Lecture – 7

Section-B

Theoretical concept of Unix Operating System

UNIX file system

The File Structure

- UNIX looks at everything as a file and any UNIX system has thousands of files.
- If you write a file, you add one file to the system, while you compile it, you add some more files.
- Files grow rapidly , & if they are not organized properly, you'll find it difficult to locate them.
- **The File:**

  The file is a container for storing information. Generally we can treat these files as a sequence of characters.

  if you write a file "foo" & write three characters a, b & c into it. then foo will contain only the string a b c & nothing else.

✓ **A file's size is not stored into the file nor even its name.**

✓ **All file attributes are kept in a separate area of the hard-disk, not directly accessible to humans, but only to the kernel.**

- Unix treats directories and devices as files as well.
- A Directory is simply a folder where you store file names & other directories.
- All physical devices like the hard-disk, memory, CD-ROM, printer & modem are treated as files.
- The shell is also a file & so is the kernel. And if you're wondering How Unix treats the main memory in your system, it's a file too.

Three categories of Files are there :

- **Ordinary Files:** Also known as regular file, it contains only data as stream of characters.
- **Directory files:** Its commonly said that a directory contains files & other directories – It contains names & numbers associated with each name.
- **Device Files**: All devices and peripherals are represented by files. To read & write a device, you've to perform these operations on its associated files.

❑ Ordinary Files

An ordinary file or regular file is the most common file type. All programs you write belong to this type – An ordinary file itself can be divided into two types:

- Text files
- Binary Files

"**A text file**" only contains printable characters & you can often view the contents & make sense out of them.

Example : All C & Java programs source codes are text files.

A "**Binary file**" contains both the printable and unprintable characters that cover the ASCII range(0 to 255)

Example : Most Unix commands are binary files, object codes & executable files you produce by compiling C programs are also binary files.

Pictures , sounds & video files are binary files.

Displaying such files with a simple cat command produces unreachable output & may even disturb your terminal settings.

❑ **Directory files:**

- A directory contains no data, but keeps some details of the files & sub-directories that it contains.
- The Unix file system is organized with a number of directories & sub directories & you can also create them as when you need.
- We often need to group a set of files belongs to a specific application.
- This allows two or more files in a separate directories tto have the same file name.
- A directory file contains an entry for every file & sub-directory It houses. If you've 20 files in directory there will be 20 entries in the directory. Each entry has two components -
  - ✓ the file name.
  - ✓ A unique identification number for the file or directory (called the inode number)

❑ Device files

- You’ll also be printing files, installing software from CD-ROMs or backing up files to CD-ROM Drives.
- All these activities are performed by reading or writing the file representing the device.
- When we restore files from CD Drive , you read the file associated with CD drive.
- