

Unix File System

The background of the slide is split vertically. The left half is a solid green color, and the right half is white. A large, white, rounded rectangular shape is positioned on the left side, overlapping the green background. The text "Unix File System" is centered within this white shape. Below the text, a thick, dark blue horizontal bar extends from the green area across the white area.

File System

- It tracks files (ie where they are located)
- In UNIX everything is treated as file
- Some striking similarities with DOS.
- UNIX has a very large no. of files.
- Files with some relations are grouped together.
- Same file with multiple names.
- Provides adequate securities to file.
- Makes little distinction between Regular files and special files
 - disk files
 - device files.
 - (printer, floppydrive, hard disk, console etc)

Characteristics of unix file system

- **Hierarchical structure**
- **Consistent treatment of file data**
- **The ability to create and delete files**
- **Dynamic growth of files**
- **The protection of files**
- **The treatment of peripheral devices as files**
- **File systems are separate and distinct**
- **File system cannot extend across a partition boundary**
- **File systems cannot overlap**
- **To access the data in the file system it has to be mounted in a directory**

- **Path names Starting with Root "/"**
- **Absolute path names, relative path names,**
- **Programs without the knowledge of internal formatted pattern of data But returning an unformatted byte stream of data.**
- **Syntax of accessing the data in a file is defined by the system which is identical for all program**
- **But semantic of data are imposed by program eg text processing package and accounting package (fixed record length)**
- **Directories are treated as files**

Concept of a file

- **A file in Unix is a stream of bytes (slots)**
- **File may contain text data and even machine language code**
- **UNIX file doesn't have a predefined structure to work with it**
- **In UNIX everything is treated as a file, even devices and directories are also treated as a file**
- **User accesses a file by a user assigned file name**

- **File name can be a sequence of characters**
- **(a maximum of 14.4)**
- **May be identified by two or more file names**
- **(Multiple links to a file)**
- **Different files with same relation can be grouped under a directory**
- **Internally each file is assigned a unique inode number**

Types of files

- **Ordinary files**
- **Used to store data**
- **Users can add data to ordinary files directly,
eg. executable programs**
- **Directory files**
- **Contains list of files in the directory**

- **Each entry in the list consists of two parts :**
 - a) name of the file**
 - b) inode number of the file**
(a pointer to the actual file on the disk)
- **Special files**
- **Used to reference physical devices, such as terminals, printers, disk, tape drives etc.**

Ordinary files

- **Used to store data**
- **Users can add data to ordinary files directly**
- **eg. executable programs**
- **Includes all data, source programs, object and executable code, all UNIX programs, as well as any files created by the user.**
- **Commands like pwd, ls, etc are treated as ordinary files.**

Data present in Ordinary file

- Text files** - **Lines of ASCII characters separated by a newline**
- Commands** - **Sequence of commands interpreted by Unix text**
- Data** - **File containing data as a stream of bytes**
- Executable** - **File containing machine language instructions.**

Concept of directories

- **Gathering together related files in a common place**
- **Every file is assigned to a directory**
- **Directories have names (maximum of 14.4 characters)**
- **A file within a directory can itself be another directory**
- **(A Sub-directory)**

- **Internally a directory is just a table that contains a list of file names and their Inode Number (a pointer to the actual file on the disk).**

