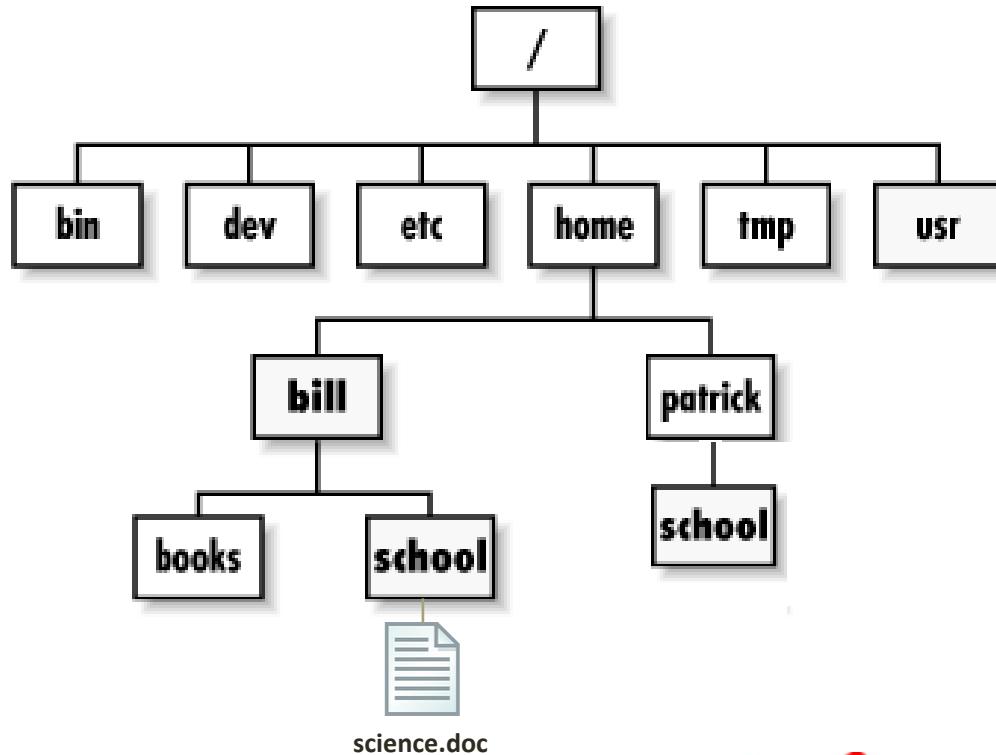
\$ mv /home/bill/school /home/patrick/



Moving/Renaming Files & Directories (mv Command)



\$ mv <existing file or dir> <new filename & destination>



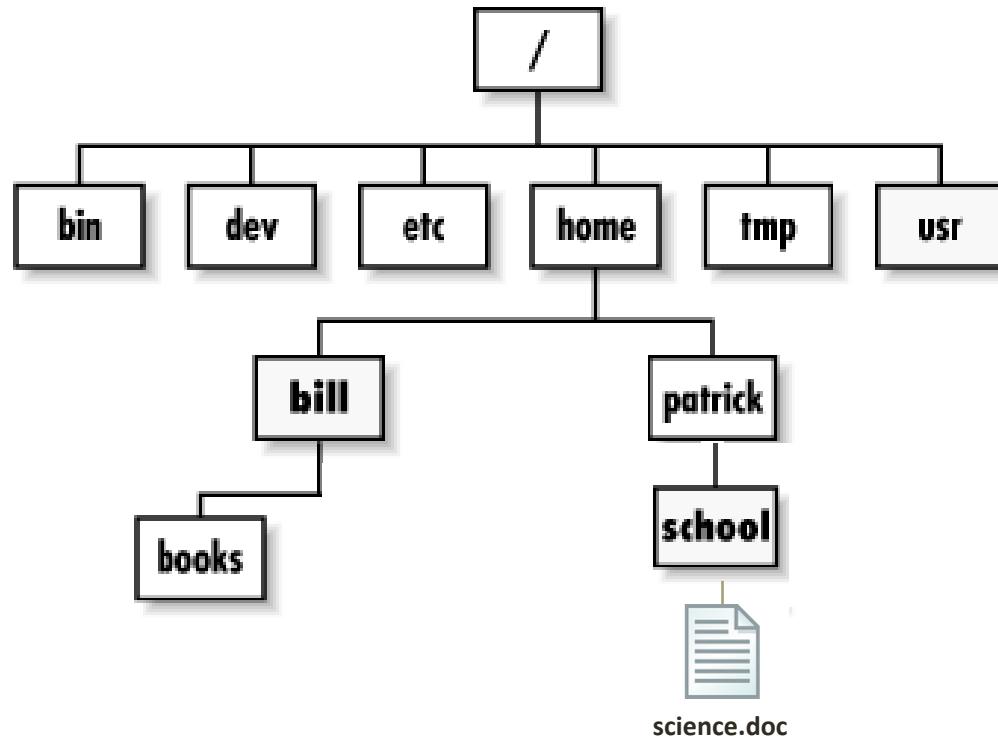
\$ sudo mv /home/bill/shool /home/patrick/



Moving/Renaming Files & Directories (mv Command)



\$ mv <existing file or dir> <new filename & destination>



\$ sudo mv -r /home/bill/books /home/patrick

Moving/Renaming Files & Directories (mv Command)



\$ mv <existing file or dir> <new destination>

\$ mv file1 file2 (rename file1 to file2)

\$ mv file1 .../projects/ (move file1 to the new location)

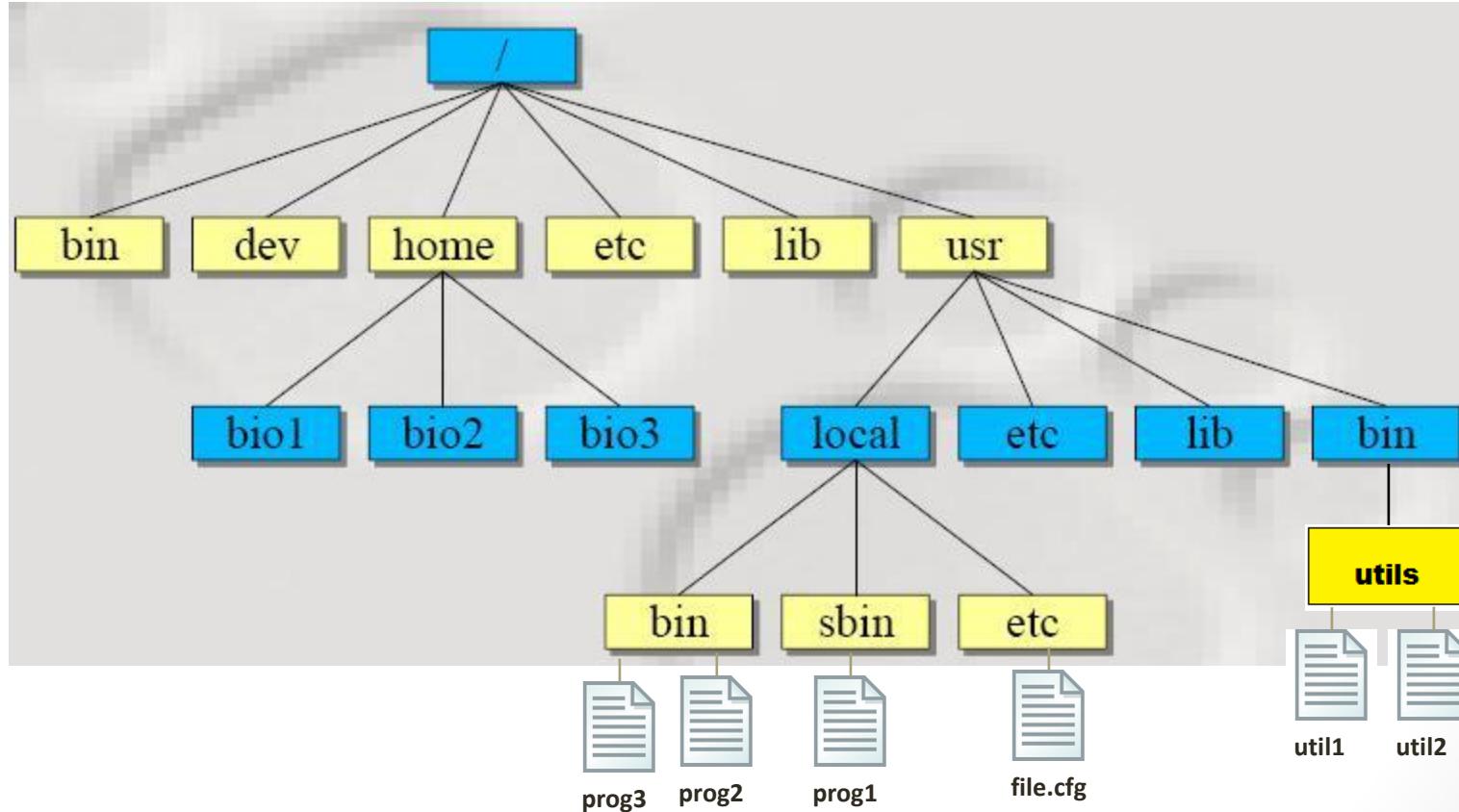
\$ mv -r folder1 .../projects/ (move the folder with its contents)

\$ mv -r folder1 .../projects/folder2 (move with new name)

Removing files and Directories (rm Command)



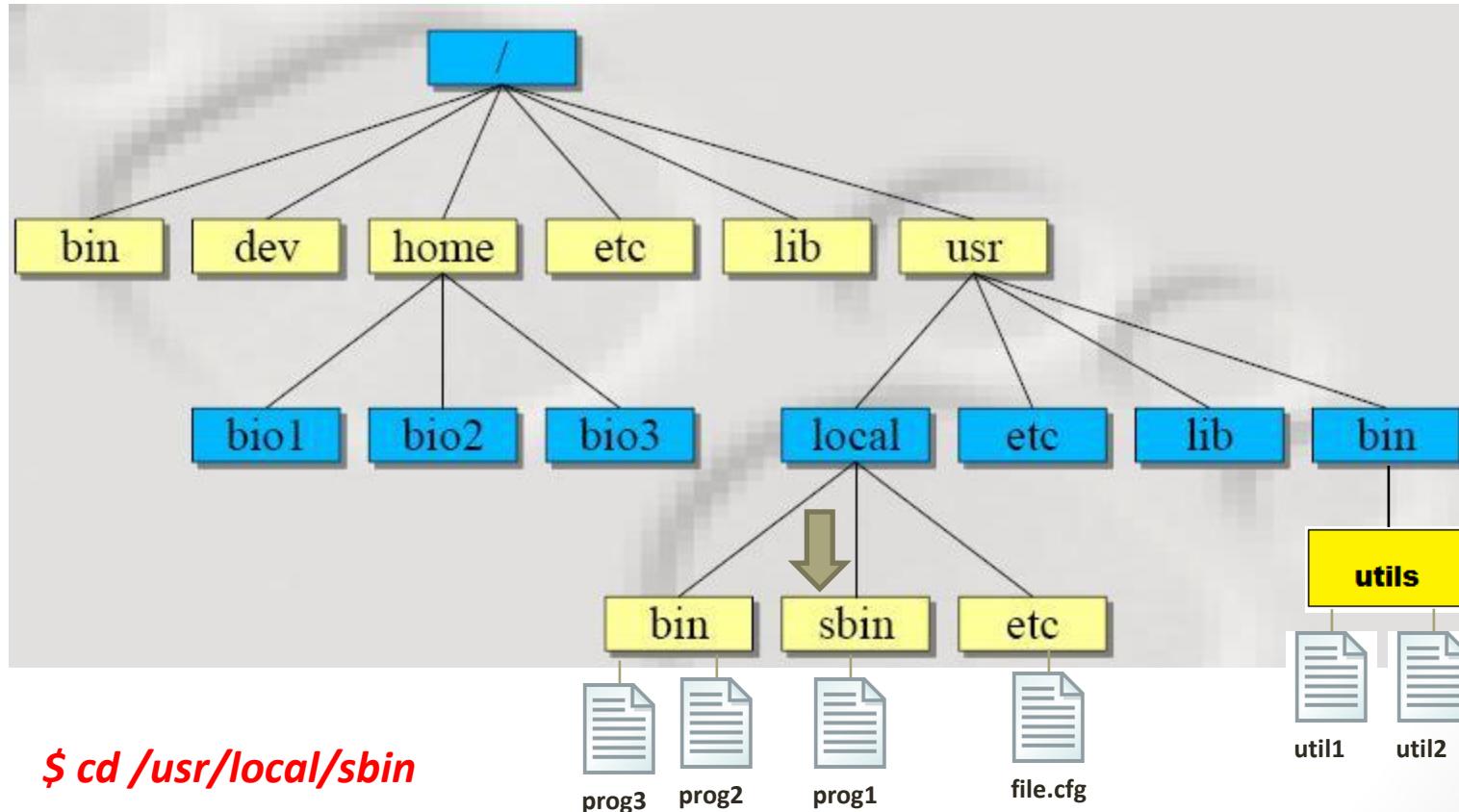
\$ rm [options] <file or dir list>



Removing files and Directories (rm Command)



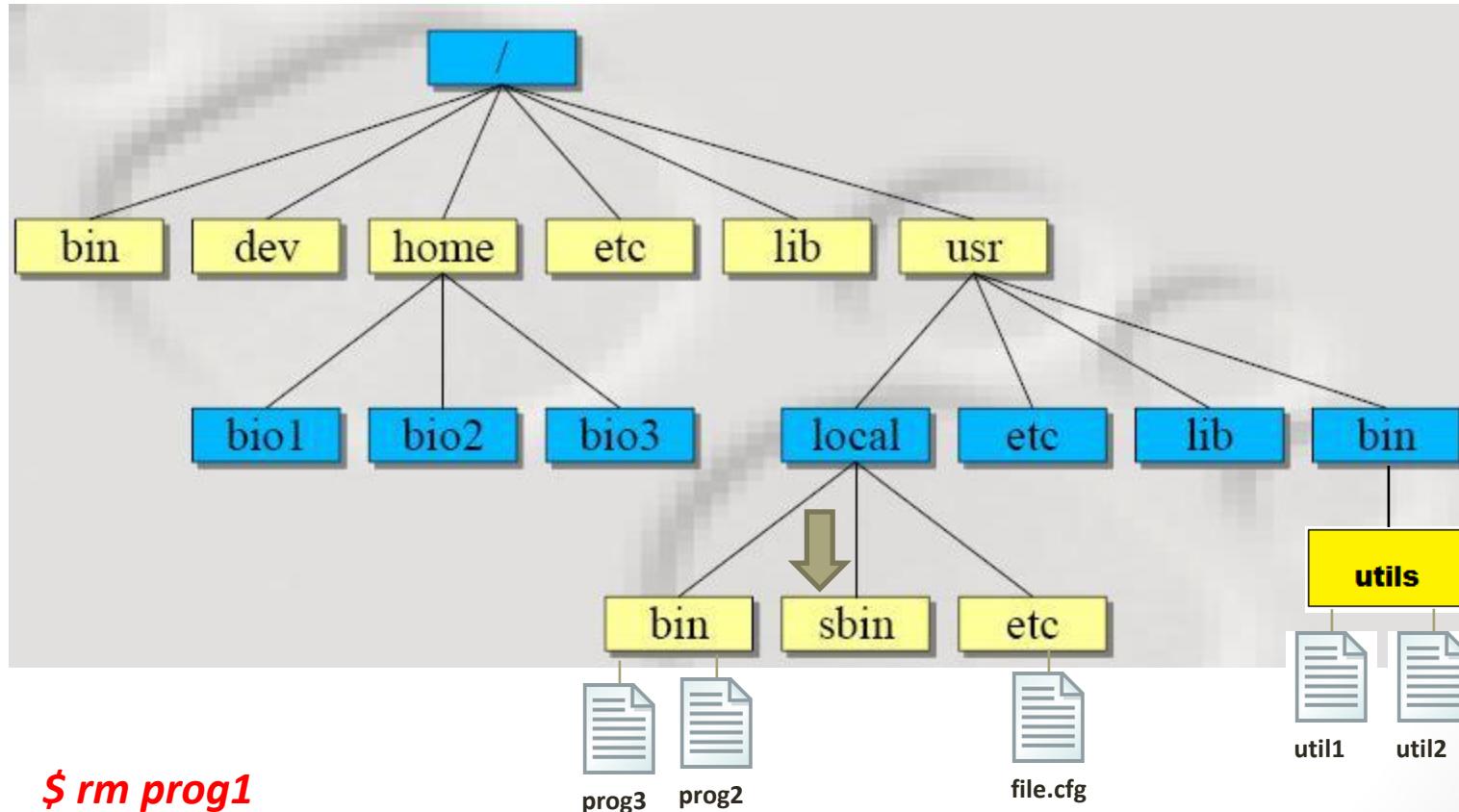
\$ rm [options] <file or dir list>



Removing files and Directories (rm Command)



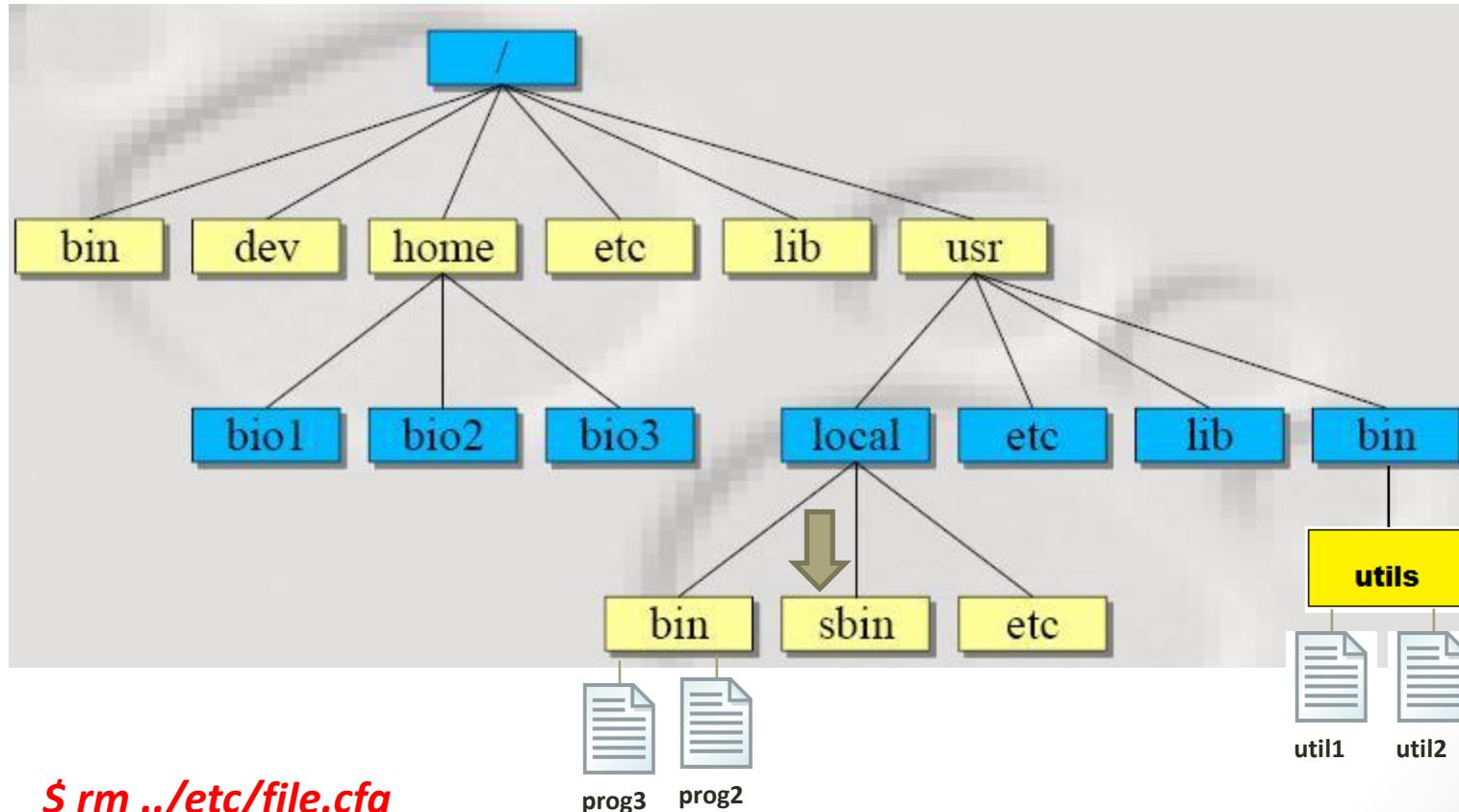
\$ rm [options] <file or dir list>



Removing files and Directories (rm Command)



