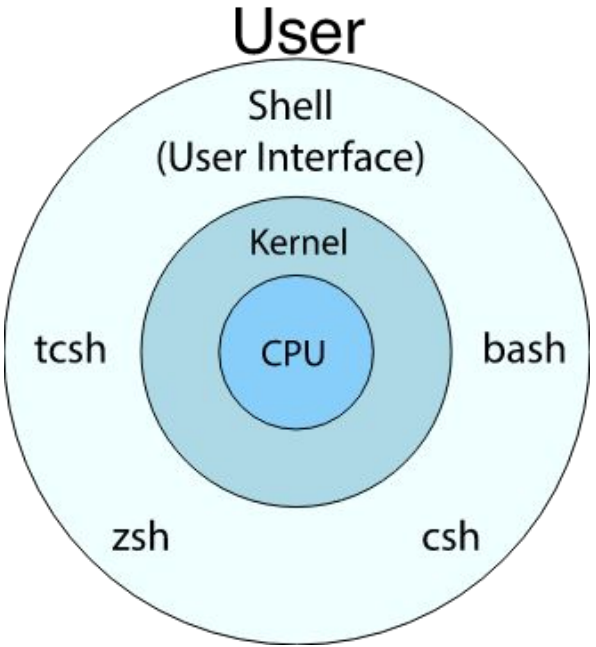
Linux Basics
Lesson 1

Why Linux

- Linux is a powerful kernel.
  - Many web sites use Linux as the operating system.
  - Even Steve Ballmer of Microsoft said Linux has 60% of the server market in 2008.
  - Tolerant of a range of hardware platforms without special configuration.
- Computer Forensics need to be able to consider server forensics.
  - Forensic issues can happen on server platforms too.
- Host-Based forensic tools often run on linux platforms.
  - Free platform
  - Flexible and reliable
  - Easier to access low-level interfaces
  - Good forensic qualities.
  - Will consider Caine (a Linux live cd) for host-based forensics, which runs The Forensic Toolkit and Autopsy.

Linux Shell

- *Shell* interprets the command and Bash, Tcsh, Zsh request service from kernel
- Similar to DOS but DOS has only one set of interface while Linux can select Kernel different shell
    - Bourne Again shell (Bash), TC shell (Tcsh), Z shell (Zsh)
- Different shell has similar but different functionality
- *Bash* is the default for Linux
- Graphical user interface of Linux is in fact an application program work on the shell



GRUB

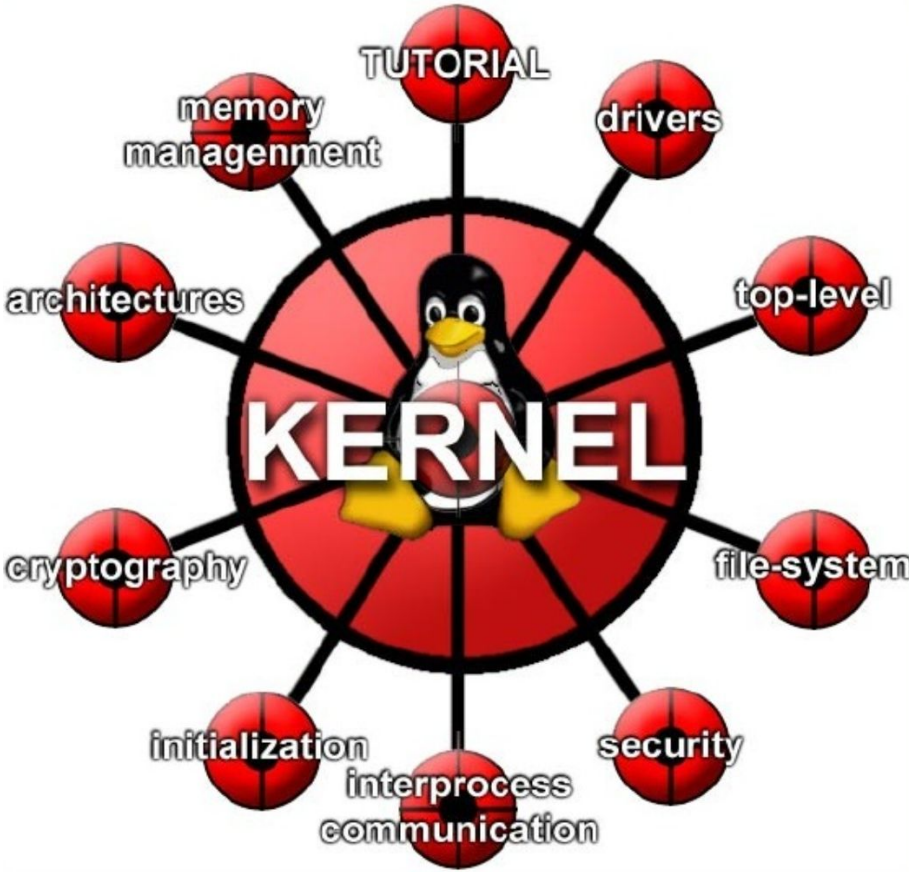
- GNU GRUB (short for GNU GRand Unified Bootloader) is a *boot loader* package from the *GNU Project*.
- GRUB is the *reference implementation* of the *Multiboot Specification*, which provides a user the choice to boot one of multiple *operating systems* installed on a computer or select a specific *kernel* configuration available on a particular operating systems partitions.