Filename	Inode	Number
	test1	23
	test2	24

An upper limit of around 35 sub-directories

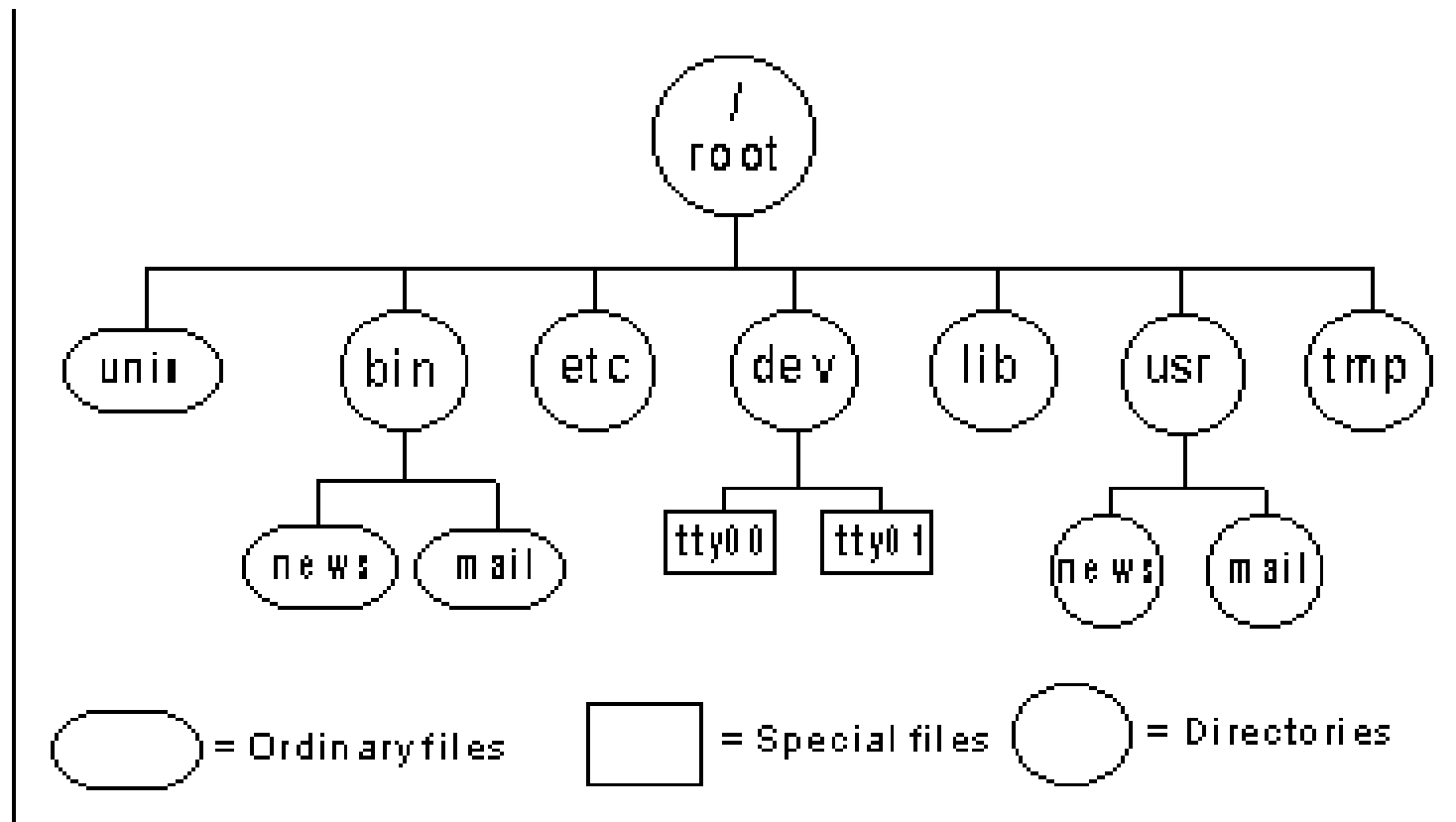
Examples:-

**/ (forwardslash) - Prime or Root Directory Note
in DOS it is \ (Backwardslash)**

/usr - Directory for the user

**/usr/trg1 - Sub-Directory for trg1 under
Directory usr**

Directory Structure of the root file system



Concept of special files

- **Used to refer physical devices such as terminals, disks, floppy drives, tape drives etc.**
- **Read and written like ordinary files**
- **Requests cause activation of the associated physical device**
- **Device Drivers are associated with each special file.**