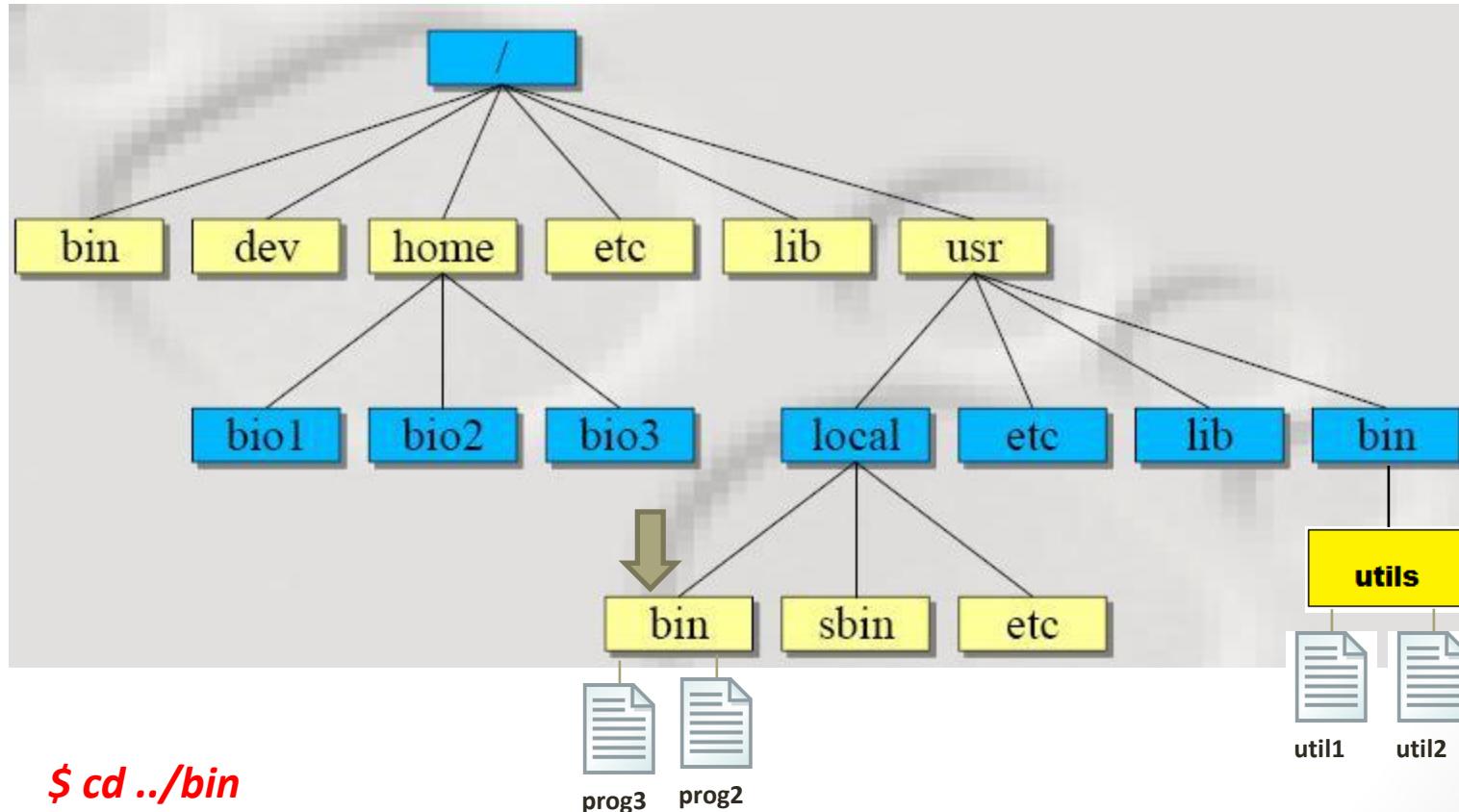
\$ rm [options] <file or dir list>



Removing files and Directories (rm Command)



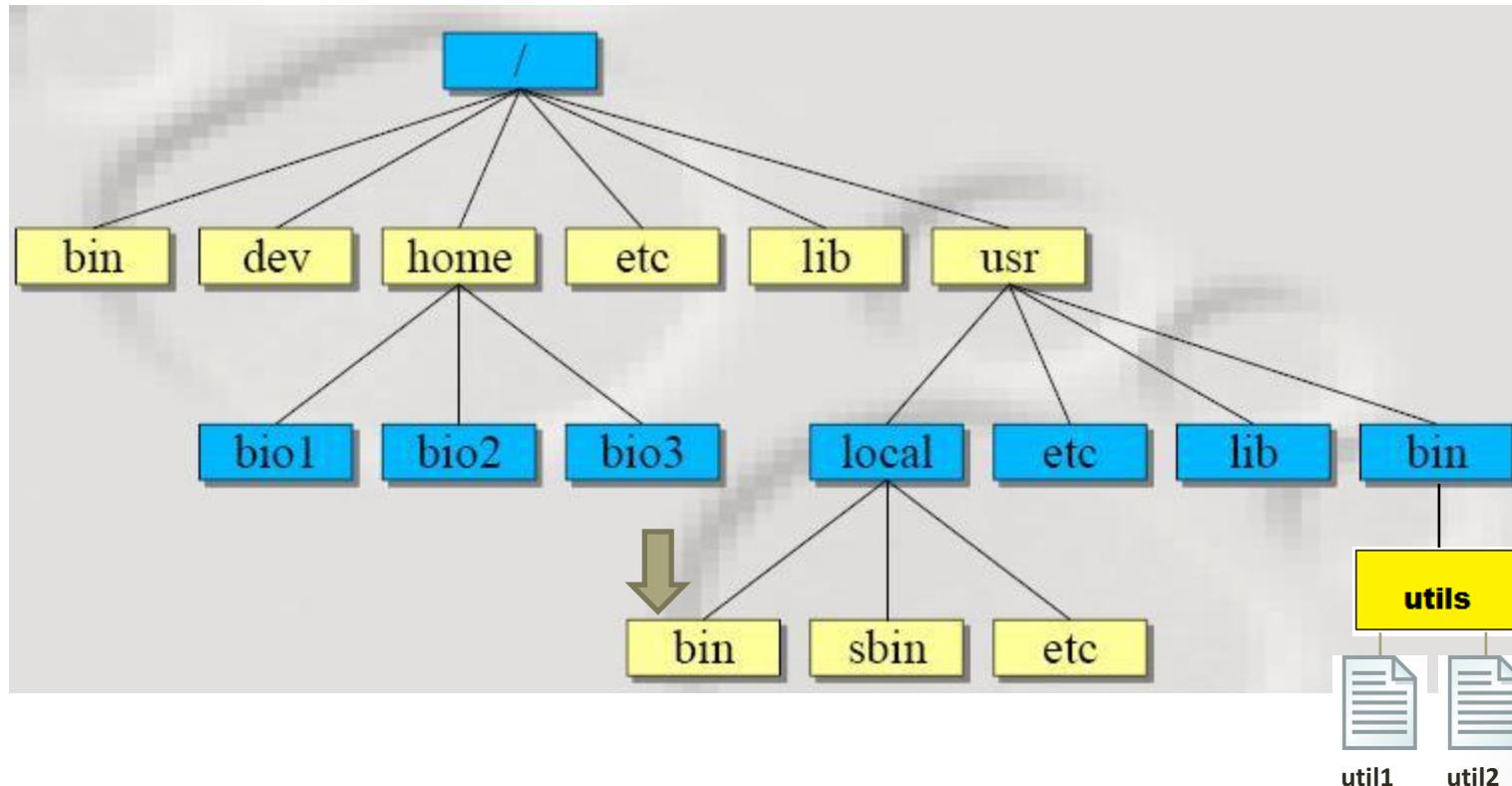
\$ rm [options] <file or dir list>



Removing files and Directories (rm Command)



\$ rm [options] <file or dir list>

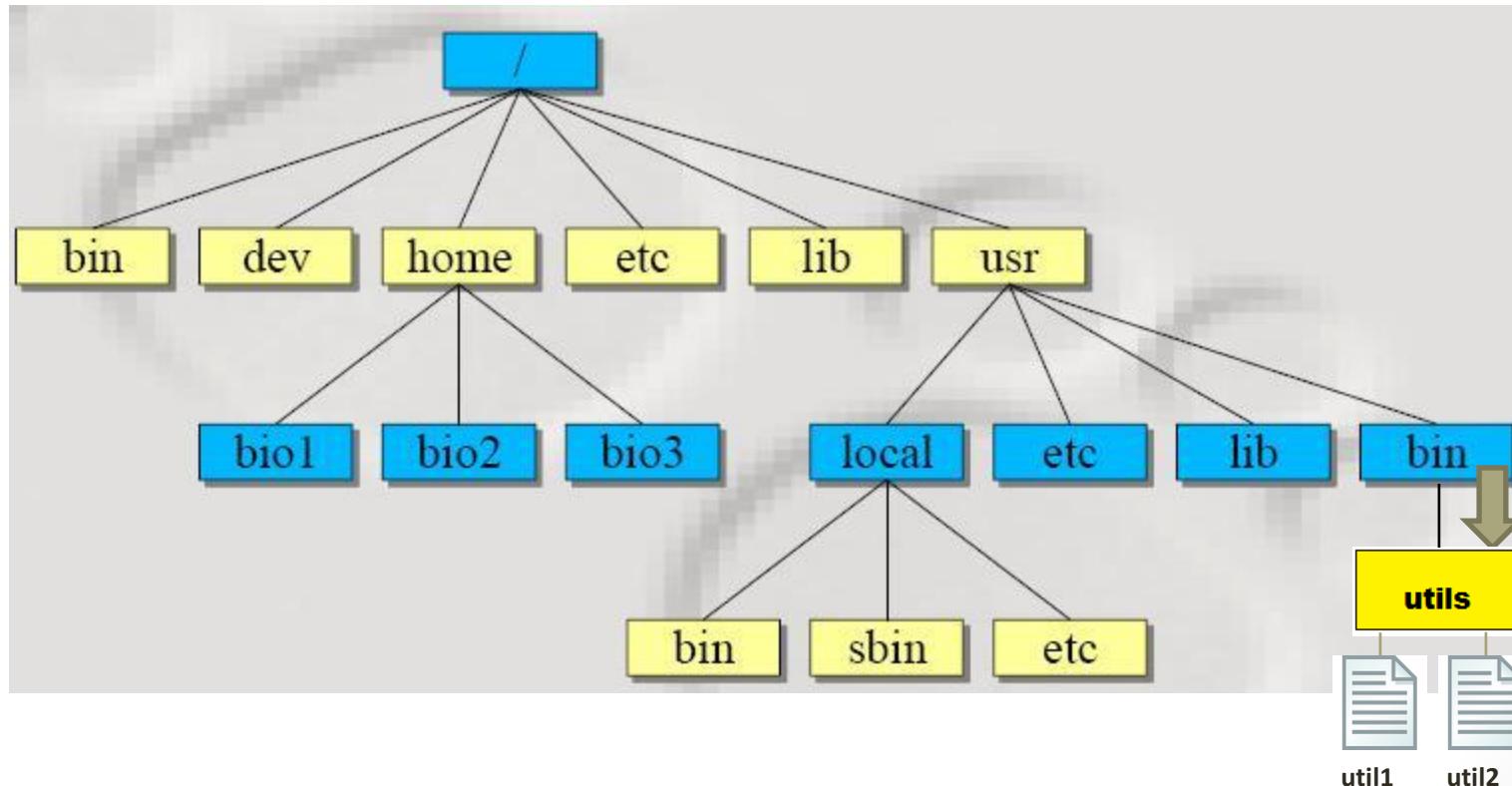


\$ rm prog3 prog2

Removing files and Directories (rm Command)



\$ rm [options] <file or dir list>

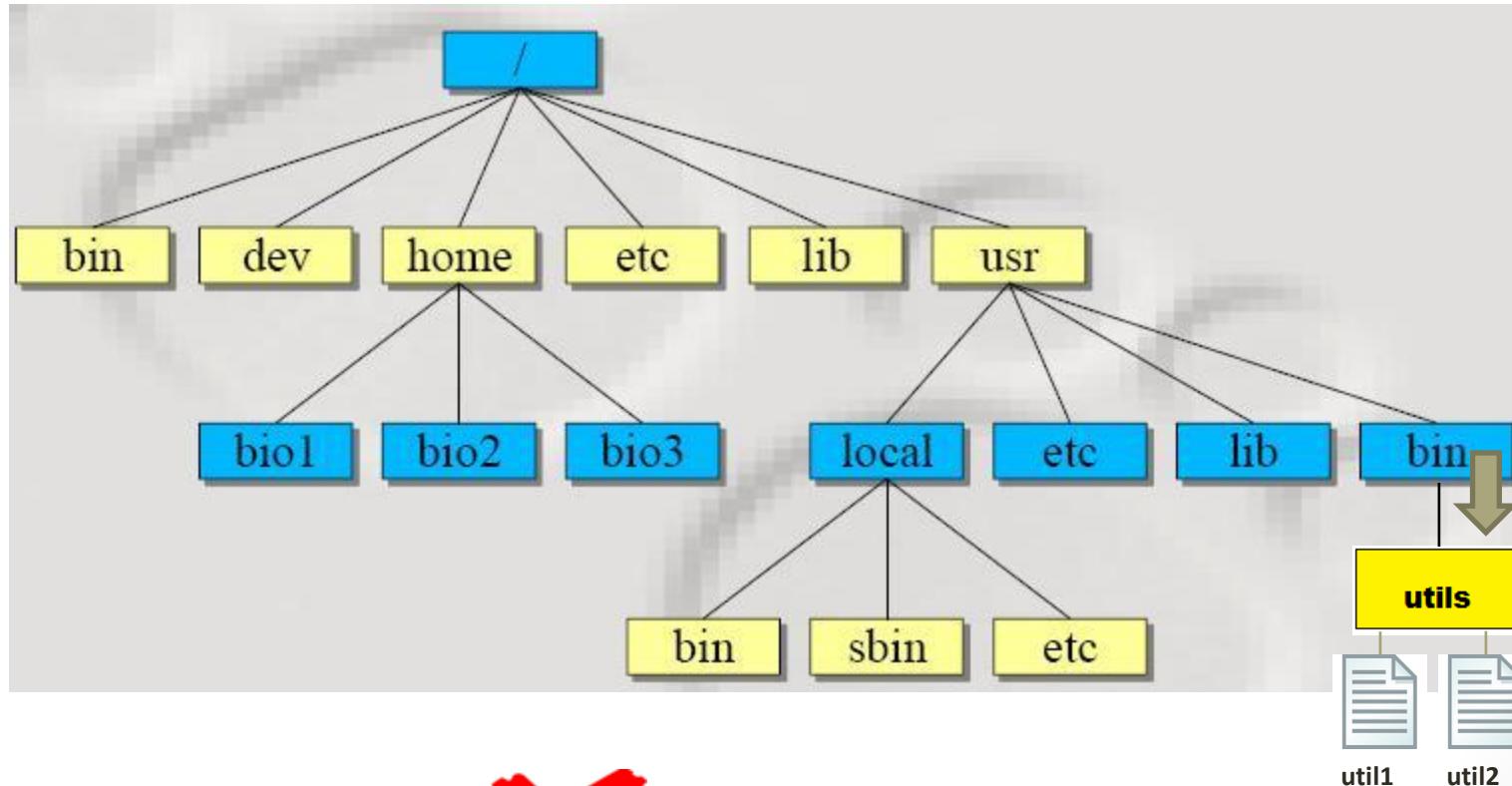


\$ cd /usr/bin/utils

Removing files and Directories (rm Command)



\$ rm [options] <file or dir list>

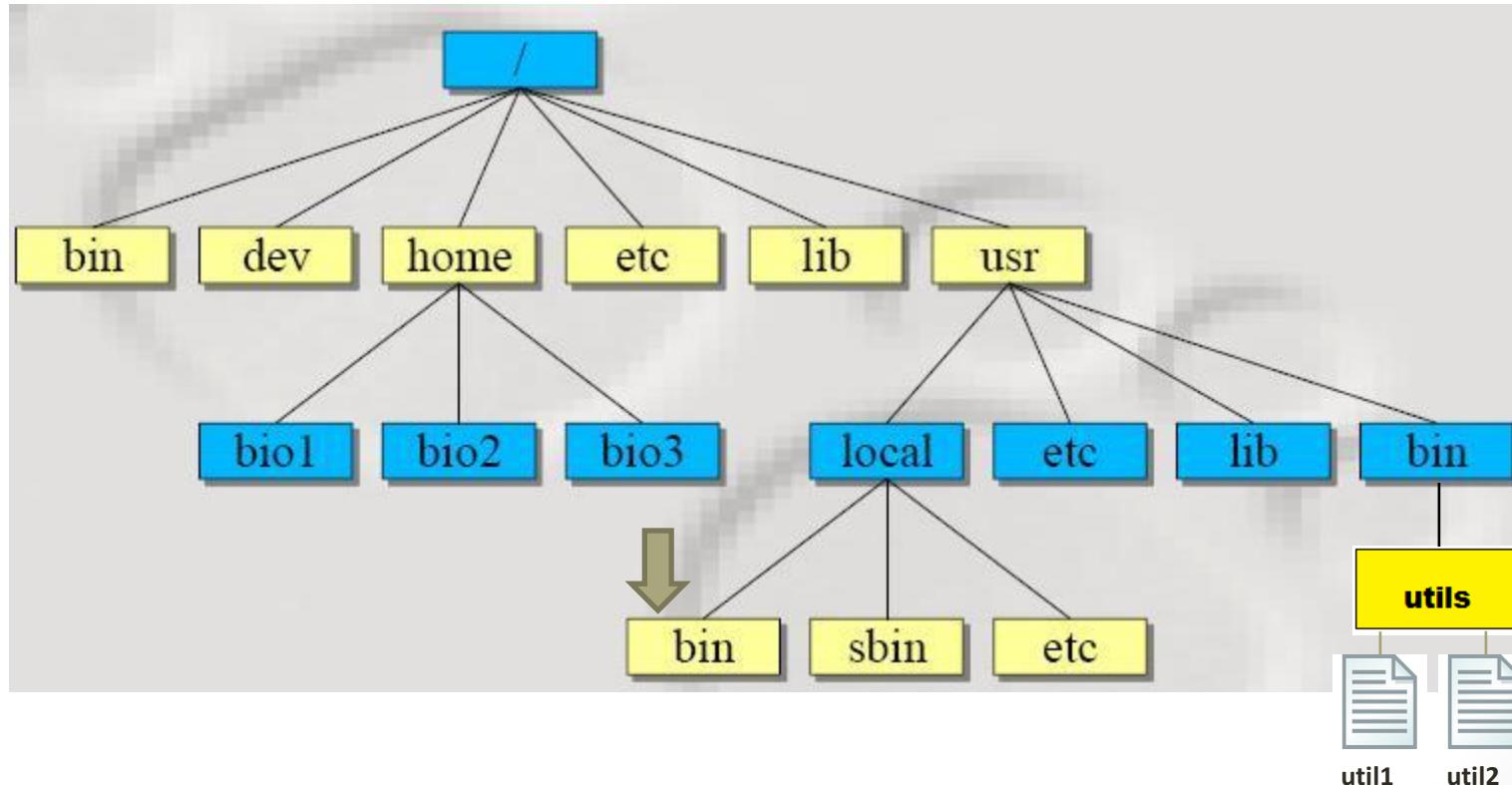


\$ rm -r /usr/bin/utils X

Removing files and Directories (rm Command)