KERNEL

- The kernel(nucleus) is the essential part of a computer operating system, the core that provides basic services for all other parts of the operating system.
- A kernel can be contrasted with a shell, the outermost part of an operating system that interacts with user commands.
- Kernel and shell are terms used more frequently in Unix operating systems than in IBM mainframe or Microsoft Windows systems.

KERNEL

- Typically, a kernel (or any comparable center of an operating system) includes an interrupt handler that handles all requests or completed I/O operations that compete for the kernels services, a scheduler that determines which programs share the kernels processing time in what order, and a supervisor that actually gives use of the computer to each process when it is scheduled.
- A kernel may also include a manager of the operating systems address spaces in memory or storage, sharing these among all components and other users of the kernels services. A kernels services are requested by other parts of the operating system or by application programs through a specified set of program interfaces sometimes known as system calls.


TUTORIAL
memory managenment
drivers
architectures
top-level
KERNEL
cryptography
file-system
initialization
security
interprocess communication

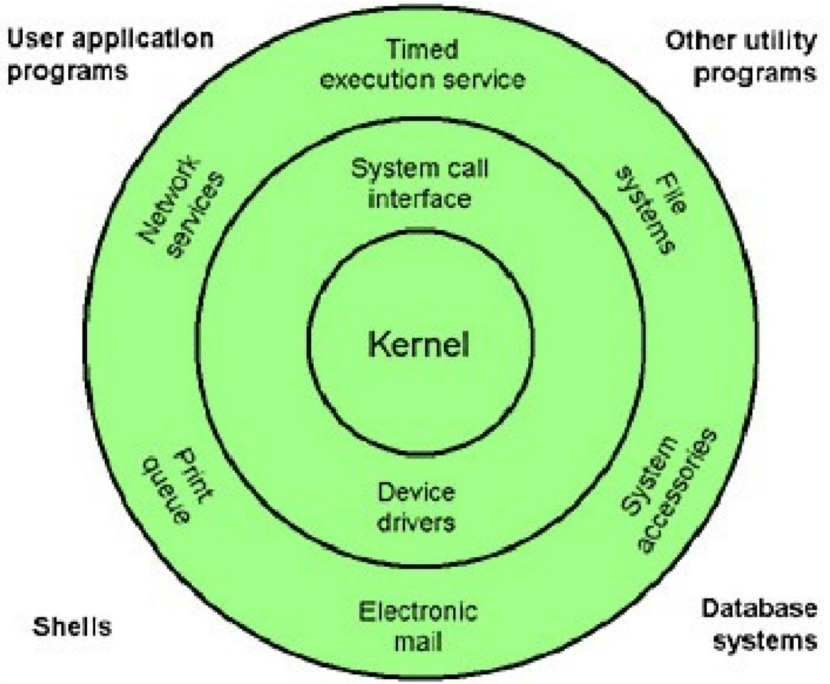
Kernel version

- **UBUNTU 11.0 running**
- What is kernel version?

Kernel version

- UBUNTU 11.0 –say company name
- TYPE umane –r to find the kernel .
- UBUNTU 11.0 has 2.6.38 kernel version.
- We can also kind kernel when we start a linux machine. (GRUB)

Kernel


User application programs
Other utility programs
Timed execution service
System call interface
Network services
File systems
Kernel
Print queue
Device drivers
System accessories
Shells
Electronic mail
Database systems

KDE and GNOME

- KDE and GNOME are two desktop environments (collection of software that provides certain functionality and a look and feel for operating systems) that run on operating systems that use X Window System (mostly Unix, Linux, Solaris, FreeBSD, and Mac OS X).
- KDE's main programming language is C++. The main reason for this is that the main functionality of KDE is coded using QT, which is written in C+ +. It takes approximately 210MBs to install the base system of KDE.
- GNOME's main *programming* language is C, because the tool kit used to write GNOME is GTK+ and it is written in C. Approximately 180 MB is required to install the base system of GNOME.

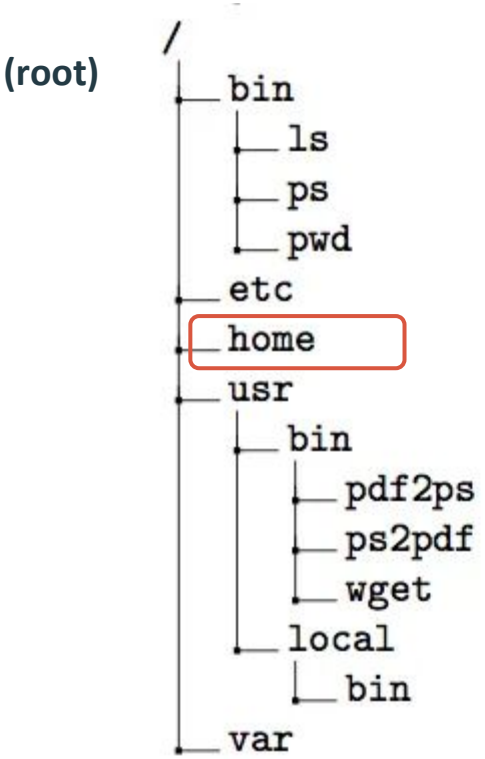
  After the recent rebranding,"KDE" actually refers to the whole collection of applications including the desktop environment while GNOME refers to a desktop environment alone.

File management

Directory Tree

Directory Tree

When you log on the the Linux OS using your username you are automatically located in your home directory.



- **/bin** System binaries, including the command shell
- **/boot** Boot-up routines
- **/dev** Device files for all your peripherals
- **/etc** System configuration files
- **/home** User directories
- **/lib** Shared libraries and modules
- **/lost+found** Lost-cluster files, recovered from a disk-check
- **/mnt** Mounted file-systems
- **/opt** Optional software
- **/proc** Kernel-processes pseudo file-system
- **/root** Administrator’s home directory
- **/sbin** System administration binaries
- **/usr** User-oriented software
- **/var** Various other files: mail, spooling and logging

Directories

- /bin : This contains commands a user can run, like ‘ls’, but which might be needed during boot.
- /dev : This contains devices, like the mouse.
- /home : This is where users store their files.
- /tmp : Temporary storage for users and the system
- /var : System files which can change.
- /etc : System config files which don’t change
- /lib : Where all the system libraries live
- /proc : Files which represent the running system (like processes).
- /sbin : Commands which only an administrator would want.
- /usr : Commands which are never needed during bootup.

Most important subdirectories inside the root directory

- /bin : Important Linux commands available to the average user.
- /boot : The files necessary for the system to boot. Not all Linux distributions use this one. Fedora does.
- /dev : All device drivers. Device drivers are the files that your Linux system uses to talk to your hardware. For example, theres a file in the /dev directory for your particular make and model of monitor, and all of your Linux computers communications with the monitor go through that file.
- /etc : System configuration files.
- /home : Every user except root gets her own folder in here, named for her login account. So, the user who logs in with linda has the directory /home/linda, where all of her personal files are kept.
- /lib : System libraries. Libraries are just bunches of programming code that the programs on your system use to get things done.

Most important subdirectories inside the root directory

- /mnt : Mount points. When you temporarily load the contents of a CD-ROM or USB drive, you typically use a special name under /mnt. For example, many distributions (including Fedora) come, by default, with the directory /mnt/cdrom, which is where your CD-ROM drives contents are made accessible.
- /root : The root users home directory.
- /sbin : Essential commands that are only for the system administrator.
- /tmp : Temporary files and storage space. Dont put anything in here that you want to keep. Most Linux distributions (including Fedora) are set up to delete any file thats been in this directory longer than three days.
- /usr : Programs and data that can be shared across many systems and dont need to be changed.
- /var : Data that changes constantly (log files that contain information about whats happening on your system, data on its way to the printer, and so on).