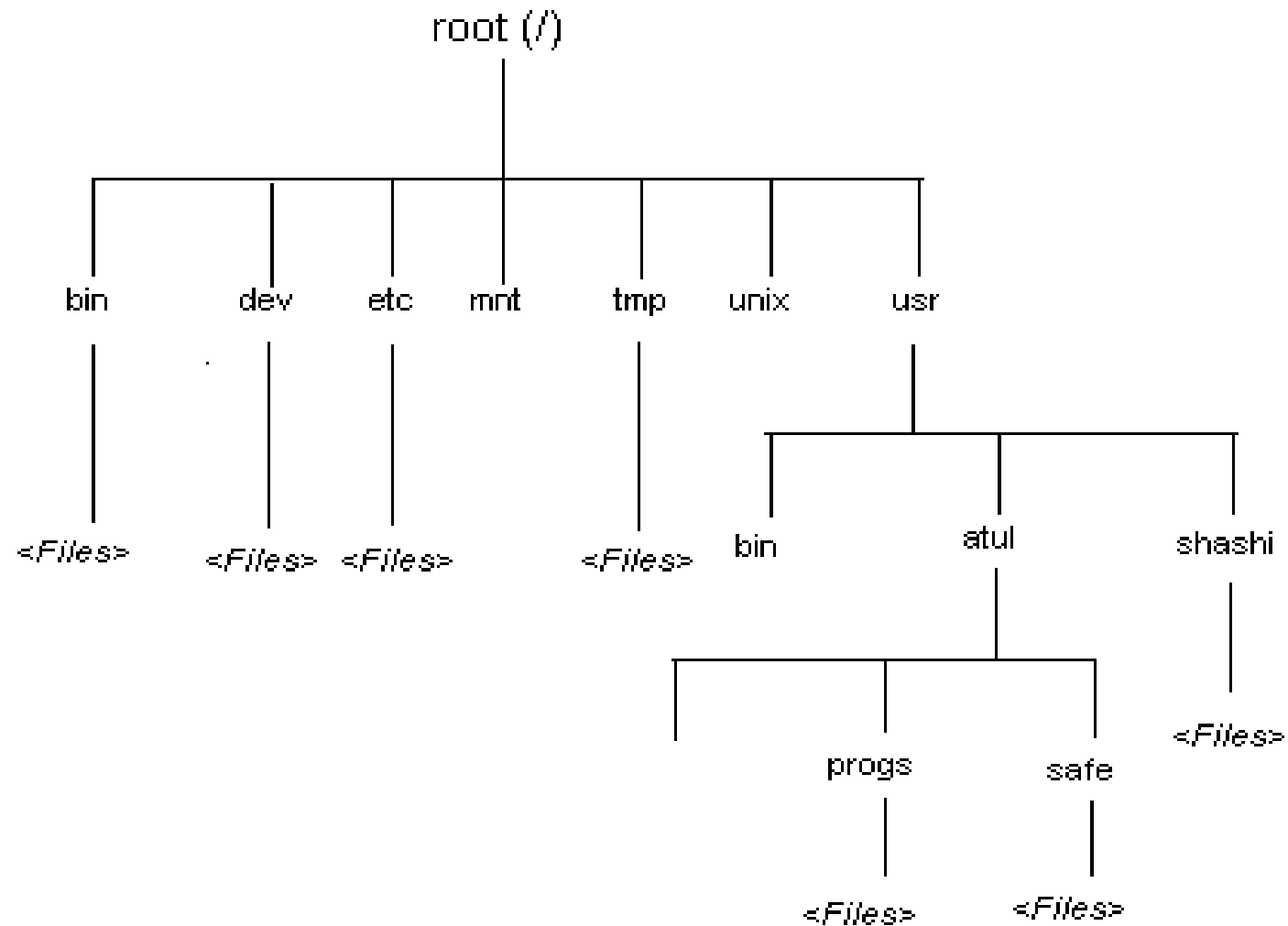
Type :

Character file	Terminal (tty0, tty1) Character	Transmits data Character by
Block file	Disk Drives	Transmits data in 512 (hd0,fd0) or 1024 byte chunks

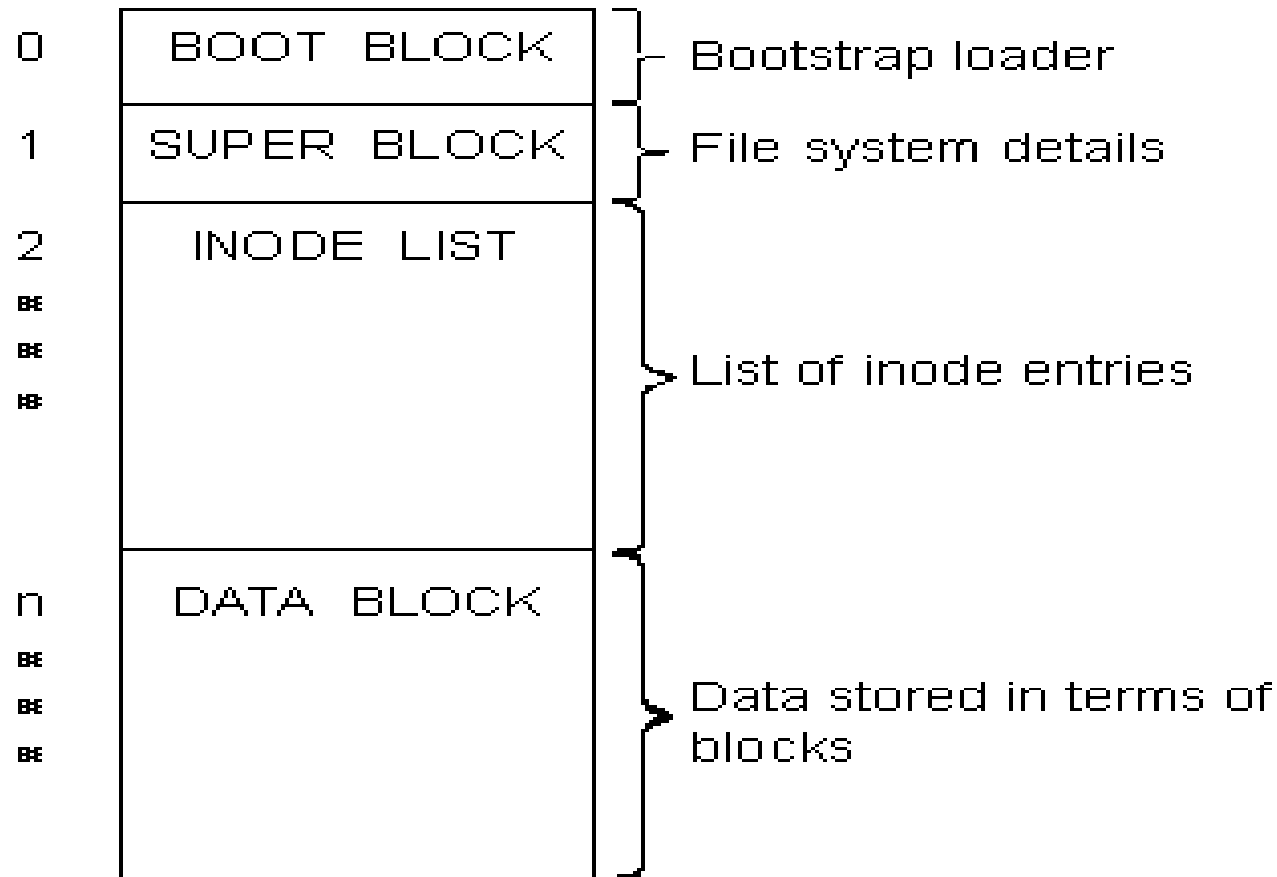
Major and Minor numbers identify the hardware link.