\$ rm [options] <file or dir list>

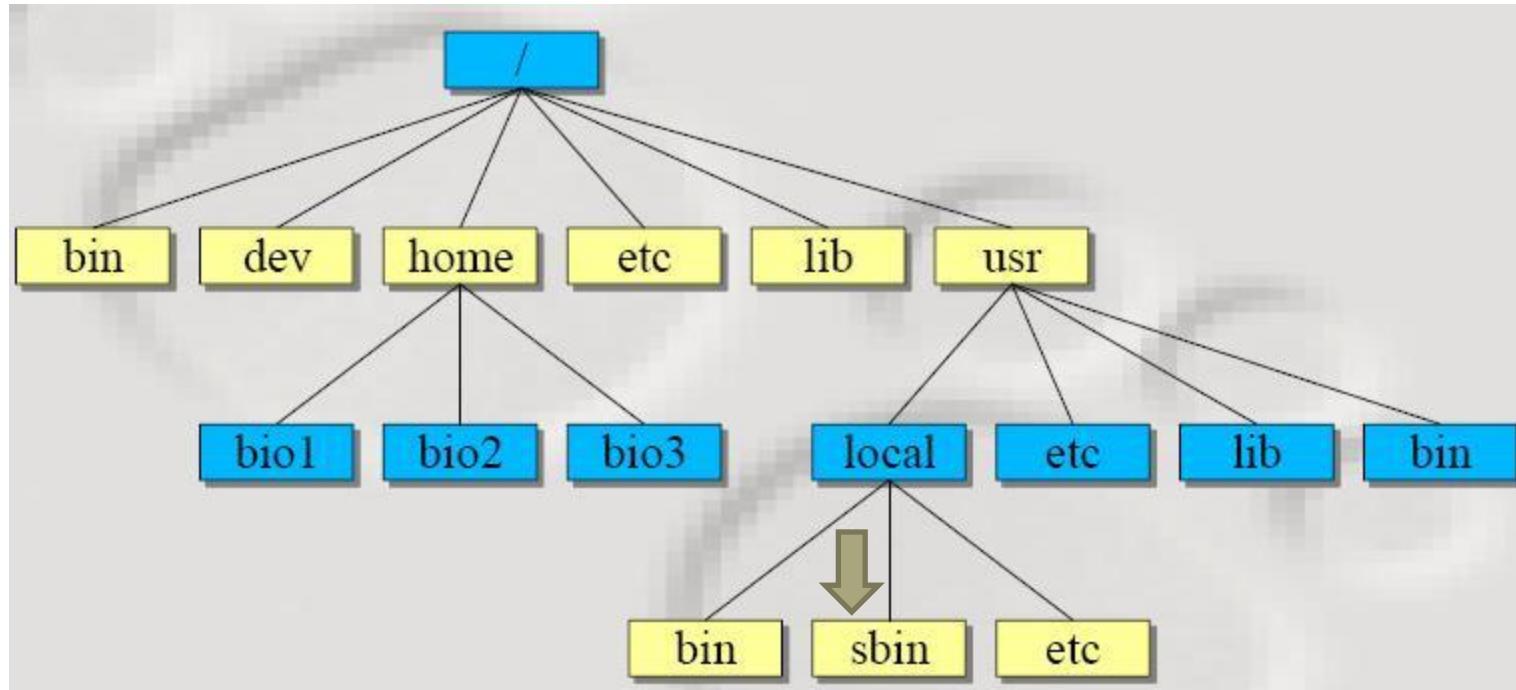


\$ cd -

Removing files and Directories (rm Command)



\$ rm [options] <file or dir list>



\$ rm -r /usr/bin/utils

Removing files and Directories (rm Command)



\$ rm [options] <file or dir list>

\$ rm file1 file2 (remove file1 and file2)

\$ rm -r .../projects/folder1 (remove the folder with its content)

\$ rm -i file1 file2 (interactive, ask me before you remove)

\$ rm -f .../projects/folder1 (force, force remove)

- Note
 - To remove directories you need always to use '-r'
 - You can not remove your current directory (or any of its parents)



Linux 4 Embedded Systems

<http://Linux4EmbeddedSystems.com>



Linux For Embedded Systems

For Arabs

Course 102: Understanding Linux

Ahmed ElArabawy

Lecture 4: Using Wild Cards

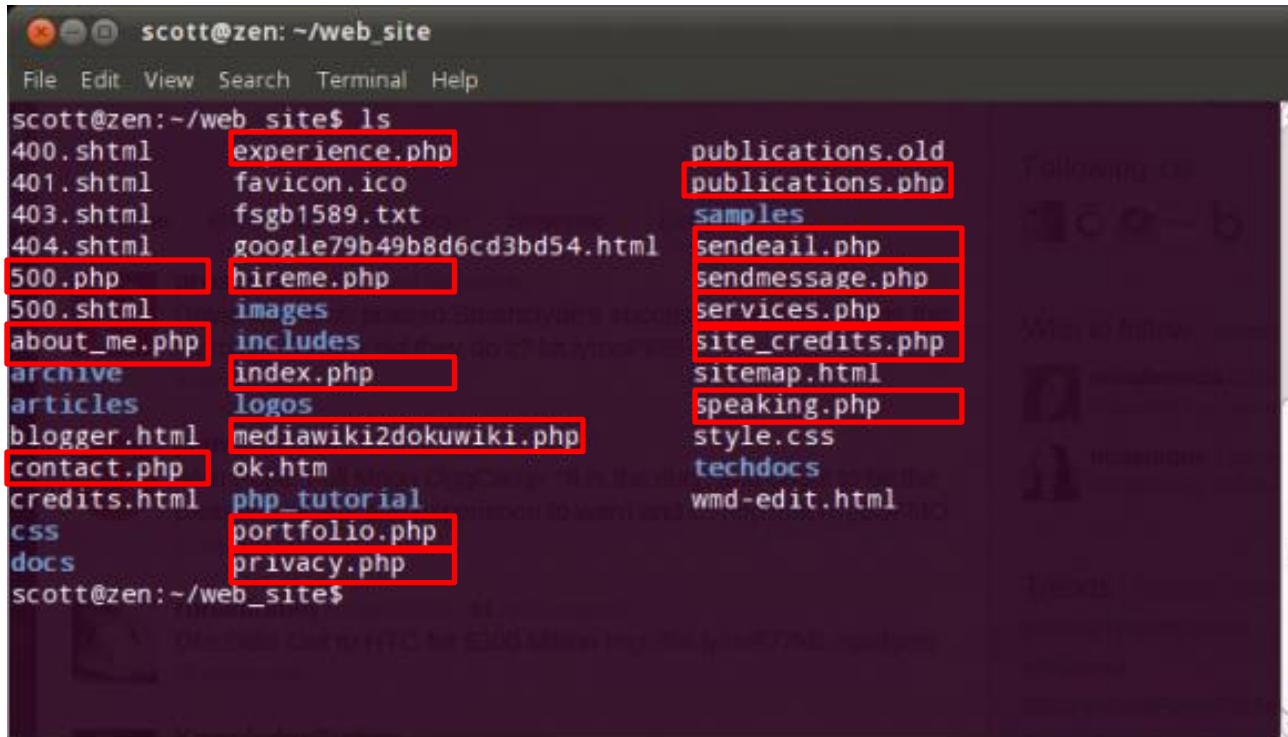


Wild Cards

- Sometimes you will need to execute a command on a group of files instead of a single file
 - Examples:
 - You want to delete all log files
 - You want to list all image files
 - You want to copy old files (ending with .old) to a different place
- The solution for that is to use Wild Cards (also called Globbing)
- Wild cards are patterns that work as placeholders in file names and directory names that are used to apply the command on a group of files/directories that share something in their name
- Remember wild cards are used for file names and directory names ONLY For normal text another patterns are used (Regular Expressions) which will be covered in a future lecture

The “*” Wild Card

- The “*” can replace any set of characters (including none) in the file/directory name

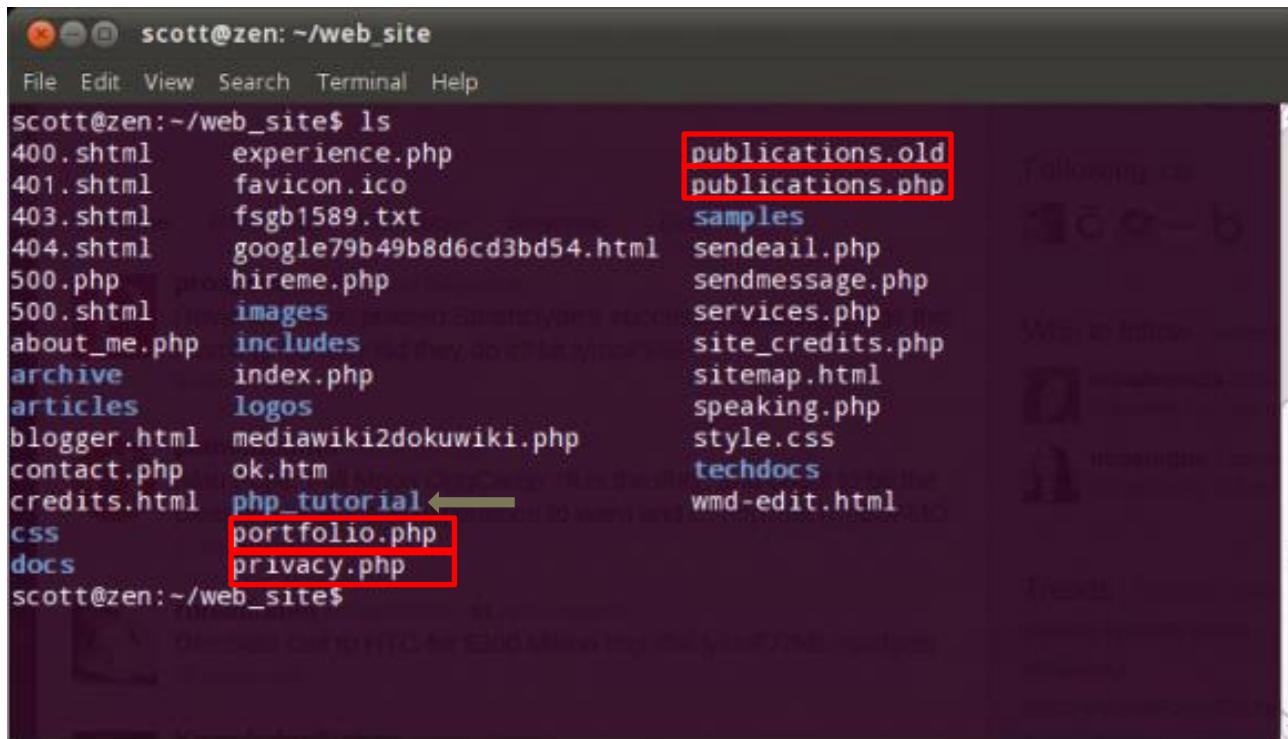


```
scott@zen:~/web_site$ ls
400.shtml      experience.php          publications.old
401.shtml      favicon.ico            publications.php
403.shtml      fsgb1589.txt           samples
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php             sendemail.php
500.shtml      images                 sendMessage.php
about_me.php    includes               services.php
archive        index.php              site_credits.php
articles       logos                 sitemap.html
blogger.html   mediawiki2dokuwiki.php style.css
contact.php    ok.htm                techdocs
credits.html   php_tutorial          wmd-edit.html
css            portfolio.php          wwd-edit.html
docs           privacy.php           wwd-edit.html
scott@zen:~/web_site$
```

\$ rm *.php

The “*” Wild Card

- The “*” can replace any set of characters (including none) in the file/directory name

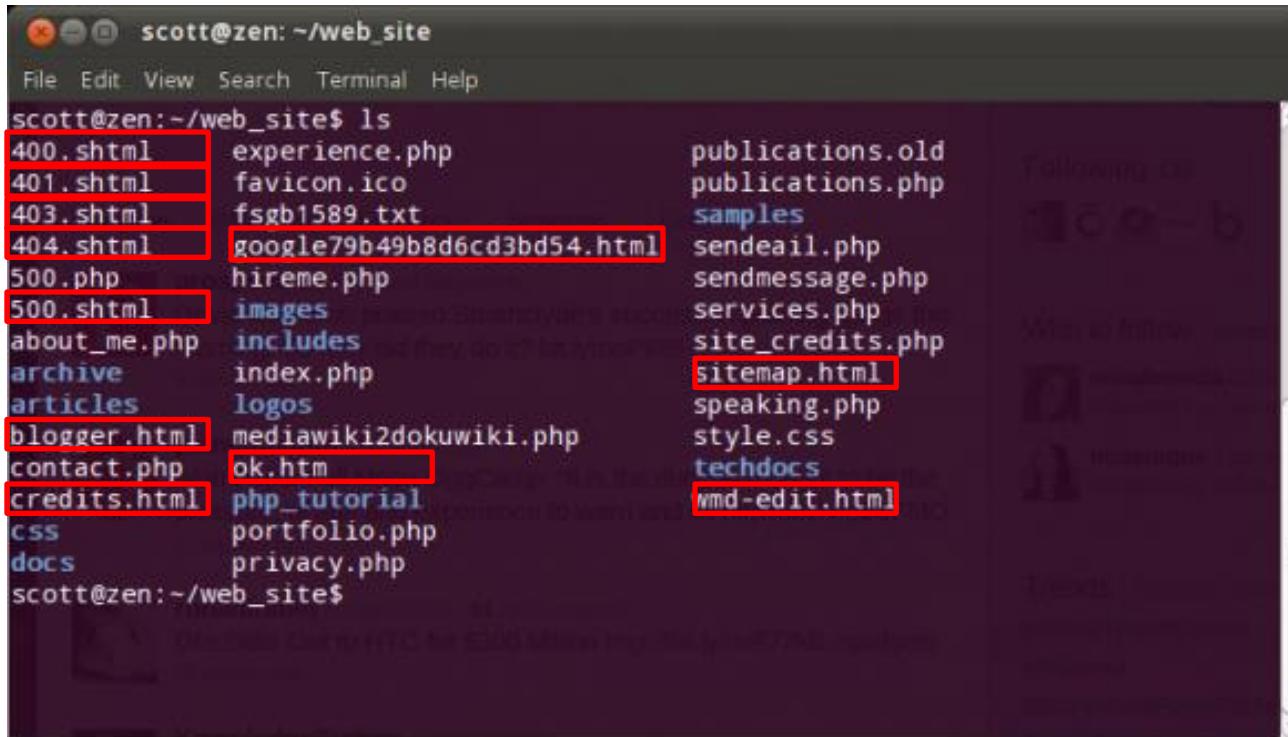


```
scott@zen:~/web_site$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive       index.php
articles       logos
blogger.html   mediawiki2dokuwiki.php
contact.php    ok.htm
credits.html   php_tutorial
css            portfolio.php
docs           privacy.php
scott@zen:~/web_site$
```

\$ rm p*

The “*” Wild Card

- The “*” can replace any set of characters (including none) in the file/directory name

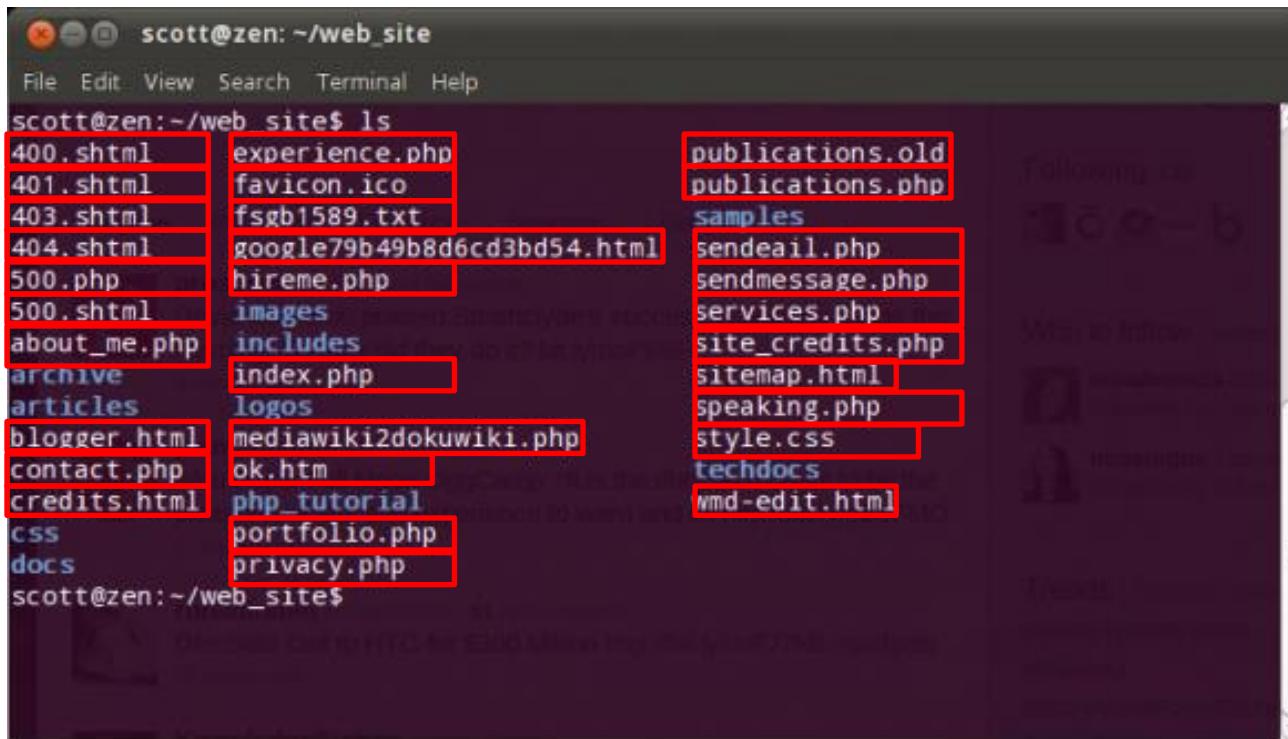


```
scott@zen:~/web_site$ ls
400.shtml      experience.php          publications.old
401.shtml      favicon.ico           publications.php
403.shtml      fsgb1589.txt          samples
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php            sendemail.php
500.shtml      images                sendMessage.php
about_me.php   includes              services.php
archive       index.php             site_credits.php
articles      logos                sitemap.html
blogger.html   mediawiki2dokuwiki.php
contact.php    ok.htm               speaking.php
credits.html   php_tutorial         style.css
css           portfolio.php        techdocs
docs          privacy.php          wmd-edit.html
scott@zen:~/web_site$
```

\$ rm *.*htm*

The “*” Wild Card

- The “*” can replace any set of characters (including none) in the file/directory name



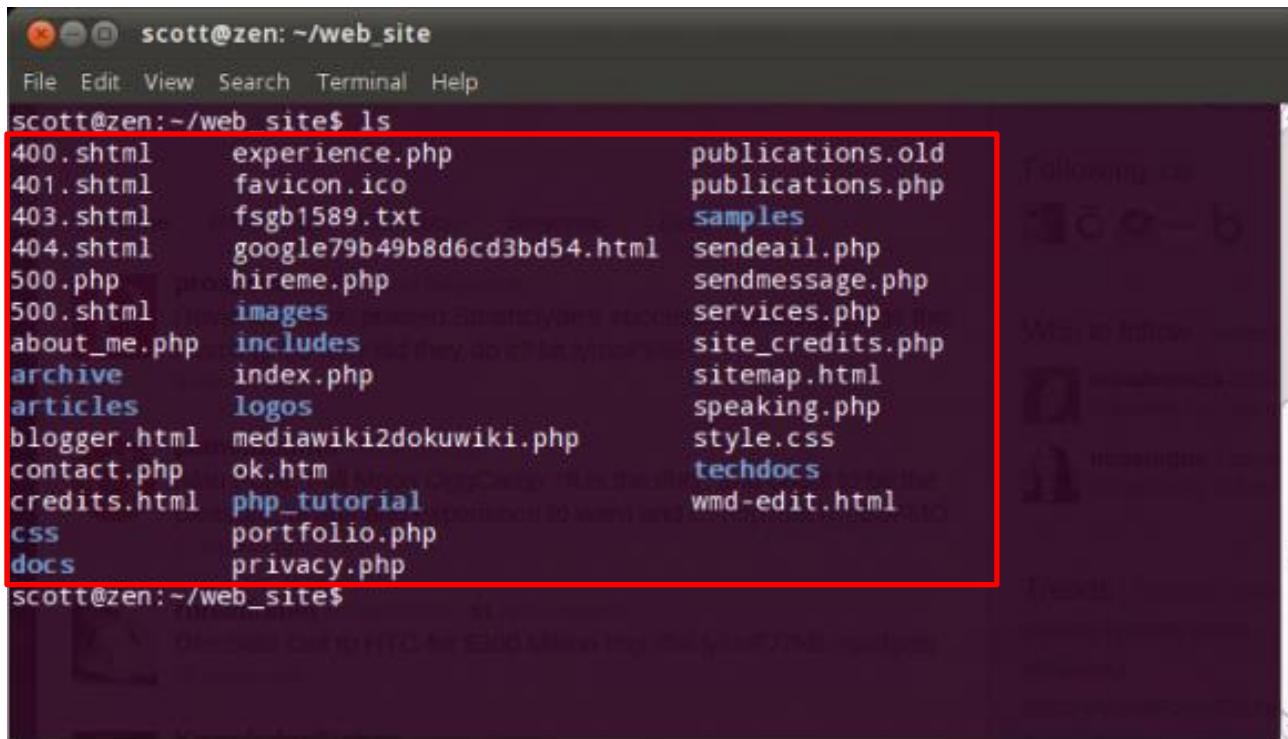
scott@zen: ~/web_site\$ ls

400.shtml
401.shtml
403.shtml
404.shtml
500.php
500.shtml
about_me.php
archive
articles
blogger.html
contact.php
credits.html
css
docs
experience.php
favicon.ico
fsgb1589.txt
google79b49b8d6cd3bd54.html
hireme.php
images
includes
index.php
logos
mediawiki2dokuwiki.php
ok.htm
php_tutorial
portfolio.php
privacy.php
publications.old
publications.php
samples
sendemail.php
sendMessage.php
services.php
site_credits.php
sitemap.html
speaking.php
style.css
techdocs
wmd-edit.html

\$ rm -r *.*

The “*” Wild Card

- The “*” can replace any set of characters (including none) in the file/directory name

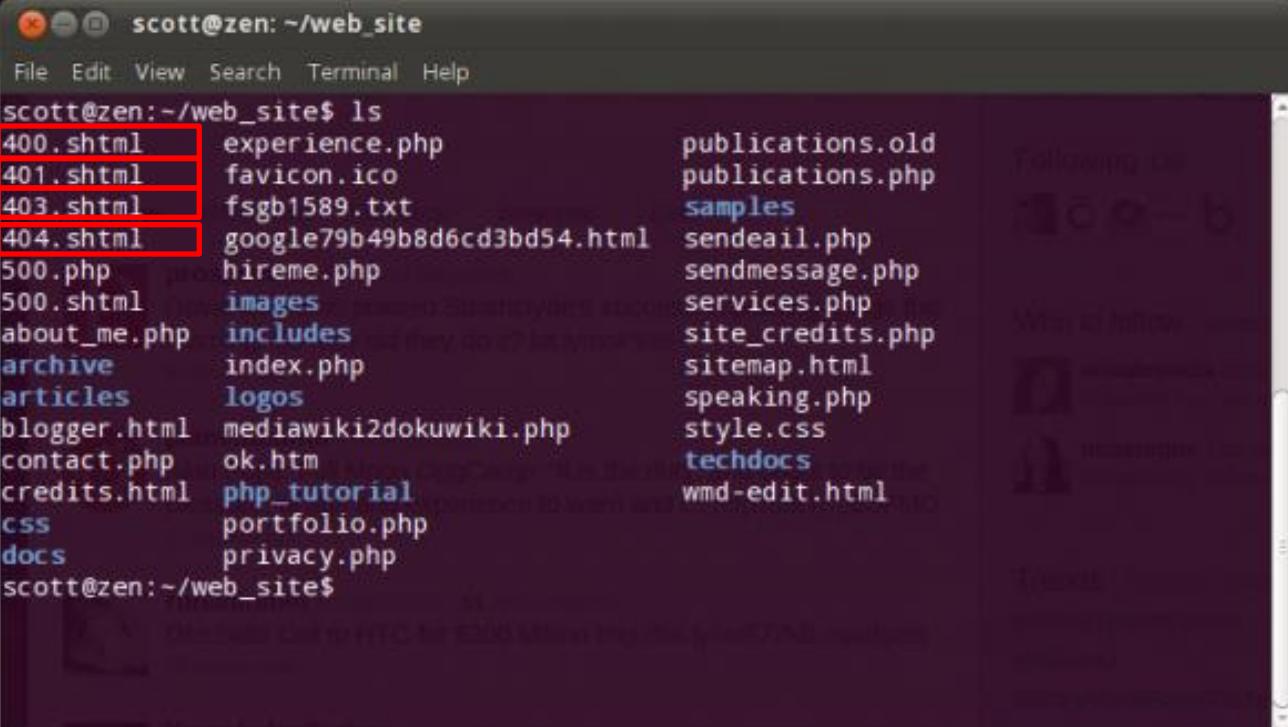


```
scott@zen:~/web_site
File Edit View Search Terminal Help
scott@zen:~/web_site$ ls
400.shtml      experience.php           publications.old
401.shtml      favicon.ico            publications.php
403.shtml      fsgb1589.txt          samples
404.shtml      google79b49b8d6cd3bd54.html sendemail.php
500.php        hireme.php            sendMessage.php
500.shtml      images                services.php
about_me.php    includes              site_credits.php
archive        index.php             sitemap.html
articles       logos                speaking.php
blogger.html   mediawiki2dokuwiki.php style.css
contact.php    ok.htm               techdocs
credits.html   php_tutorial         wmd-edit.html
css            portfolio.php
docs           privacy.php
scott@zen:~/web_site$
```

\$ rm -r *

The “?” Wild Card

- The “?” wild card stands for any single character

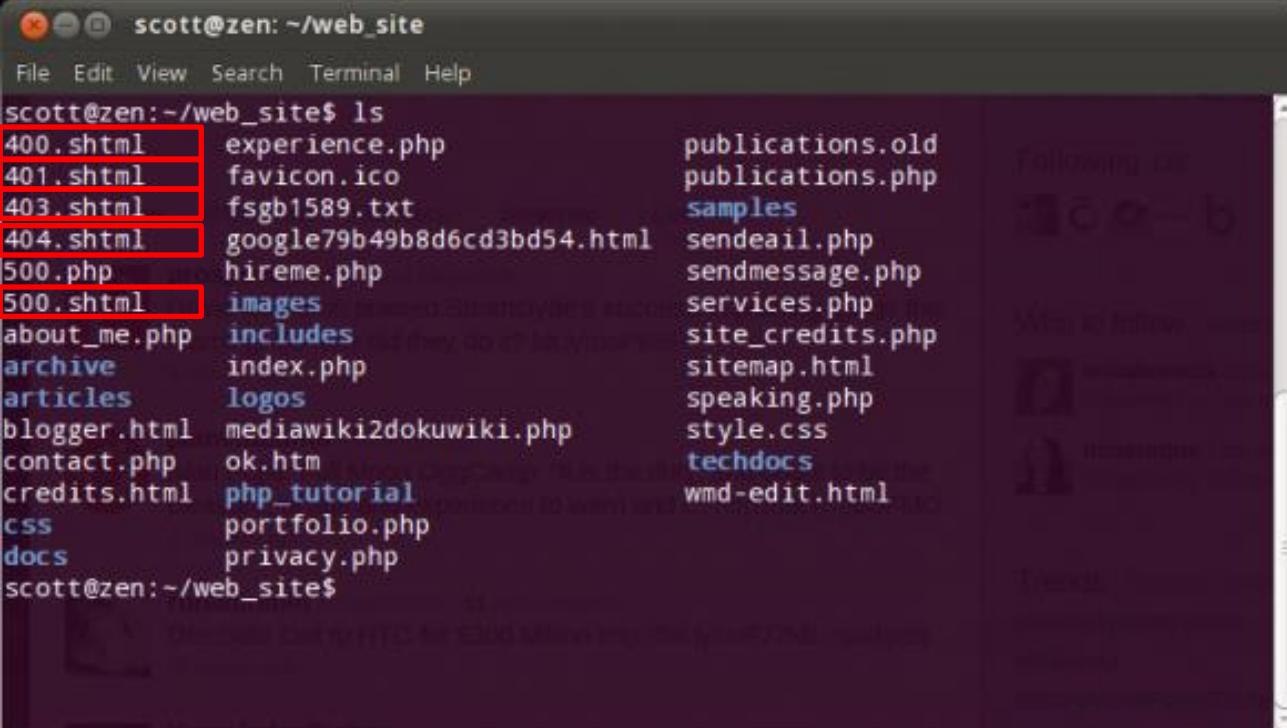


```
scott@zen:~/web_site
File Edit View Search Terminal Help
scott@zen:~/web_site$ ls
400.shtml      experience.php           publications.old
401.shtml      favicon.ico            publications.php
403.shtml      fsgb1589.txt          samples
404.shtml      google79b49b8d6cd3bd54.html sendemail.php
500.php        hireme.php            sendMessage.php
500.shtml      images                services.php
about_me.php   includes              site_credits.php
archive       index.php             sitemap.html
articles      logos                speaking.php
blogger.html  mediawiki2dokuwiki.php style.css
contact.php   ok.htm               techdocs
credits.html  php_tutorial         wmd-edit.html
css           portfolio.php
docs          privacy.php
scott@zen:~/web_site$
```

\$ rm 40?.shtml

The “?” Wild Card

- The “?” wild card stands for any single character



```
scott@zen: ~/web_site
File Edit View Search Terminal Help
scott@zen:~/web_site$ ls
400.shtml      experience.php           publications.old
401.shtml      favicon.ico            publications.php
403.shtml      fsgb1589.txt          samples
404.shtml      google79b49b8d6cd3bd54.html sendemail.php
500.php        hireme.php            sendMessage.php
500.shtml      images                services.php
about_me.php   includes              site_credits.php
archive       index.php             sitemap.html
articles      logos                speaking.php
blogger.html  mediawiki2dokuwiki.php style.css
contact.php   ok.htm               techdocs
credits.html  php_tutorial         wmd-edit.html
css           portfolio.php
docs          privacy.php
scott@zen:~/web_site$
```

\$ rm ?0?.shtml



[<chars>] and [<chars>]

- We can have more restriction than the use of "?" by specifying a limited set of options for the character

“**[ars]**” : Stands for a Single character from the list a,r,s

“**[!ars]**” : Stands for any Single character except for the list a,r,s

“**[2-5]**”: Stands for a Single character from the range 2 to 5

“**[!2-5]**”: Stands for any Single character except for the list 2 to 5

“**[a-I]**” : Stands for a Single character from range of ‘a’ to ‘I’

“**[!a-I]**” : Stands for any Single character except for the list a to I

“**[1-37-9]**”: Stands for 1,2,3,7,8,9

“**[a-chk]**”: Stands for a,b,c,h,k

- Examples:

\$ rm -r ab[c-fh-j]

removes the files/folders named abc,abd,abe,abf, abh,abi,abj

\$ ls results-[0-9][0-9].log

lists the files named results-00.log to results-99.log

[[:<Class Name>:]]

- “**“[[<class name>:]]”** stands for a single character belonging to the specified class
- Some of used classes,
 - [[:alnum:]]** Alpha Numeric characters (a-z, A-Z, 0-9)
 - [[:alpha:]]** Alphabets (a-z, A-Z)
 - [[:digit:]]** Digits (0-9)
 - [[:lower:]]** Lower case character (a-z)
 - [[:upper:]]** Upper case character (A-Z)
- Examples:
\$ cp results-[[:digit:]][[:digit:]]-[[:alpha:]].log ~ / log



Use of Curly Brackets “{ }”

- Curly brackets are used to group selections
- Examples,

\$ rm {*.log, *.txt}

\$ cp {*.pdf, *.doc} ~/documents/

- Note:

This also works

\$ rm *.log *.txt

\$ cp *.pdf *.doc ~/documents/



A nice tip,

- If you are going to remove a bunch of files/directories using wild cards, a good idea is to list them first with the same wildcard pattern to make sure you are not doing a mistake and removing the wrong files, then replace “*ls*” with an “*rm*”
- Or if not sure, use the ‘*-i*’ to confirm each file delete before it is deleted



Escape Sequence “\”

- Some special letters has a meaning (such as **space**, *, ", ', (, ...)
- It is not recommended to use these letters in file/directory names
- But, if we have to then there is a special way of dealing with them,
- If we need to delete a file named “my results.txt”

\$ rm my results.txt

\$ rm my\ results.txt

- This is called “Escaping the space letter” which means changing its default meaning from a separator letter into a general letter inside the filename



Examples of Escape Sequence

Real File Name	File Name representation
my?file.log	my\?file.log
my*file.log	my*file.log
my file (today).txt	my\ file\ \\\(today\\).txt
“my file”	\\"my\ file\\"
abc[!2]	abc\\[!2\\]



Linux 4 Embedded Systems

<http://Linux4EmbeddedSystems.com>



Linux For Embedded Systems

For Arabs

Course 102: Understanding Linux

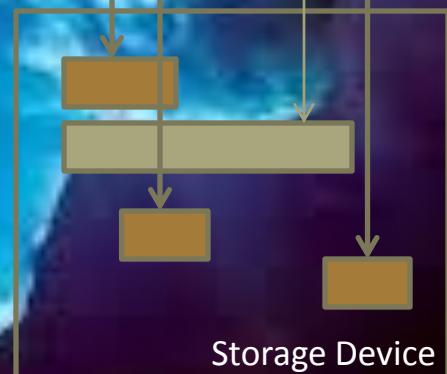
Ahmed ElArabawy

Lecture 5: File Handling Internals



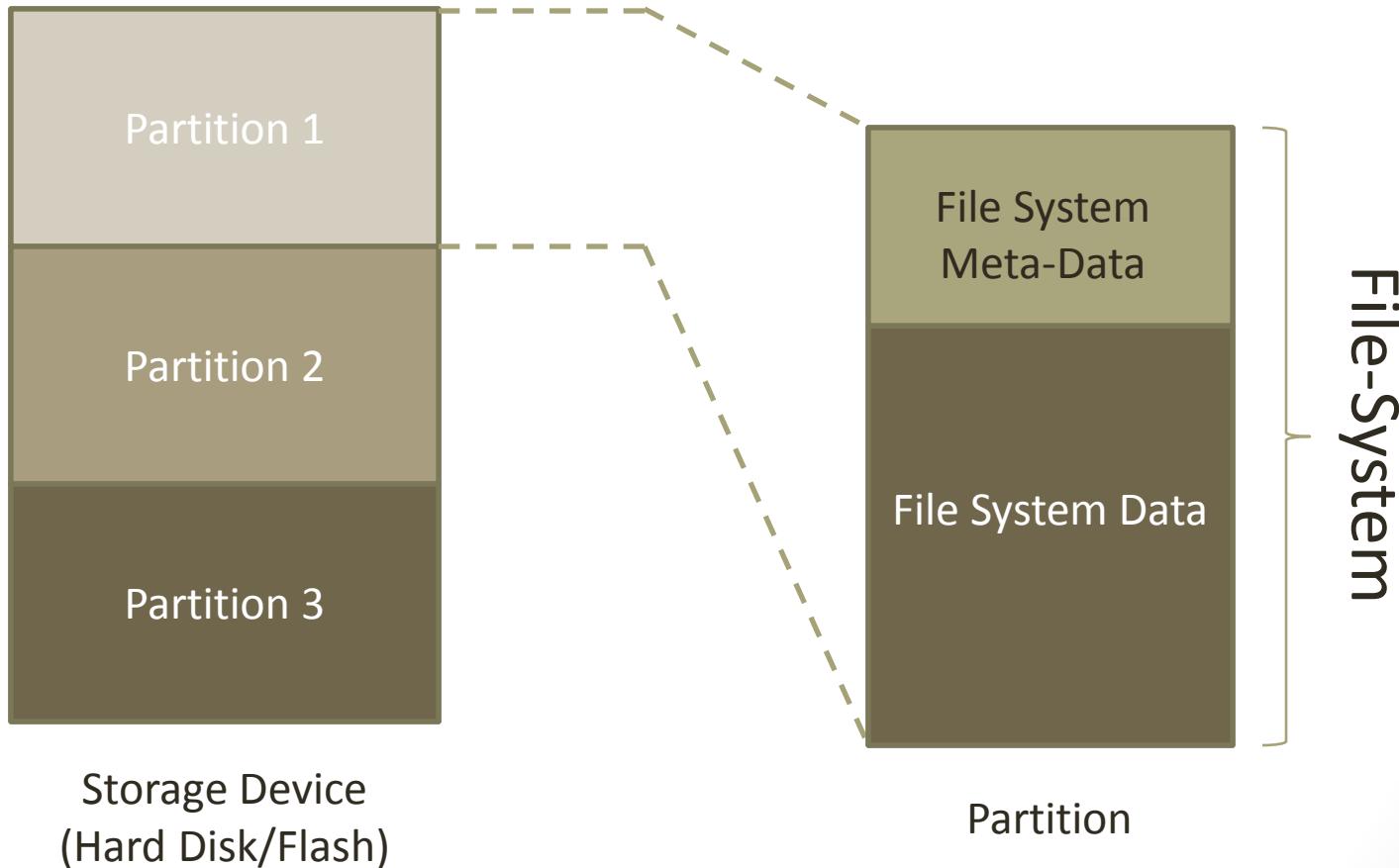
```
novisetiawatri@novisetiawatri-K84L:~/tutorial$ ls -all
total 8
drwxrwxr-x  2 novisetiawatri novisetiawatri 4096 Nov 25 08:34 .
drwxr-xr-x 31 novisetiawatri novisetiawatri 4096 Nov 25 08:35 ..
-rw-rw-r--  1 novisetiawatri novisetiawatri    0 Oct 10 10:38 file1.txt
-rw-rw-r--  1 novisetiawatri novisetiawatri    0 Oct 10 10:40 file2.txt
-rw-rw-r--  1 novisetiawatri novisetiawatri    0 Oct 10 10:40 file3.txt
-rw-rw-r--  1 novisetiawatri novisetiawatri    0 Oct 10 10:40 file4.txt
-rw-rw-r--  1 novisetiawatri novisetiawatri    0 Oct 10 10:40 file5.txt
```