Most important subdirectories inside the root directory

- /proc : virtual file sys(imp for forensics).
- /etc/passwd: contains all user specific info.
- /etc/shadow: stores password.

Home directory

- You can see what your home directory is called by entering
- *pwd* (print current working directory)

Useful commands

Command	Description
ls	List dir and files
dir	See directory
Ifconfig	See IP address config
Useradd	Add new user
Reboot	Reboot machine
Su	Switch user
Wget	To download any file for internet
Top	See process activity
Ps	Display processes

Useful commands

Command	Description
Iptraf	Eal time network statistics
Tcpdump	Detailed network traffic analysis
Cat	Displays contents of the file
vim	Opens text editor
Mount	Mounts dir
Cd	Change dir
Mkdir	Make dir
Rmdir	Remove dir
Pwdshow	Show present working dir

The PROMPT

- Once you log into your machine, you are at the prompt. Here you can perform your commands.
- Everything on linux is either a file or a directory.
- A file which is executed becomes a process.
- Processes can be seen as files too.
- Devices, such as scanners and hard drives are also files.

File System

- Windows uses letters of the alphabet to represent different devices and different hard disk partitions. Under Windows, you need to know what volume (C:, D:,...) a file resides on to select it, the files physical location is part of its name.
- In Linux all directories are attached to the root directory, which is identified by a forward- slash, "/". - root.

File System

For example, below are some second-level directories:

- # - shell command.
- # fdisk -l /*list partitions*/
- /dev/sda1
  - /dev – device
  - /sda1 or /hda1
    - sd – SATA /*SATA- tech to read/write data*/
    - hd – IDE
    - a
    - Sd ..a/b/c/d.....1/2/3.....
      - a – primary master
      - b – primary slave • c – secondary master • d – secondary slave • 1/2/3... – first/second/third partition

File System

- # fdisk /dev/sda /*’m’ for help*/
- (if type ‘l’ it will list all available file sys with their Id e.g. Windows -7 and Linux -83)
- (‘q’ to quit)
- /dev/sda1 - sys reserve
- /dev/sda2 - is C:
- (windows makes 2 partitions: 100 Mb(from 100 GB) for sys reserve and remaining C: (100GB))

Redirection

- If you end a command with “>”, its output goes to a file.
- If you end a command with “<“, its input comes from a file.

Linux Help

- man
- info
- command –help
- Forums

Man -k

- You can keyword search for commands
- For instance, what commands show a calendar?

$ man -k calendarcal

cal	(1) - displays a calendar
cal	(1p) - print a calendar
difftime	(3p) - compute the difference…

Man command

- The “-k” option
  - man –k print
- Manual pages are divided in 8 sections:
  - User commands
  - System calls
  - Libc calls
  - Devices
  - File formats and protocols
  - Games
  - Conventions, macro packages and so forth
  - System administration
- To select correct section, add section number:
  - man 1 passwd, man 5 passwd

Info command

- A program for reading documentation, sometimes a replacement for manual pages
- Example : info ls

Linux Boot Sequence

Run levels

- They initiate sys call which interact with kernel.
- Run level 0 : shutdown.
- Run level 1 : single user interface.
- Run level 2 : multi user without NFS(new file sys) support.
- Run level 3 : multi user with NFS(new file sys) support.
- Run level 4 GUI.
- Run level 5 restart.

Boot sequence summary

- BIOS
- Master Boot Record (MBR)
- LILO or GRUB
- Kernel
- init
- Run Levels

The Linux Environment

What exactly is Linux?

- "Linux" is just an OS kernel
- the rest is additional Open Source Software
- together they are a "Linux Distribution"
- [Knoppix, Ubuntu, Redhat, Novell/SuSe]

Large choice of GUI and/or command line environments

- most popular are KDE and Gnome
- Unix-like, Mac-like, MS Windows-like, NeXT-like
- advanced shell environments
- web front-ends, GUI front-ends
- [KDE, Gnome, Windowmaker, bash, zsh, emacs, mc]

systemd

Systemd

systemd is a system and service manager for Linux. It is the default init system for Debian since DebianJessie. Systemd is compatible with SysV and LSB init scripts. It can work as a drop-in replacement for sysvinit.
Systemd

- Provides aggressive parallelization capabilities
- Uses socket and D-Bus activation for starting services
- Offers on-demand starting of daemons
- Implements transactional dependency-based service control logic
- Tracks processes using Linux cgroups
- Supports snapshotting and restoring
- Maintains mount and automount points
- Systemd runs as a daemon with PID 1.

Systemd

Systemd tasks are organized as units. The most common units are services (.service), mount points (.mount), devices (.device), sockets (.socket), or timers (.timer). For instance, starting the secure shell daemon is done by the unit ssh.service. Systemd puts every service into a dedicated control group (cgroup) named after the service. Modern kernels support process isolation and resource allocation based on cgroups. Targets are groups of units. Targets call units to put the system together. For instance, graphical.target calls all units that are necessary to start a workstation with graphical user interface. Targets can build on top of another or depend on other targets. At boot time, systemd activates the target default.target which is an alias for another target such as graphical.target. Systemd creates and manages the sockets used for communication between system components. For instance, it first creates the socket /dev/log and then starts the syslog daemon. This approach has two advantages: Firstly, processes communicating with syslog via /dev/log can be started in parallel. Secondly, crashed services can be restarted without processes that communicate via sockets with them losing their connection. The kernel will buffer the communication while the process restarts.

systemd Basic usage

Getting information on system status

Show system status:

```
$ systemctl status
```

List failed units:

```
$ systemctl --failed
```

List installed unit files:

```
$ systemctl list-unit-files
```

Managing services

List all running services:

```
$ systemctl
```

Activates the service "example1" immediately:

```
# systemctl start example1
```

Deactivates the service "example1" immediately:

```
# systemctl stop example1
```

Restarts the service "example1" immediately:

```
# systemctl restart example1
```

Shows status of the service "example1":

```
# systemctl status example1
```

Enables "example1" to be started on bootup:

```
# systemctl enable example1
```

Disables "example1" to not start during bootup:

```
# systemctl disable example1
```

Creating or altering services

Units are defined by individual configuration files, called unit files. The type of the unit is recognized by the file name suffix, .mount in case of a mount point. Unit files provided by Debian are located in the /lib/systemd/system directory. If an identically named local unit file exists in the directory /etc/systemd/system, it will take precedence and systemd will ignore the file in the /lib/systemd/system directory. Some units are created by systemd without a unit file existing in the file system.
System administrators should put new or heavily-customized unit files in /etc/systemd/system.
For small tweaks to a unit file, system administrators should use the "drop-in directory" feature Start by determining the canonical systemd service name (e.g. ssh.service, not an alias like sshd.service). We'll use "name.service" for this example.

Creating or altering services

- Create the directory /etc/systemd/system/name.service.d
- Create files inside this directory ending with a ".conf" suffix. For example, /etc/systemd/system/name.service.d/local.conf
- Each file should contain the section headers and section options to be overridden, using the same format as unit files.

Failed units

In some cases units enter a failed state. The statuscommand can be used to find out some details:

$ systemctl status <UNITNAME>

Failed units can be manually cleared out:

 # systemctl reset-failed

File system

Second extended file system(EXT2)

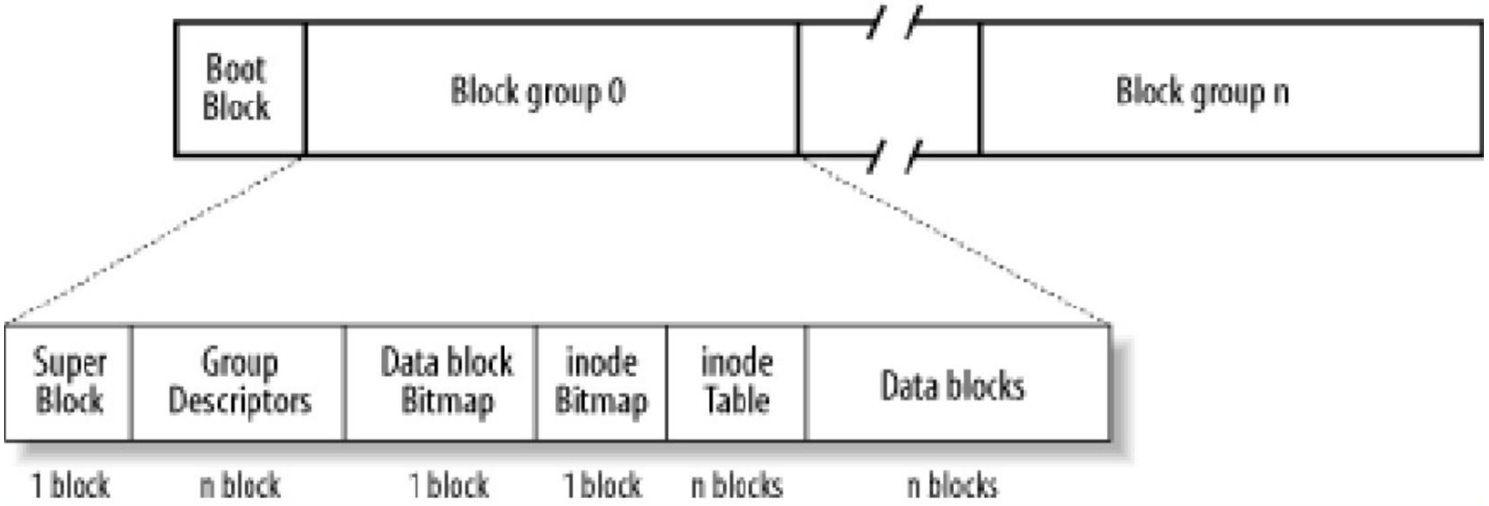
- Commonly used in Linux.
- Building blocks known as data blocks(4 Kb in size) : similar to FAT32 file sys.
- Allocation of blocks similar to FAT32.Thus physical sizes and logical sizes in EXT2 differ as they do in FAT32.
- Main components:
  - Directories.
  - Inodes.
  - Data blocks.

Second extended file system(EXT2)

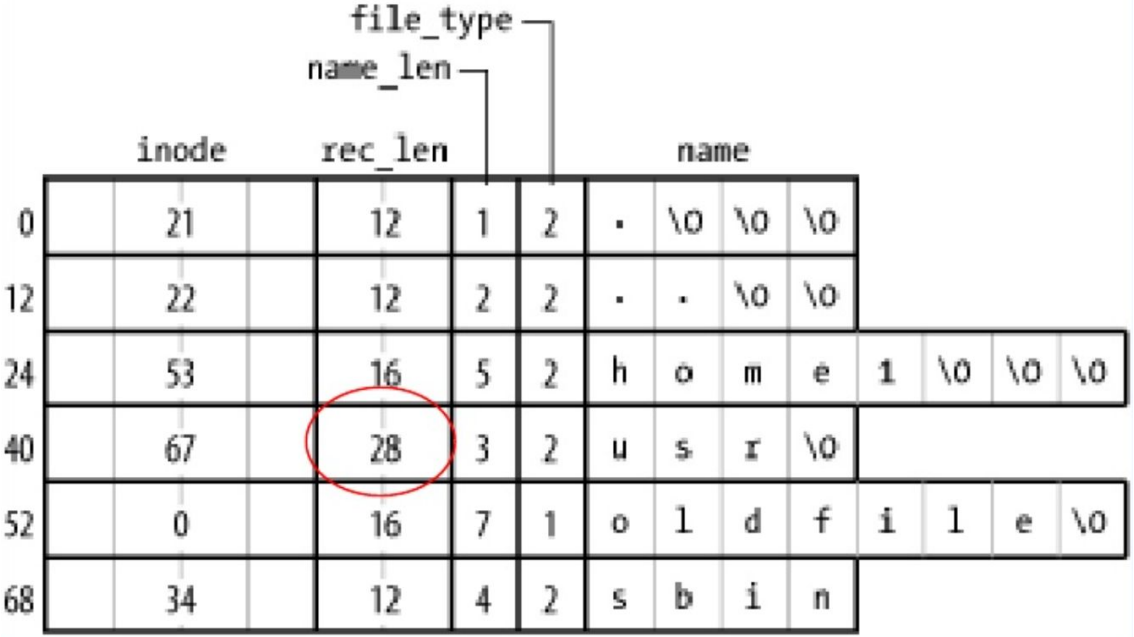
- Directories store the filenames in the file sys and the inodes associated with the files.
- In EXT2, files are represented by inodes. An inode contains attributes like file type, size, access rights, timestamps address of data blocks.
- An Inode is similar to a directory entry in FAT32. Each inode has a unique no. identifying it.
- File sys made up of “block groups” a sequential arrangement of a group of data blocks. The entire file sys is thus managed as a series of block groups.
- Block group meta data is provided by a ‘block group descriptor’. It contains a copy of super block(size of inode table and file sys), Block bitmap(tracks allocation of each data block), part of inode table and data blocks.