Standard Root Directory

/bin	Basic Unix Utilities Contains many executable programs and utilities.	cp, mv, ln
/dev	Special I/O device files Contains special files that represent peripheral devices such as the console, the line printer, user terminals and disks.	fd0, lp0
/etc	Administrative Programs Contains programs and data files for System Administration	passwd shutdown
/lib	Libraries used by UNIX Contains libraries for programs and languages	libc.a
/usr/bin	Unix utilities	
/usr/adm	Administrative commands and files	cal, bc, spell
/tmp	Temporary files Created on error conditions or for data transfer	adduser
/unix	UNIX kernel	



Structure of file system partition



Boot Block

- **It occupies the first block of the file system**
- **Every file system has an empty Boot Block**
- **Boot Block in the Root file system only**
contains a bootstrap code which is read to boot
or initialize the operating system

Super Block

It contains the following information of the file system :

- **The state of the file system**
- **Size of the file system**
- **Number of free blocks in the file system.**
- **List of free blocks available in the file system**
- **Index of next free block in free block list**
- **Size of inode list**

- **Number of free inodes in the file system**
- **Index of next free inode in the free inode list**
- **Lock fields for the free blocks and free inode lists**
- **Flag indicating that super block has been modified**

Inode list

- **Inode list follows the super block in the file system.**
- **It contains inode entries for each file in the file system**
- **These inode entries are always referred while accessing the file**
- **Administrator specifies the size of the inode list while configuring a file system**
- **The size of the inode list defines the maximum number of file entries in the file system**
- **The Kernel references inodes by indexing into the inode list.**

Data Blocks

- **Remaining space in a file system after the Inode List is taken up by Data Block**
- **It is used to store files in terms of blocks**
- **When a file is created the free blocks in the Data Block are allocated to it**
- **An allocated block can belong to one and only one file in the file system.**

Inodes

- **An inode (information node) is a table containing information about a file.**
- **Inode entry for each file is there in the inode list**
- **Inode entry for each file is of 64 bytes**
- **The inode describes the collection of blocks used for a file**

- **The name of the file is not included in the inode**
- **Inodes are identified by their position in the file system, (the inode number)**
- **Inode numbers are unique only within a file system**