File System

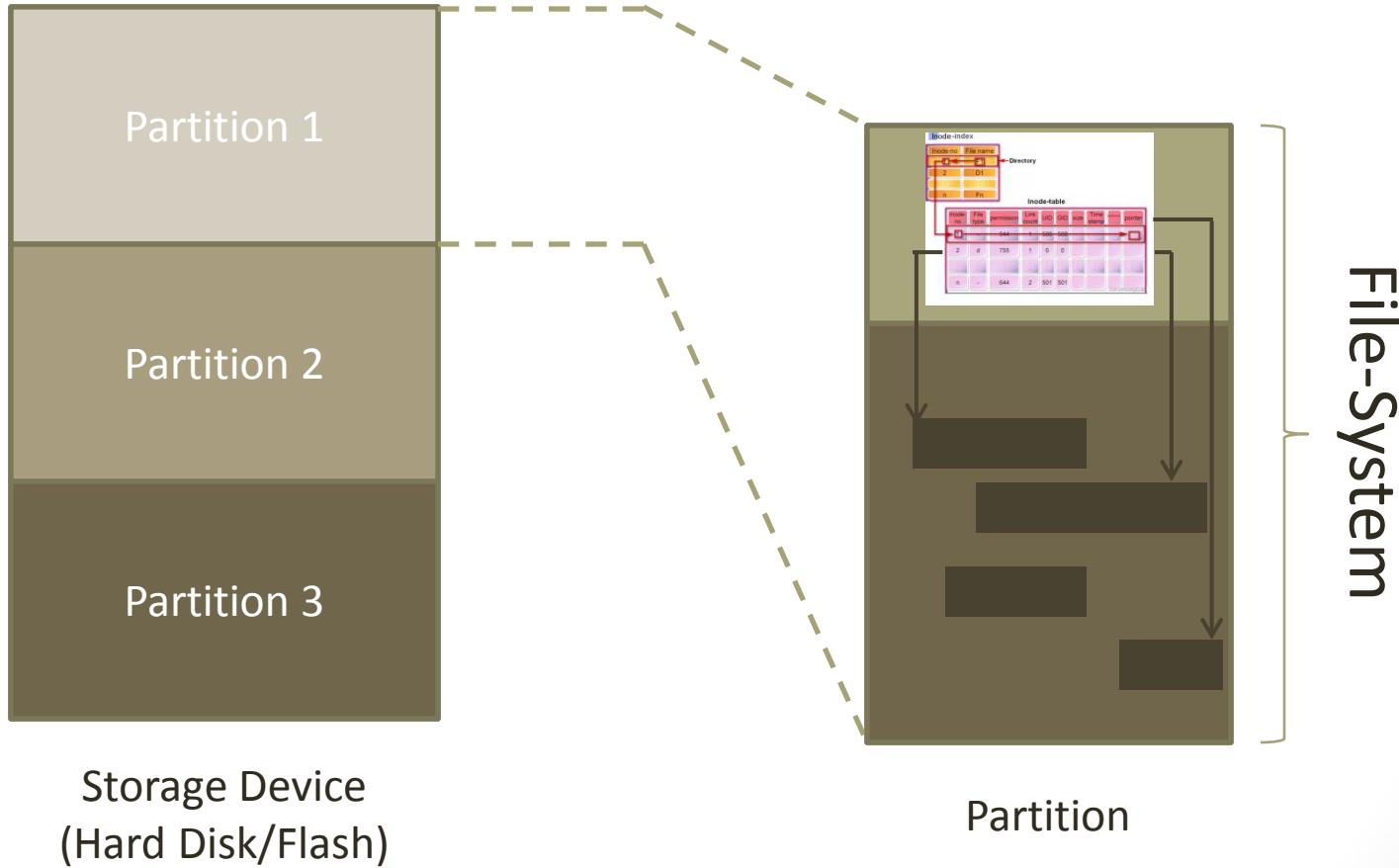


Storage Device

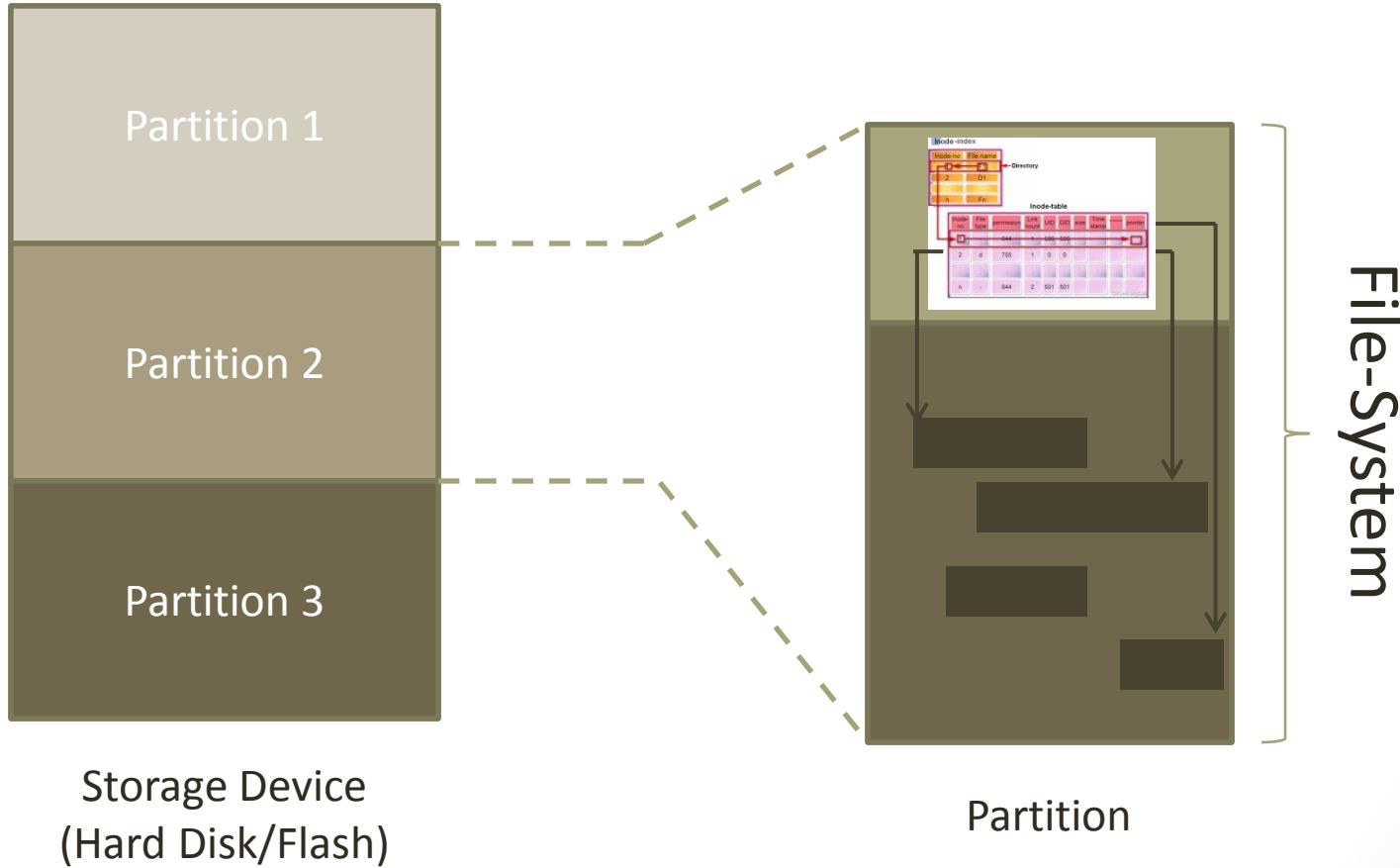
FileSystem



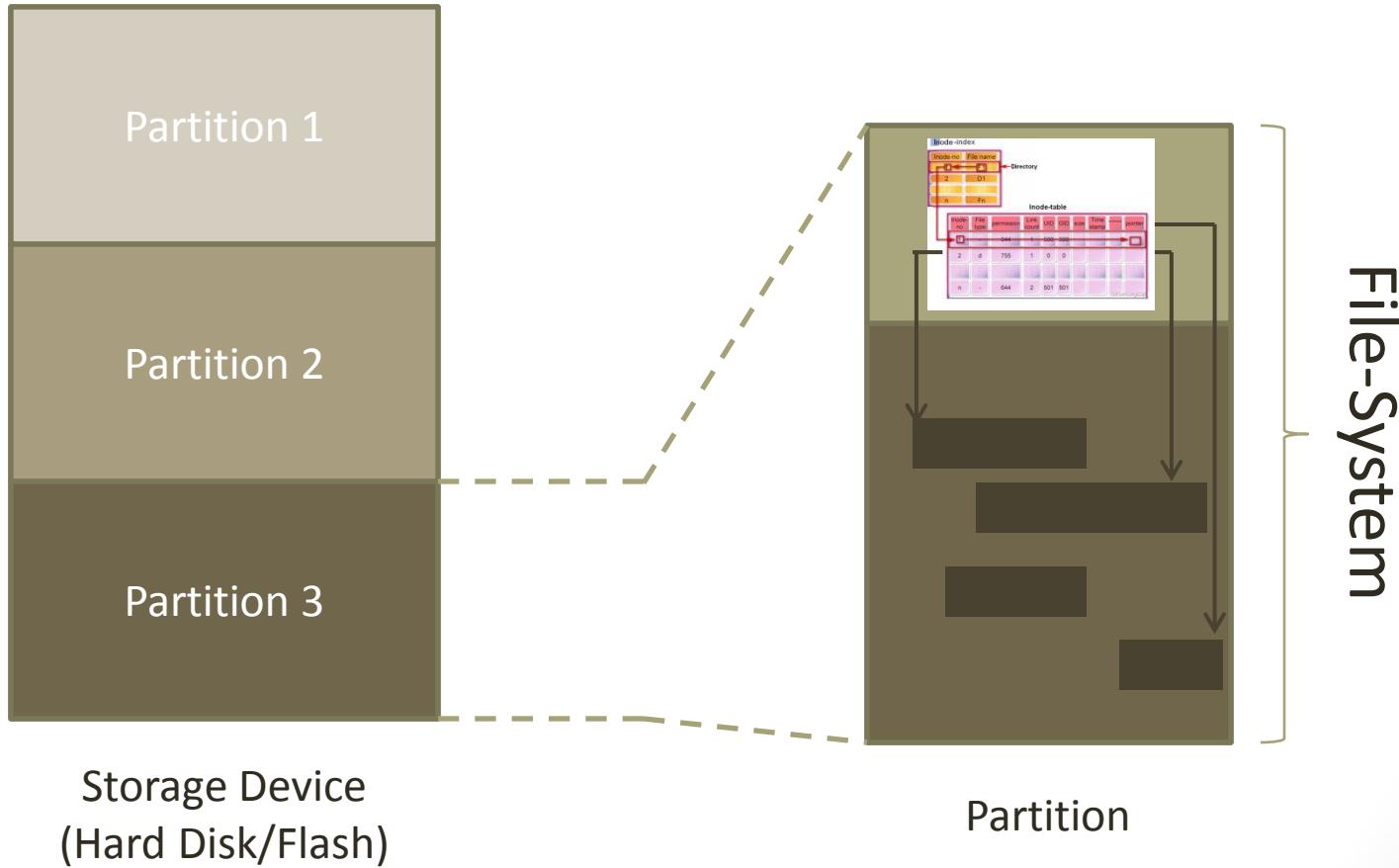
File-System



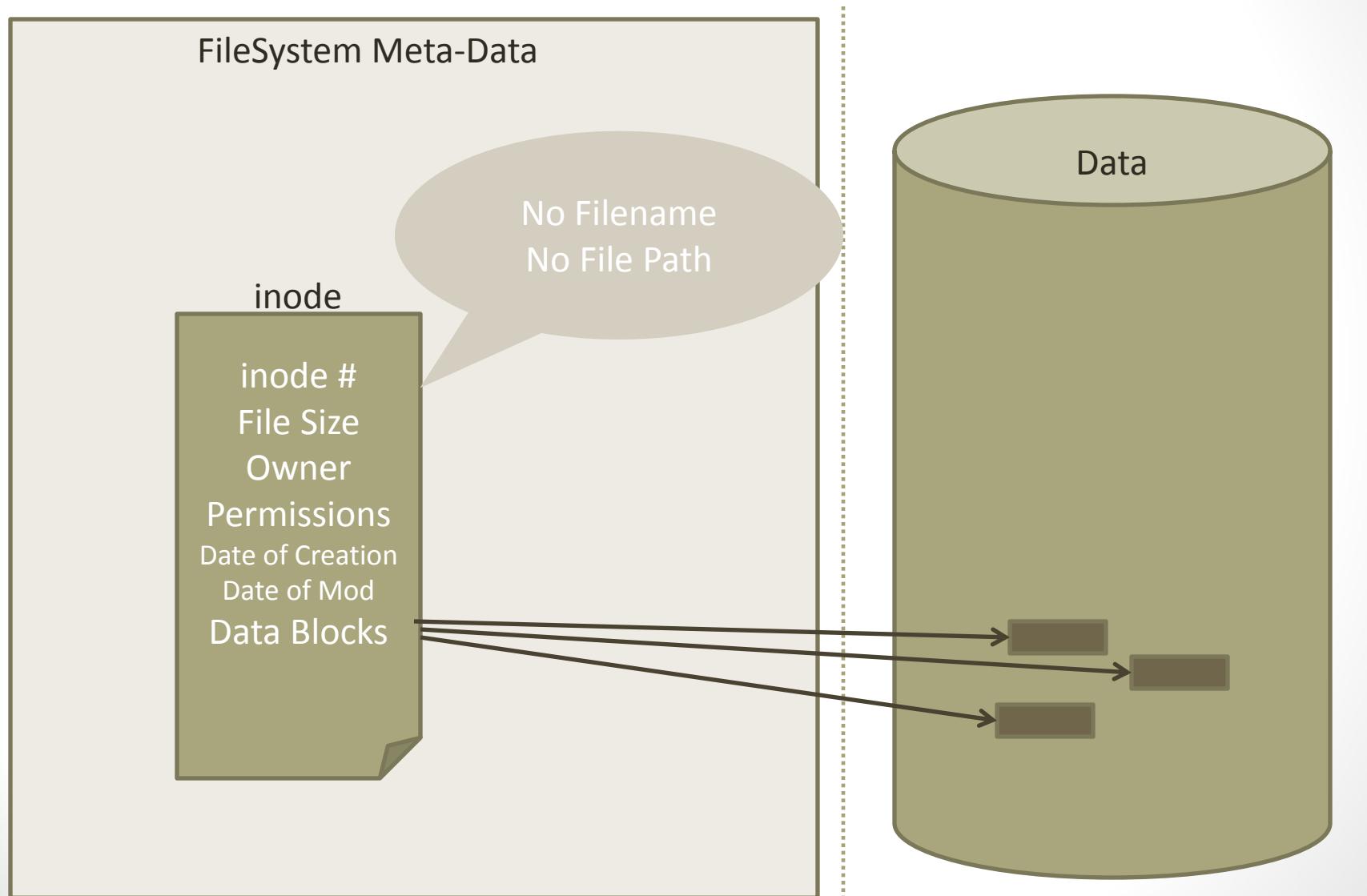
File-System



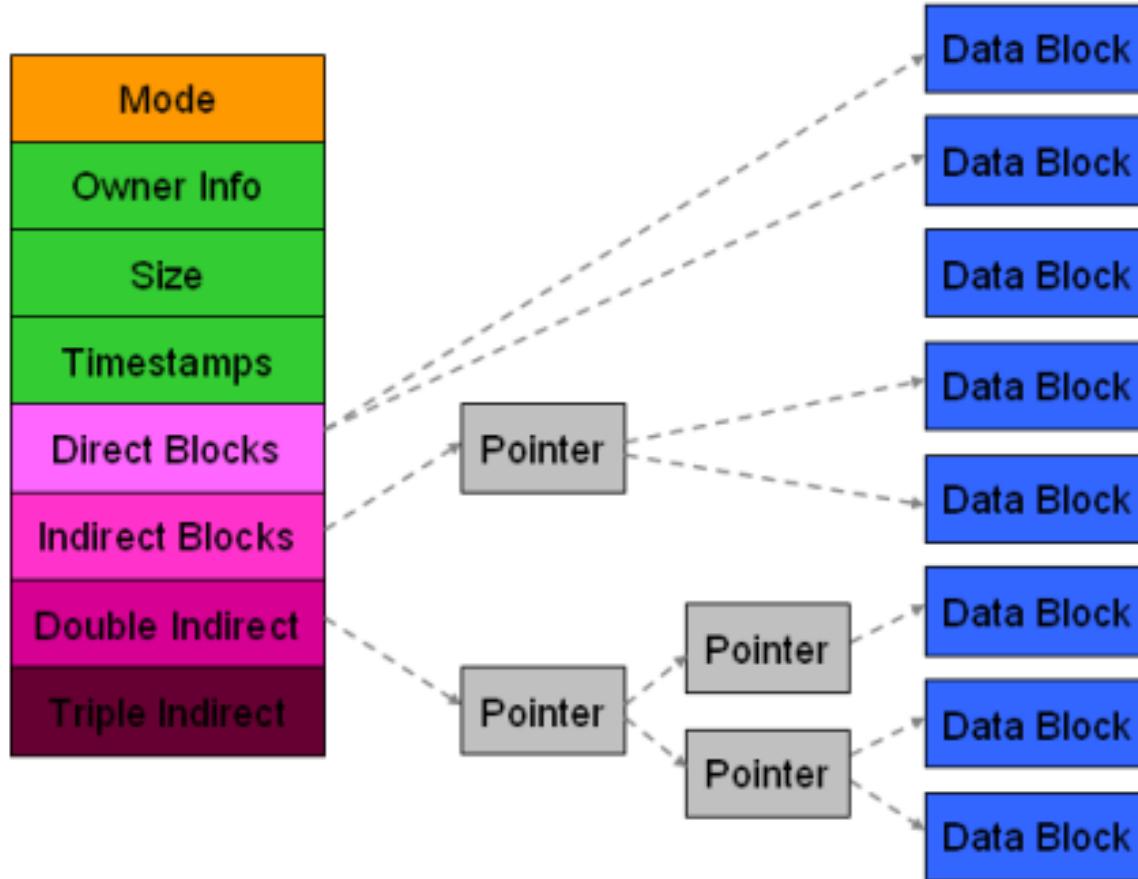
File-System



What is a File ???



The “inode” Structure



What is a File ??

- A file is a set of bytes that represent some content (pdf document, excel sheet, binary executable, ...)
- The file is stored in a (partition in a) storage device as a single data block or fragmented into a group of data blocks (within the same partition)
- The fileSystem is responsible for managing the data block(s), and their representation to the user
- For this management, the fileSystem needs to maintain some extra info about the file which is called file Meta-data
 - File Size
 - File Owner (user & group)
 - File Permissions
 - Date of creation/last modification
 - Pointers to the file content data blocks
 - etc ...
- These meta-data are stored in an “inode” structure
- Note: the inode does not contain the file name or its location



What is a File ??

- This means that the filesystem maintains a table of inode structures (one structure per file)
- The “inode” structure will contain all file meta-data (except its filename)
- The “inode” structure will also point to the data blocks of the file
- Each data node has a unique number across the filesystem (the inode number)
- Inode numbers are unique per filesystem (and not across the system)
- Directories are a special type of files, accordingly, they are treated the same way

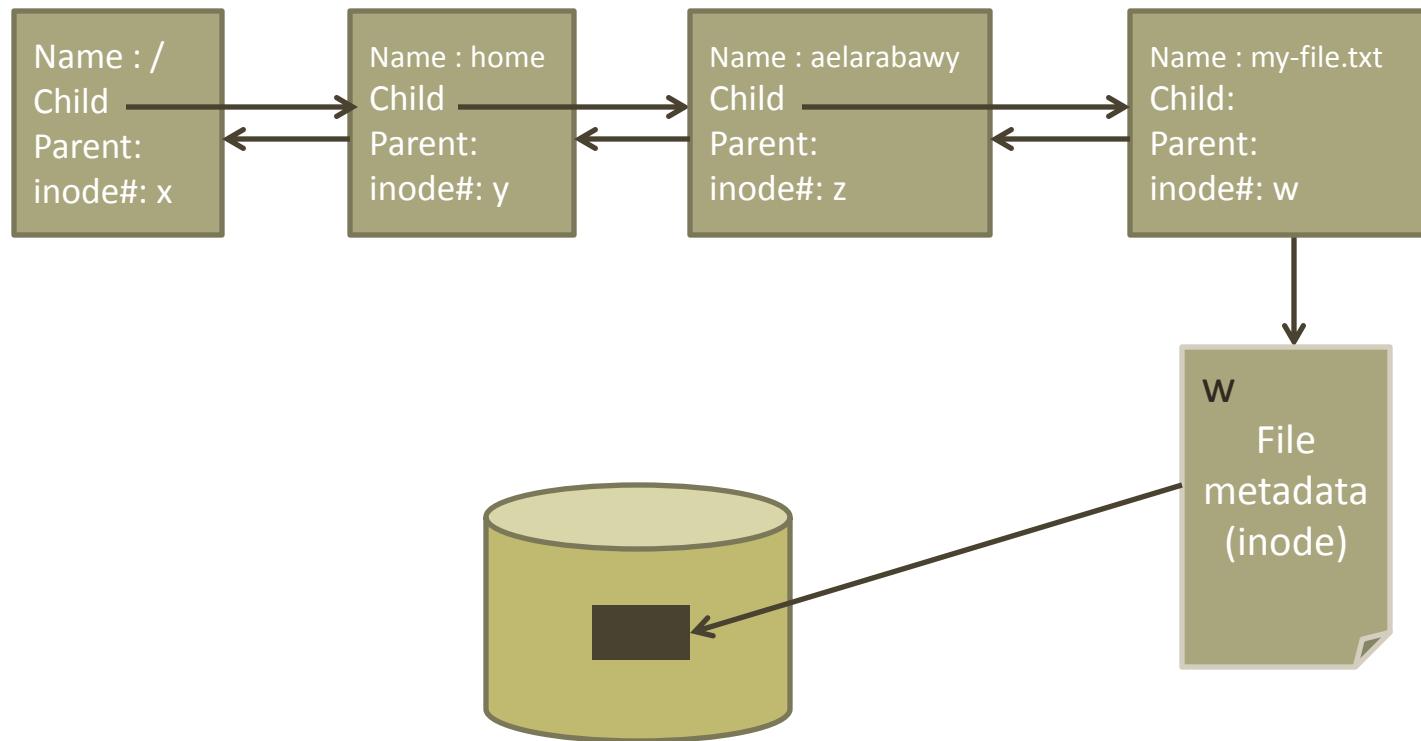


So, What about Filenames & paths ??

- The inode structure does not have knowledge about the filename or its location
- Instead, each file or directory has another structure named “**dentry**”, this structure maps a file/directory to its “**inode#**”
- The “**dentry**” structure forms the directory tree

Example

/home/aelarabawy/my-file.txt



dentry Cache

Inode-no	File name
1	F1
2	D1
:	:
n	Fn

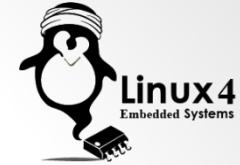
Inode-table

Inode-no	File type	permission	Link count	UID	GID	size	Time stamp	pointer
1	-	644	1	500	500				
2	d	755	1	0	0				
:	:	:	:	:	:	:			
n	-	644	2	501	501				



LINUX COMMANDS

Listing Files/Directories (ls Command)



\$ ls -i (List with showing the inode#)

```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ls -i
2884089 bin          2632359 log        2621634 software
2621449 Desktop      2621454 Music      2621451 Templates
3939335 directorName 2621726 Perforce   2621456 Videos
2621453 Documents    2621455 Pictures   2759177 work
2621450 Downloads     2621452 Public
2621445 examples.desktop 2759157 sketchbook
aelarabawy@aelarabawy-demo-backup64:~$ █
```

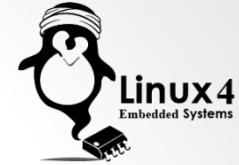
Listing Files/Directories (ls Command)



\$ ls -il (List with showing the inode# with long format)

```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ls -il
total 112
2884089 drwxrwxr-x 5 aelarabawy aelarabawy 4096 Dec 10 10:15 bin
2621449 drwxr-xr-x 2 aelarabawy aelarabawy 4096 Nov 21 14:52 Desktop
3939335 drwxrwxr-x 2 aelarabawy aelarabawy 4096 Mar 21 17:30 directorName
2621453 drwxr-xr-x 2 aelarabawy aelarabawy 4096 Nov 21 14:52 Documents
2621450 drwxr-xr-x 4 aelarabawy aelarabawy 4096 Mar 19 14:33 Downloads
2621445 -rw-r--r-- 1 aelarabawy aelarabawy 8445 Apr 16 2012 examples.desktop
2632359 -rw-rw-r-- 1 aelarabawy aelarabawy 42289 Apr 11 10:32 log
2621454 drwxr-xr-x 2 aelarabawy aelarabawy 4096 Nov 21 14:52 Music
2621726 drwxrwxr-x 3 aelarabawy aelarabawy 4096 Nov 21 16:25 Perforce
2621455 drwxr-xr-x 3 aelarabawy aelarabawy 4096 Apr 29 14:27 Pictures
2621452 drwxr-xr-x 2 aelarabawy aelarabawy 4096 Nov 21 14:52 Public
2759157 drwxrwxr-x 5 aelarabawy aelarabawy 4096 Dec 10 10:16 sketchbook
2621634 drwxrwxr-x 3 aelarabawy aelarabawy 4096 Nov 21 14:56 software
2621451 drwxr-xr-x 2 aelarabawy aelarabawy 4096 Nov 21 14:52 Templates
2621456 drwxr-xr-x 2 aelarabawy aelarabawy 4096 Nov 21 14:52 Videos
2759177 drwxrwxr-x 5 aelarabawy aelarabawy 4096 Apr 22 17:44 work
aelarabawy@aelarabawy-demo-backup64:~$
```

Showing File Status (stat Command)



\$ stat (Show File Status info)

```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ stat log
  File: `log'
  Size: 42289          Blocks: 88          IO Block: 4096   regular file
Device: 803h/2051d      Inode: 2632359      Links: 1
Access: (0664/-rw-rw-r--) Uid: ( 1001/aelarabawy)  Gid: ( 1001/aelarabawy)
Access: 2014-04-21 11:49:41.372354617 -0700
Modify: 2014-04-11 10:32:05.224869809 -0700
Change: 2014-04-11 10:32:05.224869809 -0700
 Birth: -
aelarabawy@aelarabawy-demo-backup64:~$
```

Show FileSystem Disk Space Usage (df Command)



\$ df (Show FileSystem Disk Space Usage)

```
aelarabawy@aelarabawy-demo-backup64:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda2        98430596  15735912   77694700  17% /
udev             4070772       4   4070768   1% /dev
tmpfs            1632048     924   1631124   1% /run
none              5120       0     5120   0% /run/lock
none              4080116     464   4079652   1% /run/shm
cgroup            4080116       0   4080116   0% /sys/fs/cgroup
/dev/sda3        757071712  99267280  619347344 14% /home
/dev/sda1        96318212  5078604   86346888   6% /media/ec95cd0b-aebf-4161-8fb4-
b24d1384a905
aelarabawy@aelarabawy-demo-backup64:~$
```

Show FileSystem Disk Space Usage (df Command)



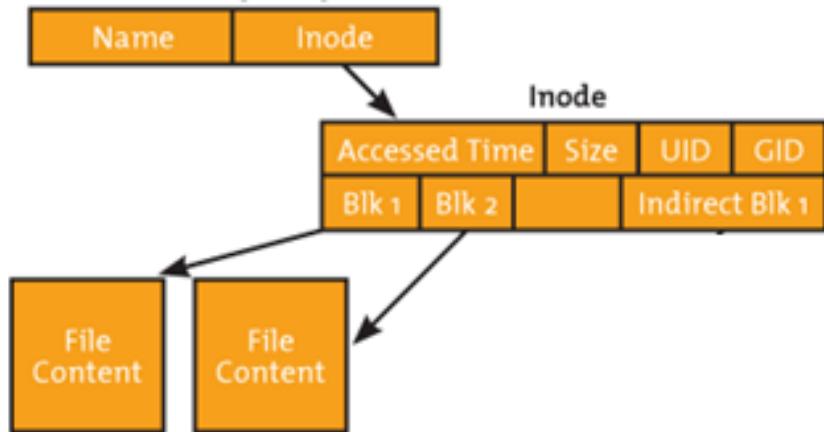
\$ df -i (Show FileSystem inode Usage)

```
aelarabawy@aelarabawy-demo-backup64:~$ df -i
Filesystem      Inodes   IUsed   IFree  IUse% Mounted on
/dev/sda2        6250496  363722  5886774    6% /
udev            1017693    518  1017175    1% /dev
tmpfs           1020029    464  1019565    1% /run
none            1020029      5  1020024    1% /run/lock
none            1020029     51  1019978    1% /run/shm
cgroup          1020029      9  1020020    1% /sys/fs/cgroup
/dev/sda3        48078848  470652  47608196   1% /home
/dev/sda1        6119424  256011  5863413    5% /media/ec95cd0b-aebf-4161-8fb4-b24
d1384a905
aelarabawy@aelarabawy-demo-backup64:~$ █
```

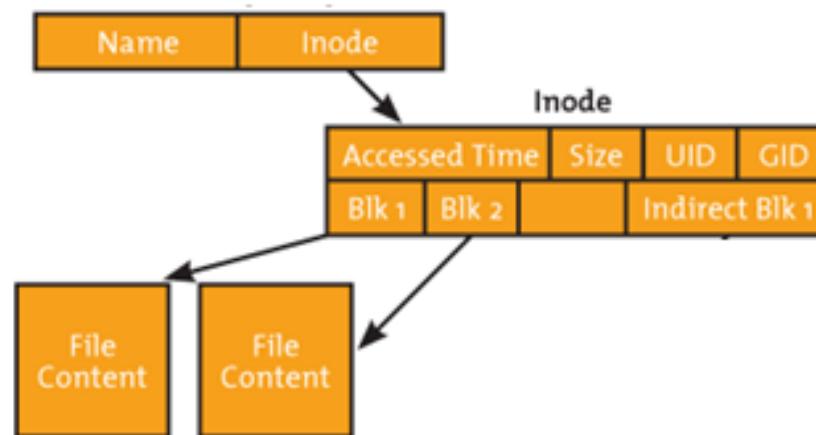
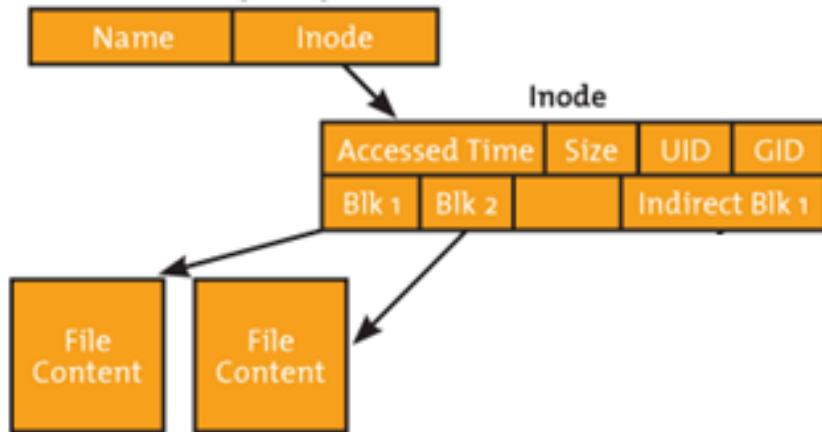


FILE OPERATIONS

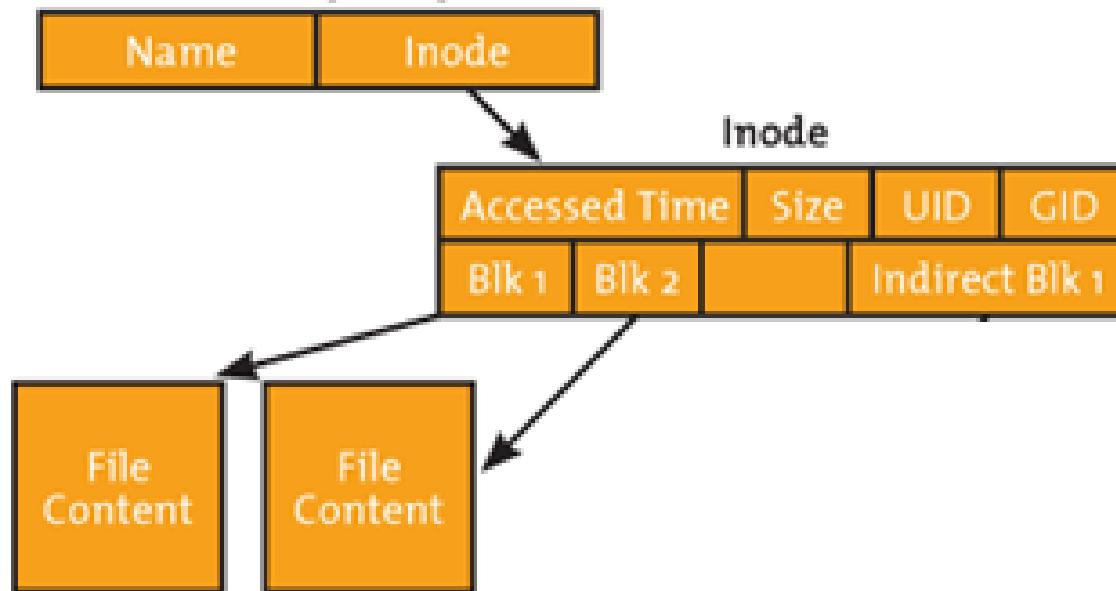
Copying a File



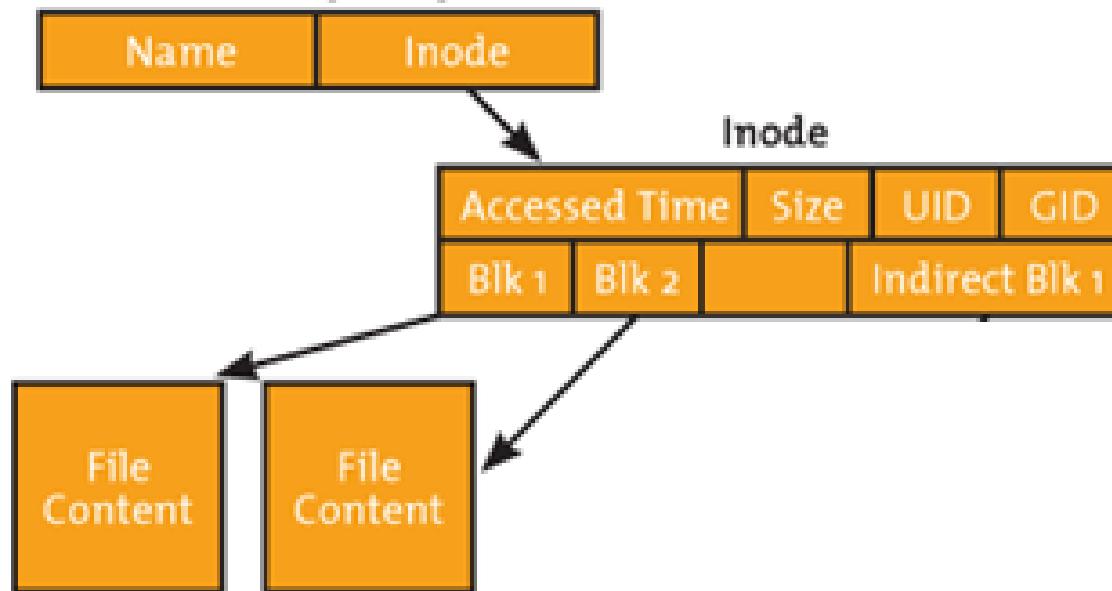
Copying a File



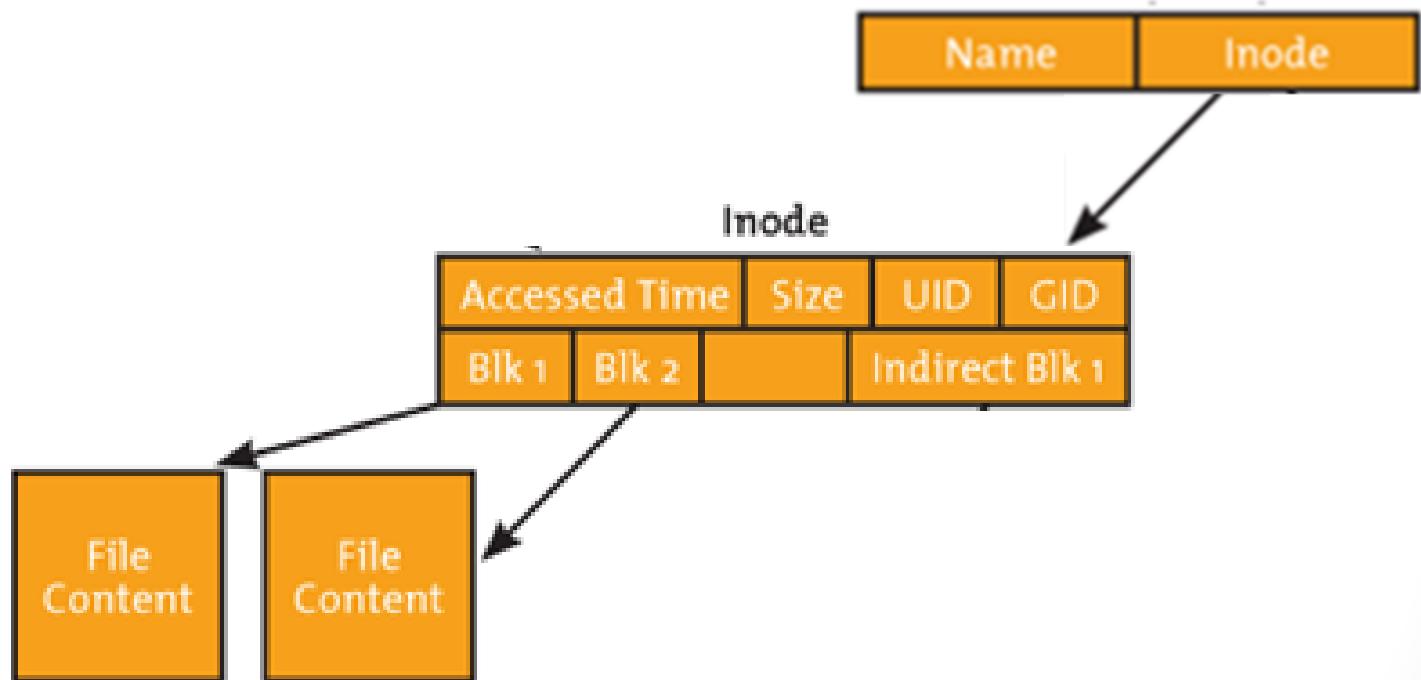
Renaming a File



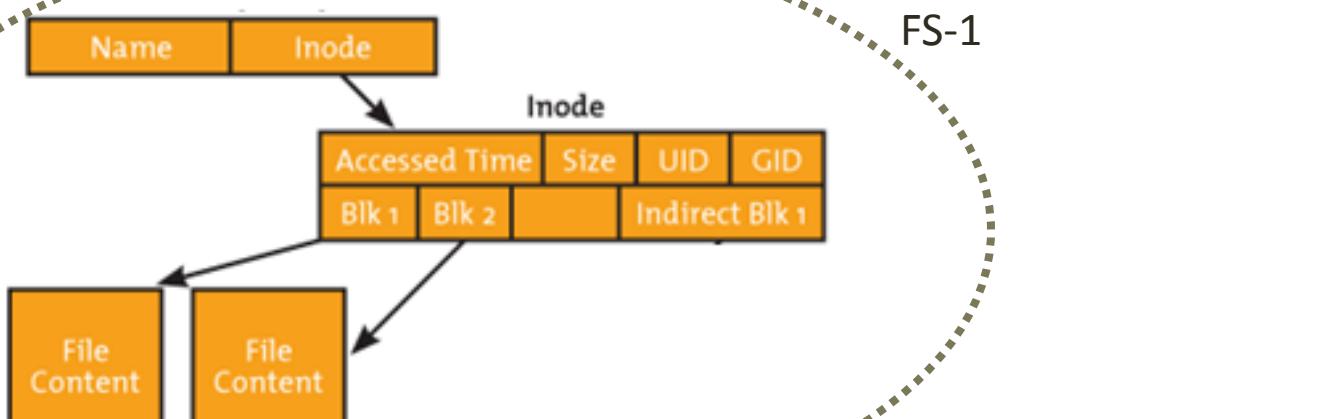
Moving a File (Within Same FS)



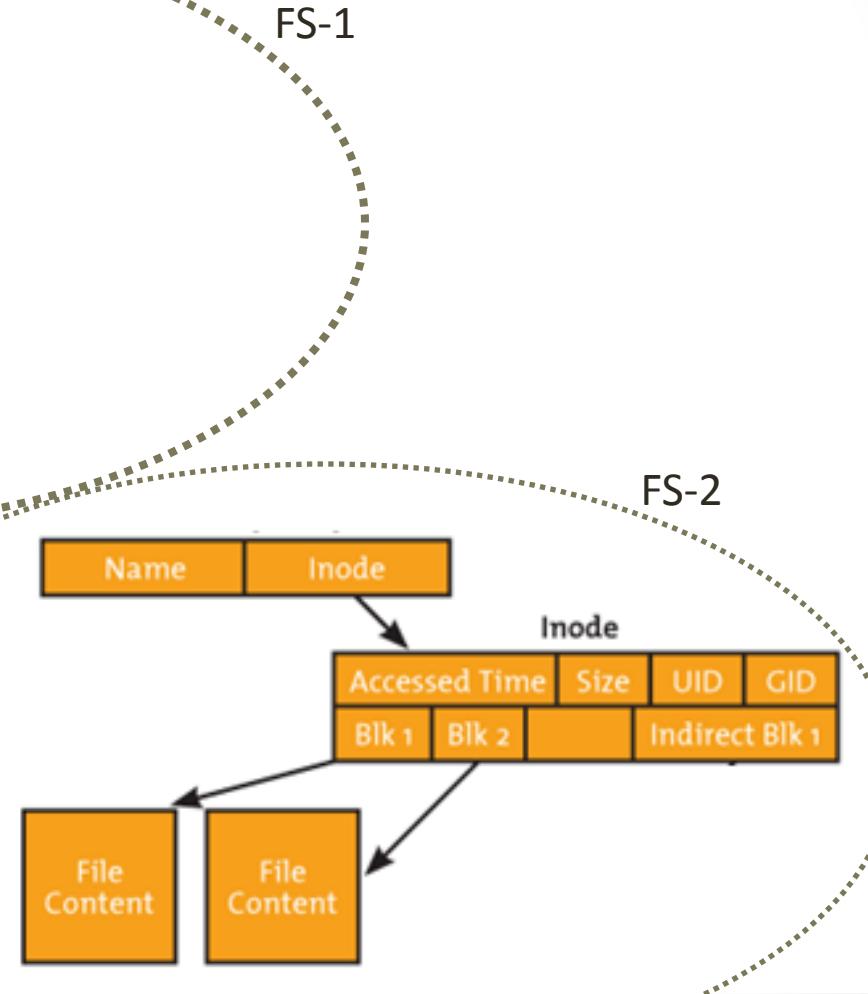
Moving a File (Within Same FS)



Moving a File (Between FSs)