ext2 Characteristics

Complicated internal structure to enhance performance, but on-disk structure is straightforward



ext2 Directory Structure



inode

There are thousands and thousands of each structure. Every file and directory requires an inode, and because every file is in a directory, every file also requires a directory structure. Directory structures are also called directory entries, or “dentries.” Each inode has an inode number, which is unique within a file system. The same inode number might appear in more than one file system. However, the file system ID and inode number combine to make a unique identifier, regardless of how many file systems are mounted on your Linux system.

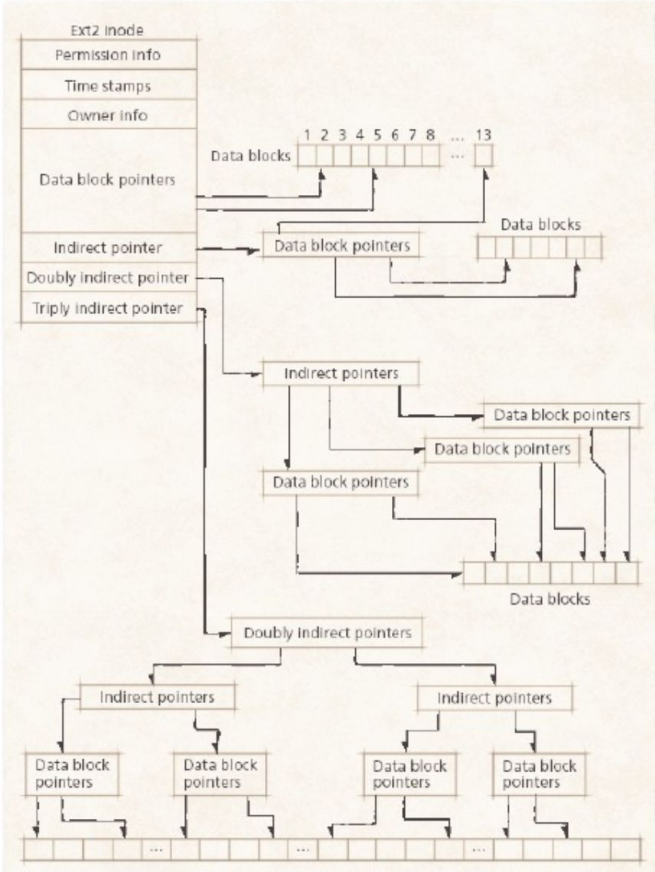
Remember, in Linux, you don’t mount a hard drive or partition. You mount the file system that’s on the partition, so it’s easy to have multiple file systems without realizing it. If you have multiple hard drives or partitions on a single drive, you have more than one file system. They might be the same type—all ext4, for example—but they’ll still be distinct file systems. All inodes are held in one table. Using an inode number, the file system easily calculates the offset into the inode table at which that inode is located. You can see why the “i” in inode stands for index.

The variable that contains the inode number is declared in the source code as a 32-bit, unsigned long integer. This means the inode number is an integer value with a maximum size of 2^32, which calculates out to 4,294,967,295—well over 4 billion inodes.

You can see the inode number of a directory just as easily as you can see them for files. In the following example, we’ll use ls with the -l (long format), -i (inode), and -d (directory) options, and look at the directory:

```
user@Ubuntu18-1:~$ ls -lid Demo/
```

```
3539490 drwxrwxr-x 4 user user 4096 sep 30 11:54 Demo/
```



ext3 Characteristics

- Binary compatible with ext2 on-disk
- Reason for existence: huge disks == huge amounts of time to restore file system consistency after improper shutdown
- (Must check inodes, etc.)
- Major improvement over ext2: log that stores info about in-progress file operations
- On boot, can check log and quickly restore file system consistency
- Journaling file systems: DANGER!

File Deletion: Linux

- *ext2* file deletion
  - Adjust previous directory entry length to obscure deleted record
  - No reorganization to make space in directories
  - “first fit” for new directory entries, based on real name length
  - Directory entry’s inode # is cleared
- *ext3* file deletion
  - Same as for ext2, but…
  - inode is wiped on file deletion, so block numbers are lost
  - Major anti-forensics issue!
  - But directory entry’s inode # isn’t cleared…

Q&A