The contents of an inode indicates

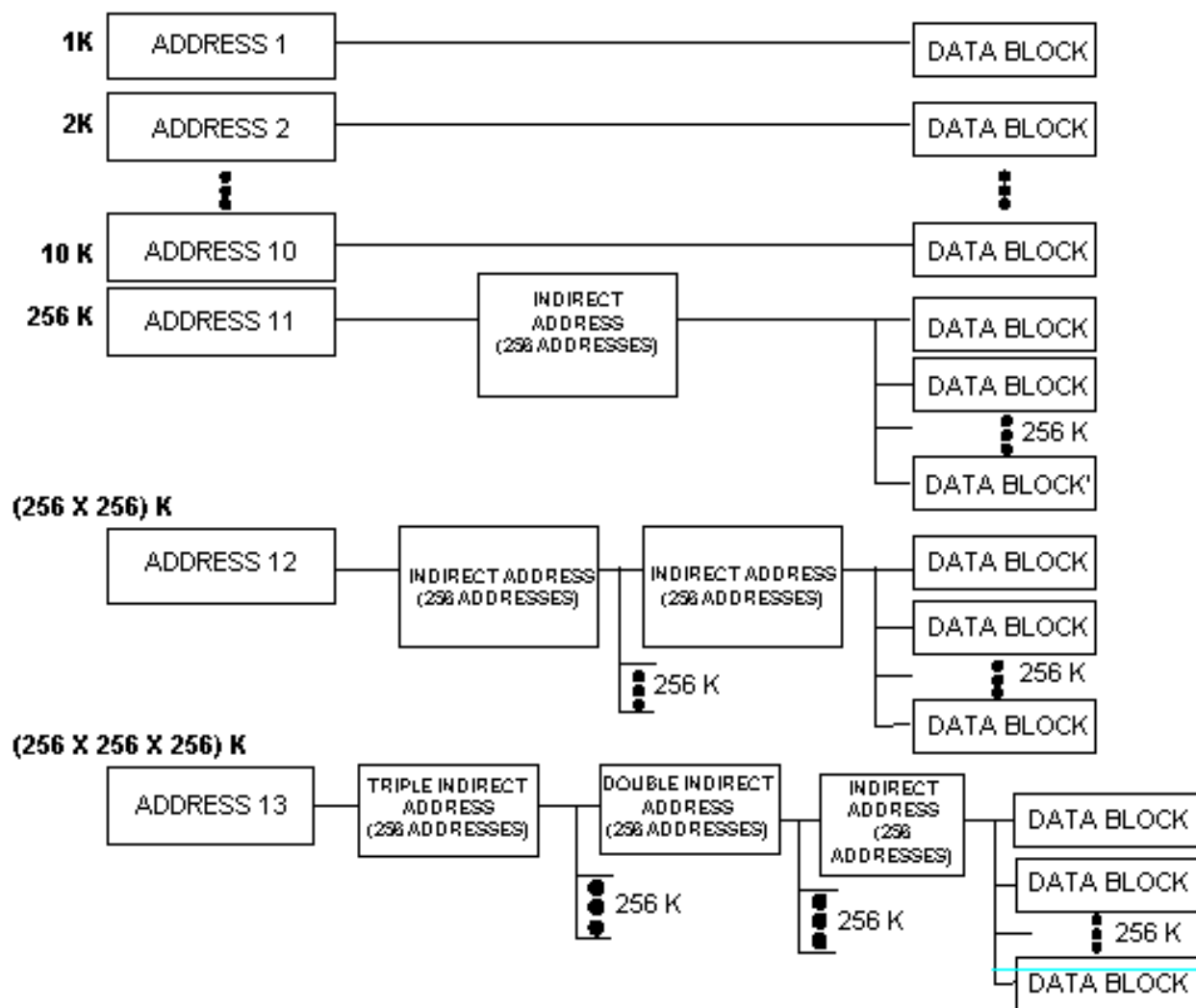
- **File type regular, directory or device file**
- **File access permissions**
- **User id number of file owner**
- **Group id number of file group owner**
- **Link count (the number of aliases the file has)**
- **Size of file in bytes**

- **Date and time of last modification**
- **Date and time of last file access**
- **Date and time of last inode change**
- **Addresses of the data blocks in which file is stored are there in the last 13 bytes**
- **An array of 13 pointers to the file**

How Inode Points to Data Blocks

- The inodes use 13 pointers to access the datablocks related to files
- The first 10 are direct pointers and each points to a data block
- The next three are indirect pointers and each points to a block containing addresses of data blocks or address of indirect blocks themselves
- An indirect block is a block containing address of other data blocks, or indirect block

Addressing of the data blocks



Security Features

- **Security on each file:-**
- **Owner**
- **Group**
- **All others**
- **Separate security for :-**
- **Read**
- **Write**
- **Execute**
- **Time stamp on each file**
- **Modification time**
- **Access time**

File access modes

- **Read access mode**
- **For file:- allows examination of file contents.**
For directory:- allows listing of files within directory.

Write access mode

- **For file:-allows changing contents of file. For directory:- allows creating new files & removing old ones .**

- **Execute access mode**
- **For file:- allows executing file as command**
- **For directory:- allows searching directory**