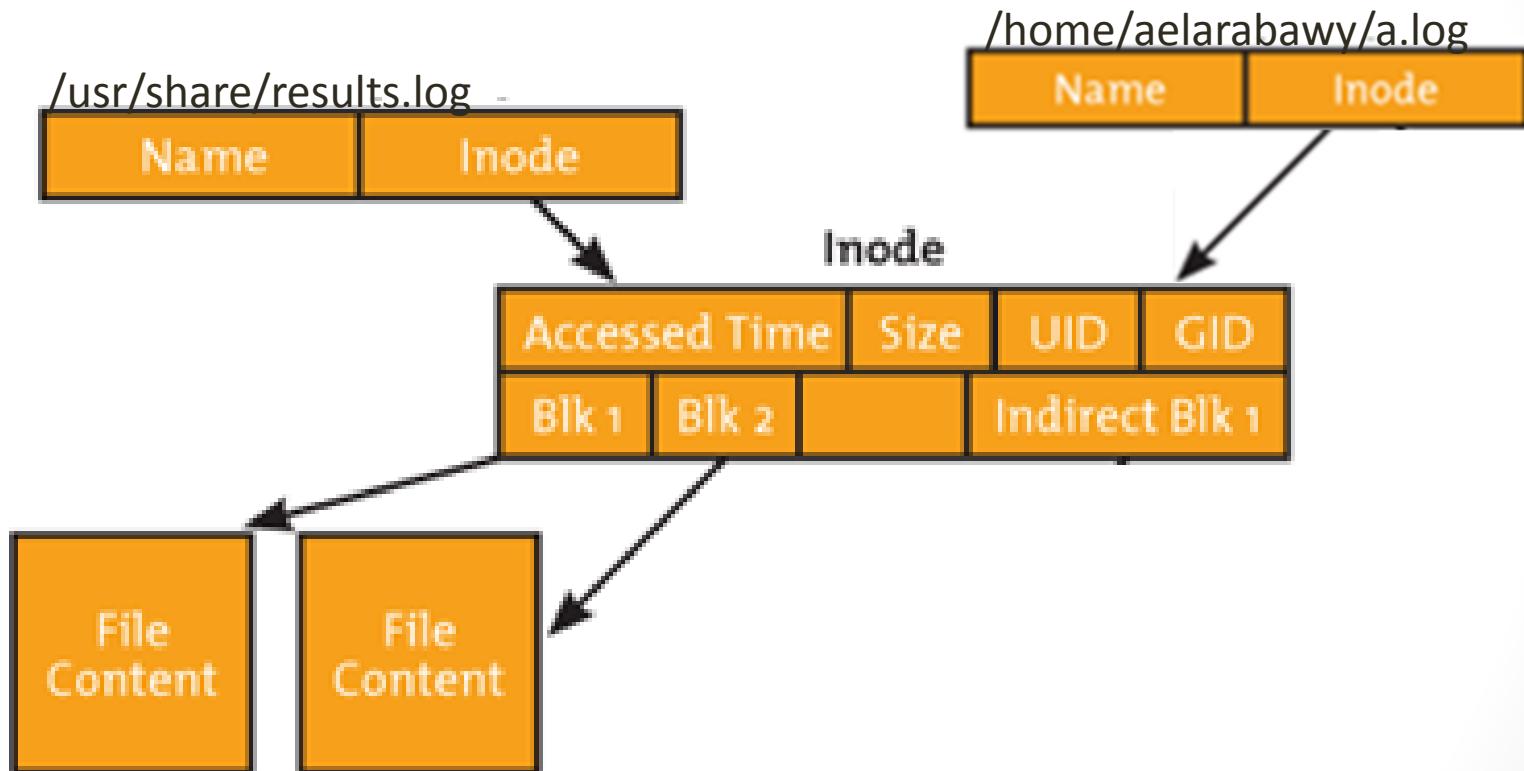


Moving a File (Between FSs)





LINKING FILES



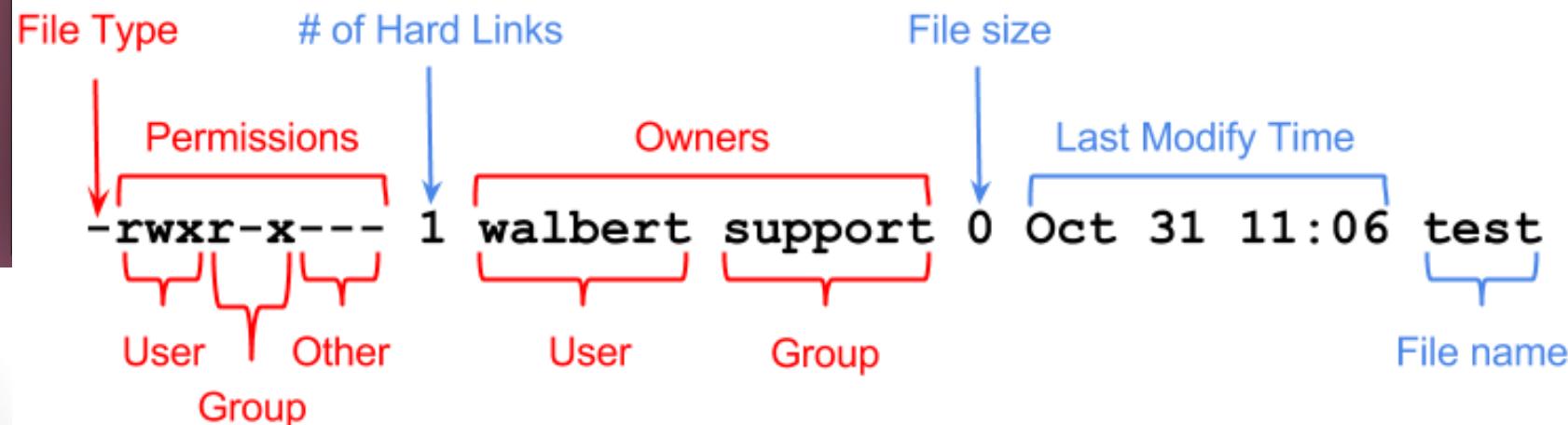
Hard Link



Hard Links

- The decision of not including the filename and path in the “inode” structure was to enable the use of hard links
- Hard links were introduced from the early days of Unix
- A hard link
 - Not a new file
 - Same file content
 - Same inode
 - Just an additional “dentry” with a different filename/path, but with the same inode#
- This is useful if we need to have the same file with two names, or in two locations

```
robb@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 robb robb 4096 2011-10-04 22:25 Desktop
drwxr-xr-x 3 robb robb 4096 2011-05-14 15:57 Documents
drwxr-xr-x 3 robb robb 4096 2011-10-03 19:34 Downloads
-rw-r--r-- 1 robb robb 179 2011-05-12 09:11 examples.desktop
drwxr-xr-x 4 robb robb 4096 2011-09-29 23:37 hw02
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Music
drwxr-xr-x 3 robb robb 4096 2011-10-04 22:23 Pictures
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Public
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Templates
drwxr-xr-x 2 robb robb 4096 2011-05-12 09:15 Videos
robb@ubuntu:~$
```

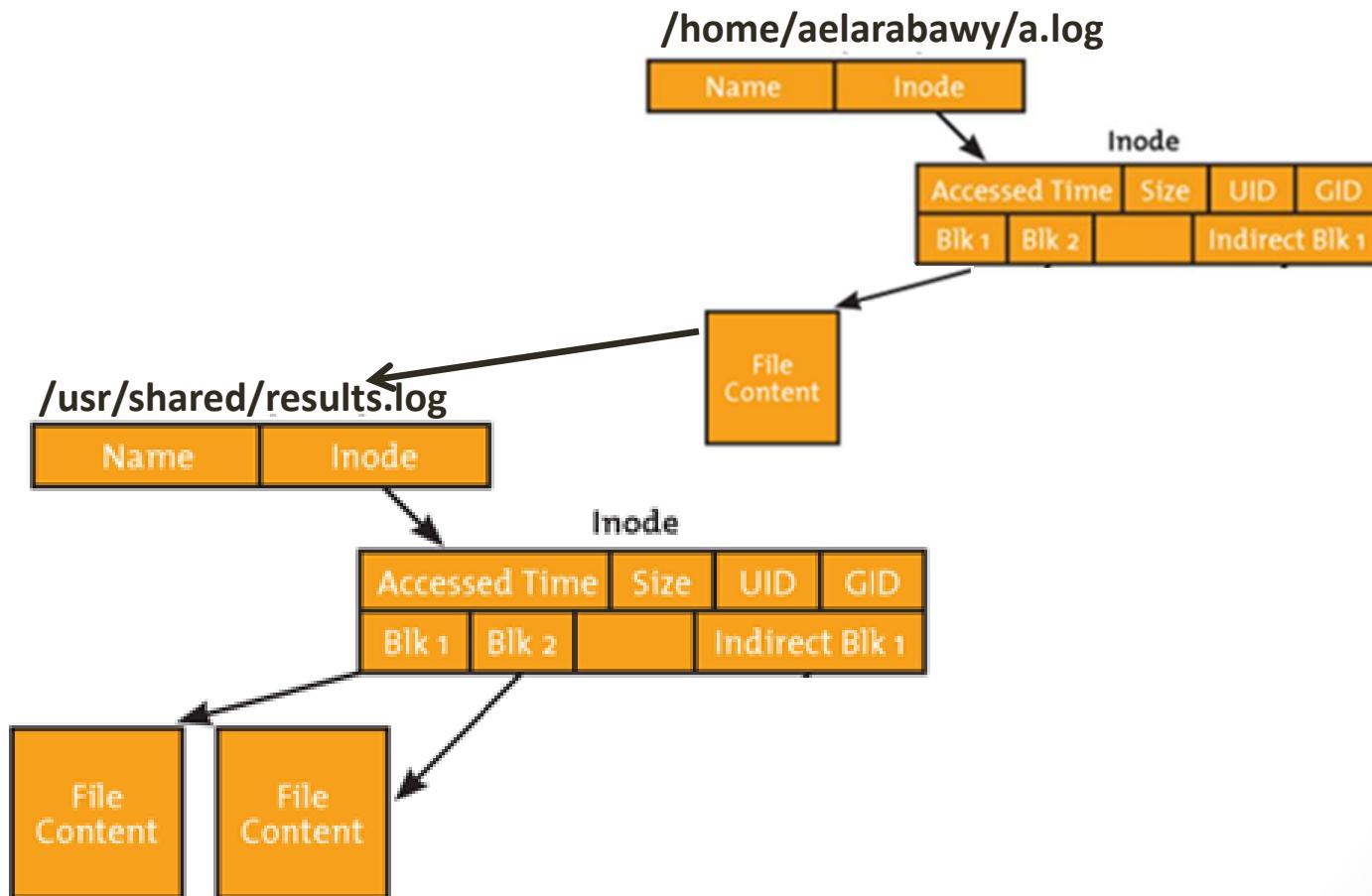




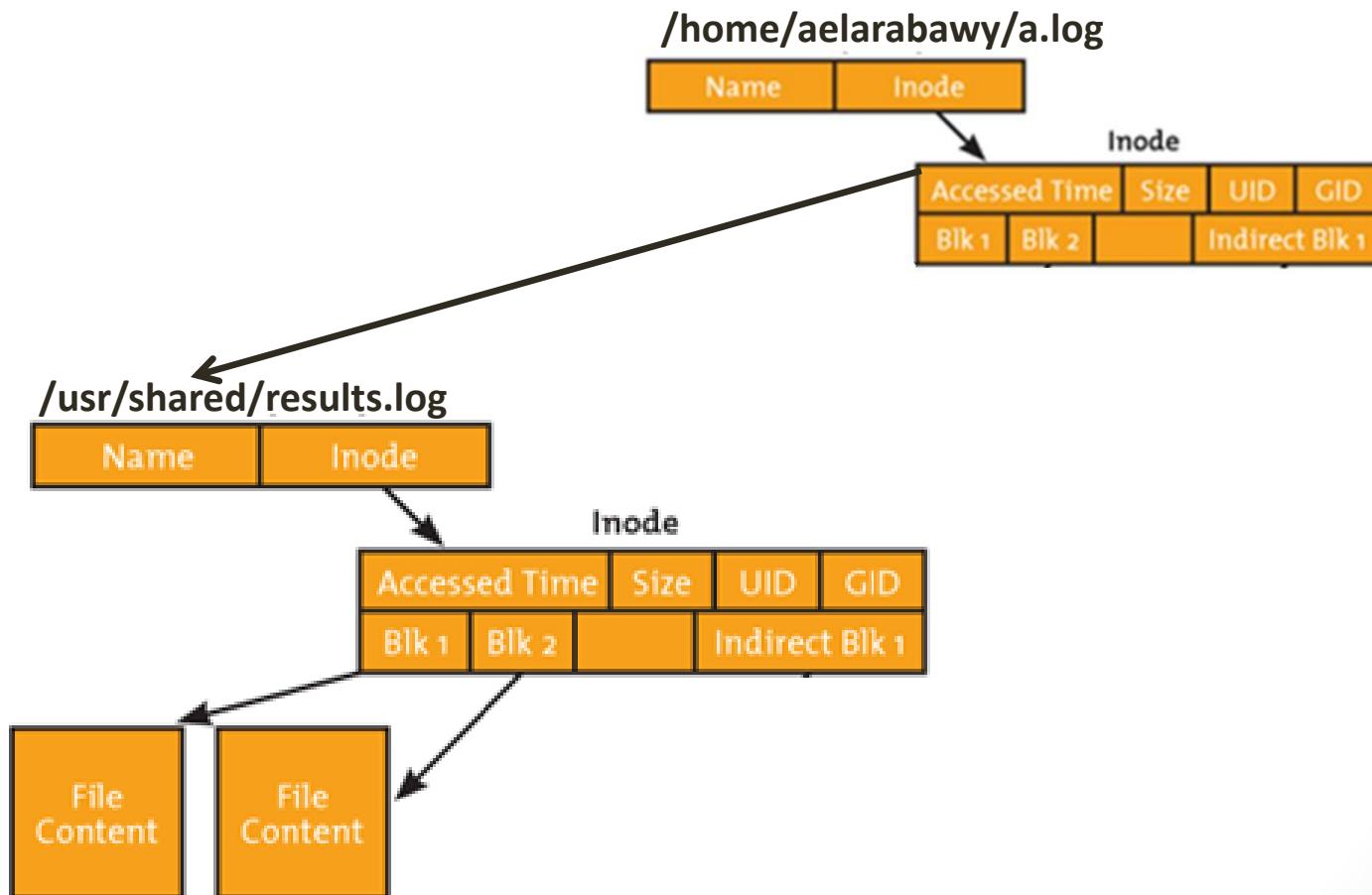
However,

- Hard links are not very common these days, they have some limitations,
 - Only applicable for files, not used for directories
 - After implementing it for directories, a security hole was found
 - Can cause loops of links which result in system faults
 - So it was disabled in latest releases
 - Does not work across filesystems
 - We just link using the inode#
 - But inode# is only unique within the same filesystem
 - Hence, we can not link to a file in a different filesystem
 - This is very limiting, specially Linux merges all the FS in a unified tree

Symbolic Links



Symbolic Links



Symbolic Links

- A symbolic link is introduced to fix the problems of Hard Links
- A symbolic Link is not just a dentry structure; it is a file with an inode structre
- The inode structure
 - The type is set to 'l' for a symbolic link
- Two types of Implementation:
 - Slow Symbolic Links:
 - The data block of the new file include the path of the file it is linking to
 - Fast Symbolic Links:
 - A field in the inode points to the path and name of the file/directory it is pointing to
 - Faster, no need to read the data block
 - Not possible if the path is too long to fit in the inode structure
- Since a symbolic link it has its own inode, with an obvious indication that it is a link,
 - Some commands is able to treat it differently
 - Avoid the security hole in hard links with linking directories
 - We can link to a file/directory in a different file system

Symbolic Links

- Symbolic links are like shortcuts in windows
- You can have a symbolic link to a file or a folder

```
andrew@D630:~/labs$ ls -l
total 8
lrwxrwxrwx 1 andrew andrew 14 Sep 7 11:21 doc -> /usr/share/doc
-rw-rw-r-- 1 andrew andrew 0 Sep 7 10:48 file1
-rwxrwxr-x 1 andrew andrew 0 Sep 7 10:48 file.sh
prw-rw-r-- 1 andrew andrew 0 Sep 7 11:05 pipe
drwxrwxr-x 3 andrew andrew 4096 Mar 1 2013 tools
drwxrwxr-x 3 andrew andrew 4096 Jan 25 2013 usp
andrew@D630:~/labs$ 
```

Symbolic Links

- Symbolic links are like shortcuts in windows
- You can have a symbolic link to a file or a folder

```
andrew@D630:~/labs$ ls -l
total 8
lrwxrwxrwx 1 andrew andrew 14 Sep  7 11:21 doc  -> /usr/share/doc
-rw-rw-r-- 1 andrew andrew  0 Sep  7 10:48 file1
-rwxrwxr-x 1 andrew andrew  0 Sep  7 10:48 file.sh
prw-rw-r-- 1 andrew andrew  0 Sep  7 11:05 pipe
drwxrwxr-x 3 andrew andrew 4096 Mar  1  2013 tools
drwxrwxr-x 3 andrew andrew 4096 Jan 25  2013 usp
andrew@D630:~/labs$
```

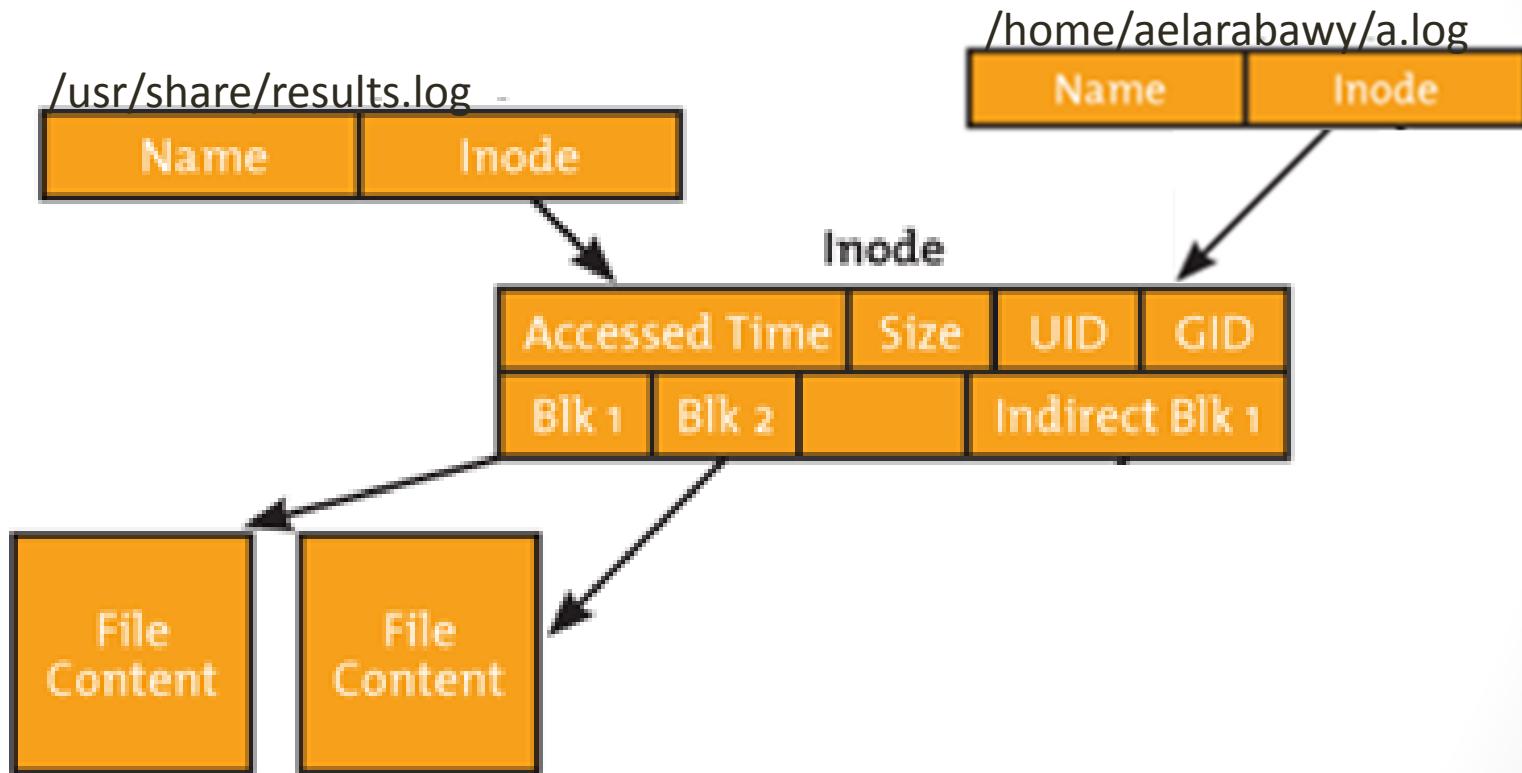
Question:

Why do you think the size of “doc” is 14 bytes ???

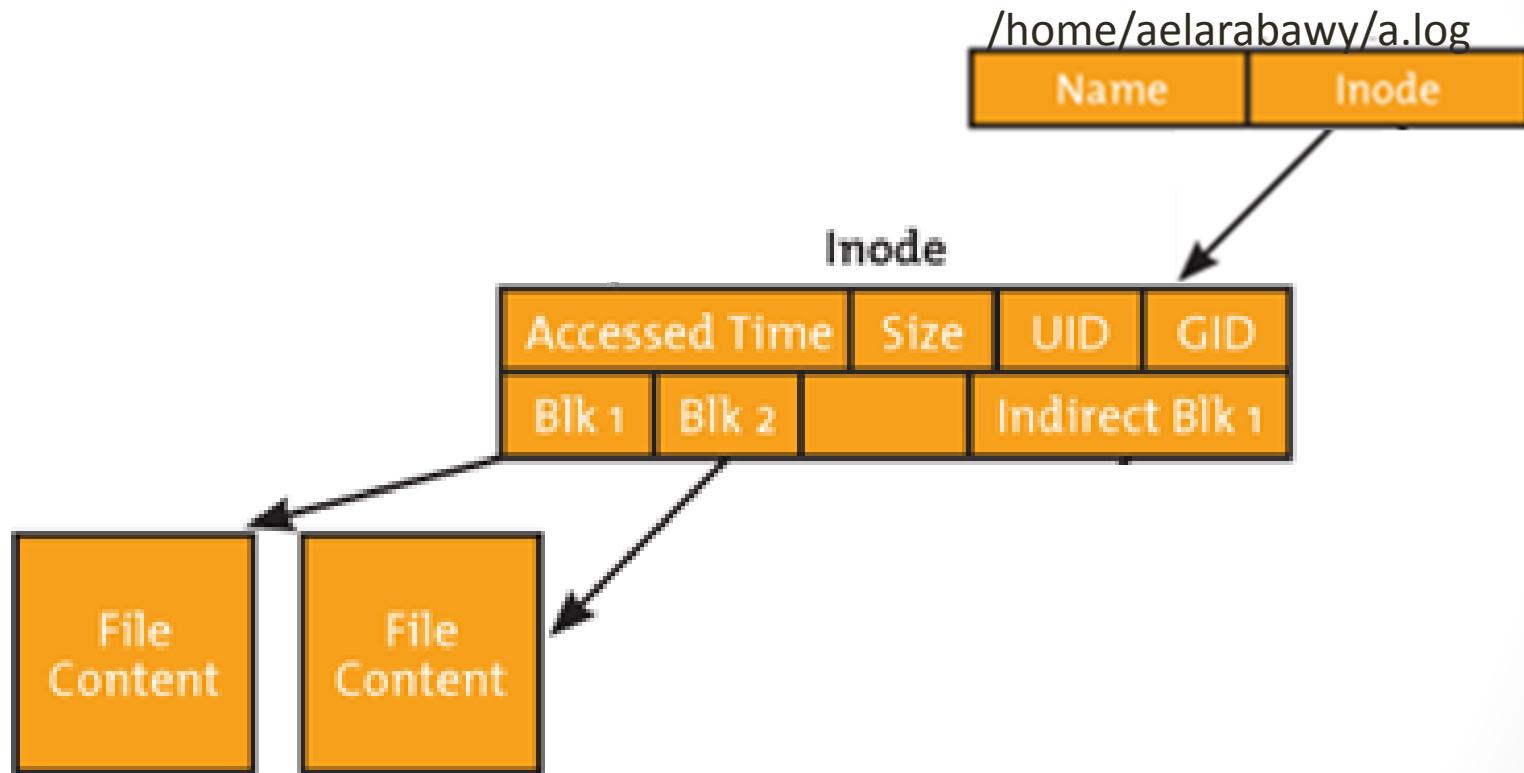


DELETING FILES & LINKS

Hard Links

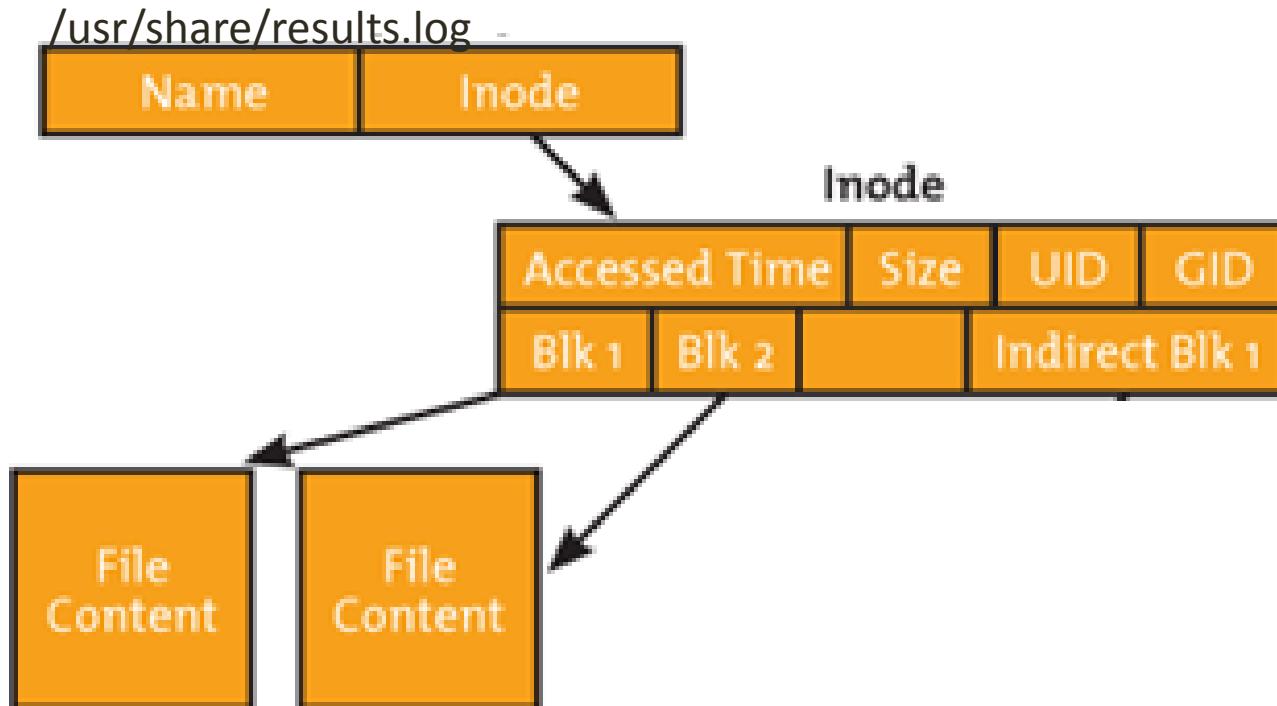


Hard Links



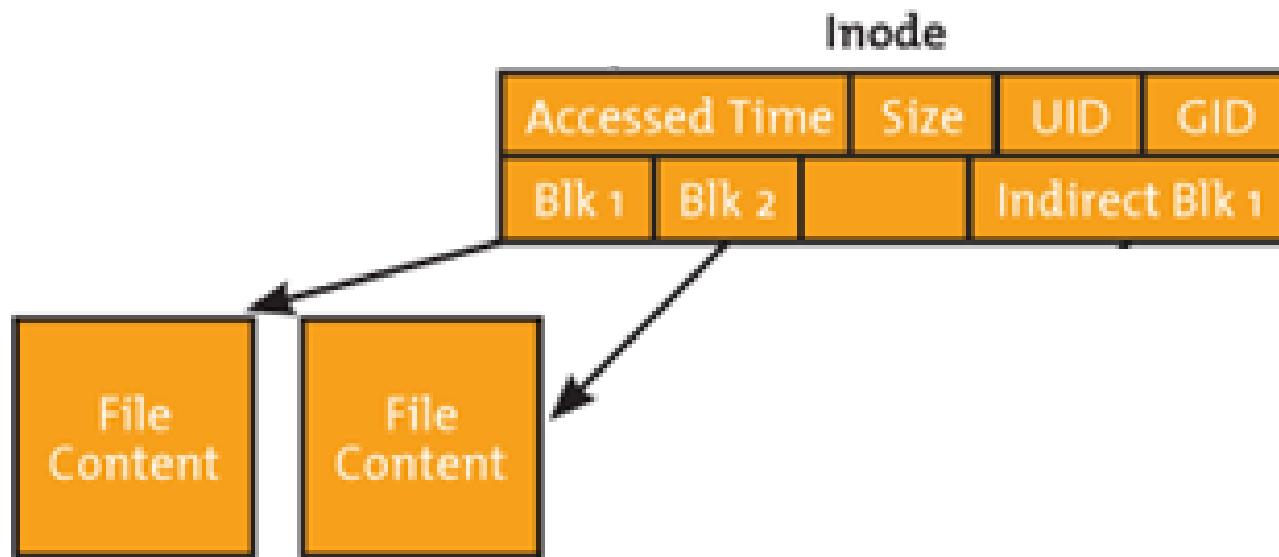
`$ rm /usr/share/results.log`

Hard Links



\$ rm ~/a.log

Hard Links



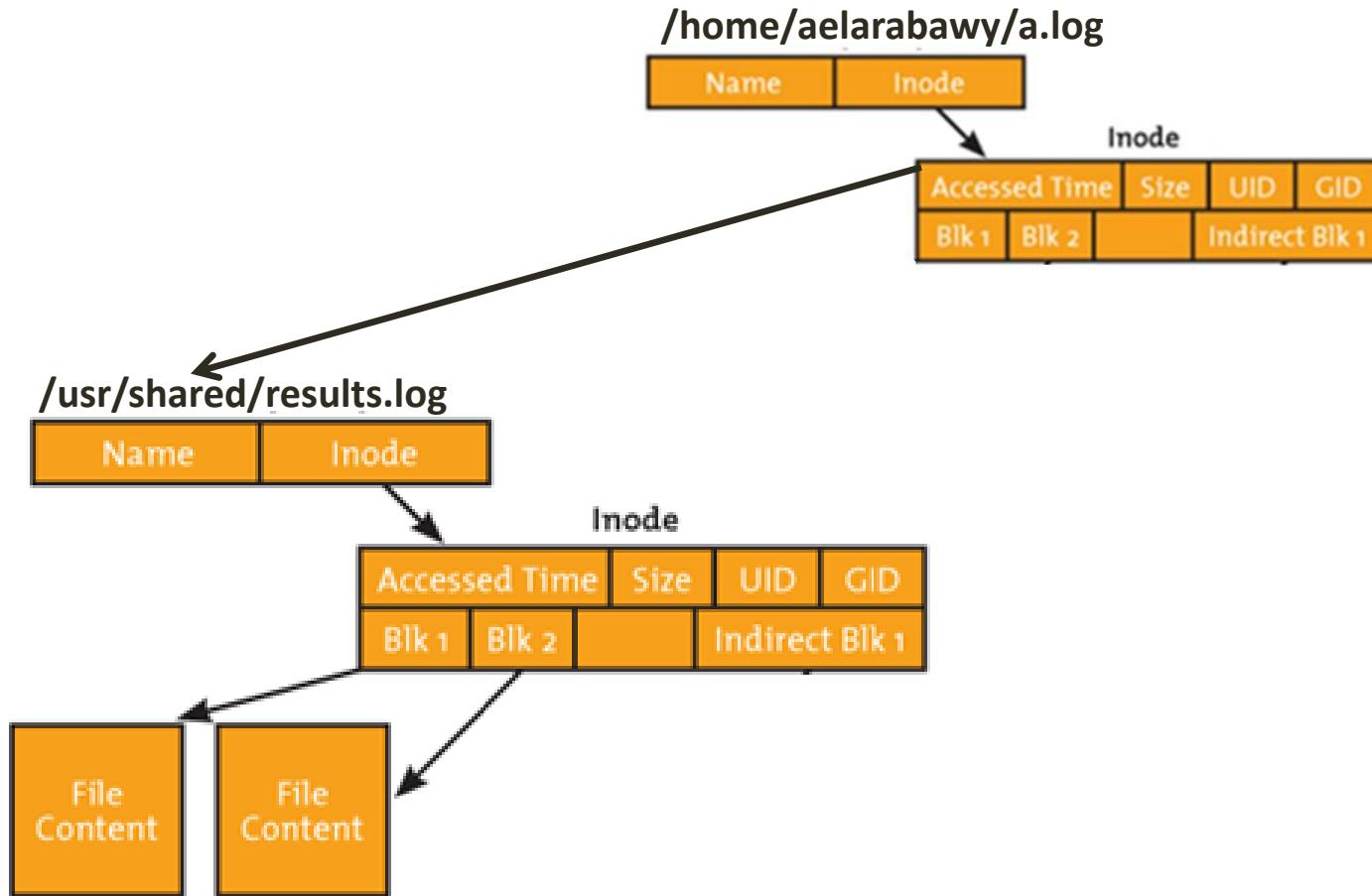
```
$ rm /usr/share/results.log  
$ rm ~/a.log
```

Hard Links

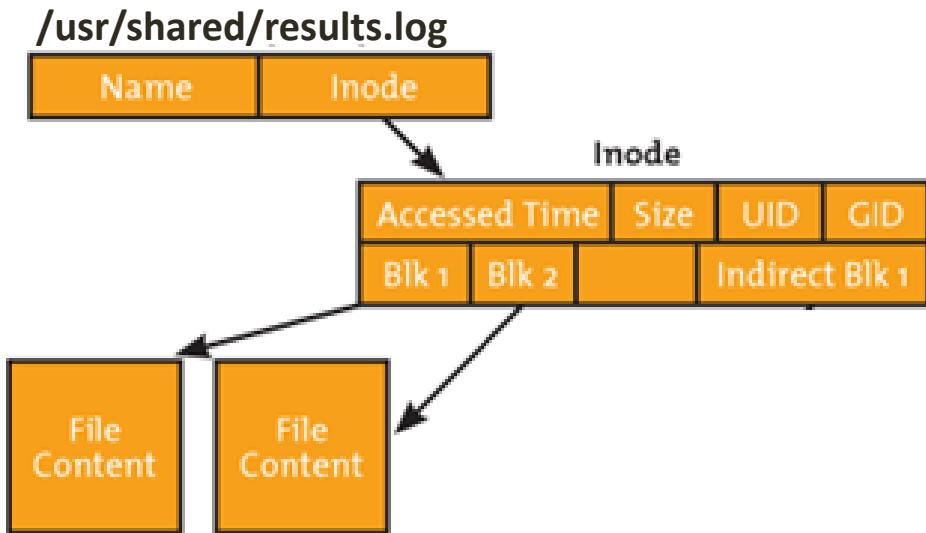


```
$ rm /usr/share/results.log  
$ rm ~/a.log
```

Symbolic Links

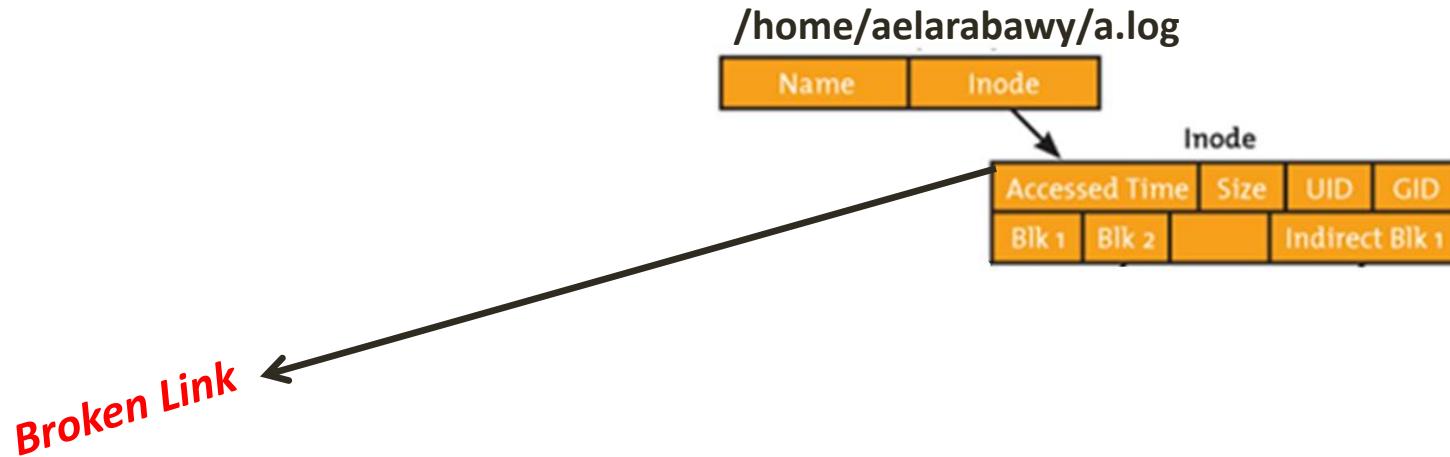


Symbolic Links



\$ rm ~a.log

Symbolic Links



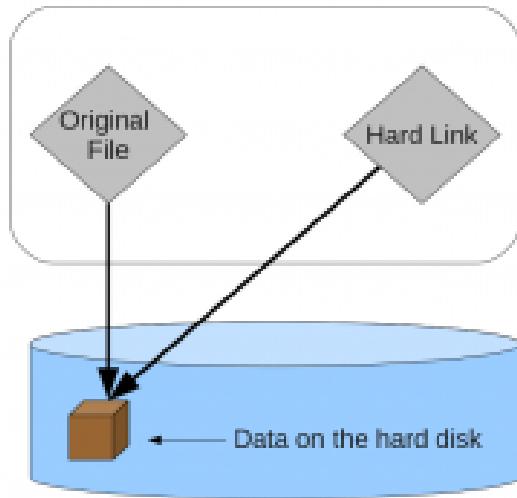
\$ rm /usr/shared/results.log

Hard Link Vs Symbolic Link

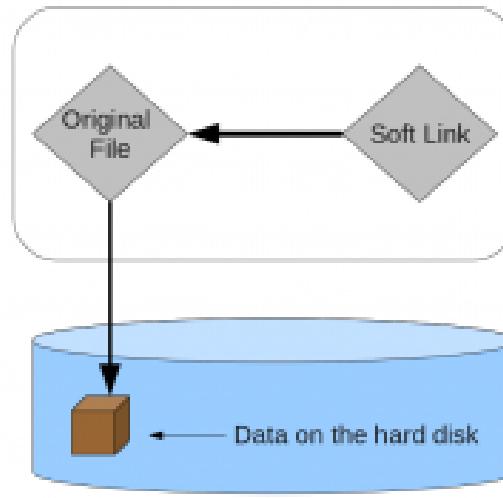
Abstracted View



Hard Links



Symbolic Links



Question: Which type of link is more space efficient ??



LINUX COMMANDS

Creating File Links (ln Command)



- To create a Hard Link

\$ ln <File to link to> <link name & location>

\$ ln file.log ~log-files/a.log

- To create a Symbolic Link

\$ ln -s <File to link to> <link name & location>

\$ ln -s ~/file.log ~log-files/a.log

Important Note:

Always use absolute paths for the file to link to when creating symbolic links Never use relative path format

Question....

We agreed that Hard links are not allowed for directories...

Now doing a simple listing results in:



```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ls -il
total 112
2884089 drwxrwxr-x 5 ael
2621449 drwxr-xr-x 2 ael
3939335 drwxrwxr-x 2 ael
2621453 drwxr-xr-x 2 ael
2621450 drwxr-xr-x 4 ael
2621445 -rw-r--r-- 1 ael
2632359 -rw-rw-r-- 1 ael
2621454 drwxr-xr-x 2 ael
2621726 drwxrwxr-x 3 ael
2621455 drwxr-xr-x 3 ael
2621452 drwxr-xr-x 2 ael
2759157 drwxrwxr-x 5 ael
2621634 drwxrwxr-x 3 ael
2621451 drwxr-xr-x 2 ael
2621456 drwxr-xr-x 2 ael
2759177 drwxrwxr-x 5 ael
aelarabawy@aelarabawy-demo-backup64:~$
```



Linux 4 Embedded Systems

<http://Linux4EmbeddedSystems.com>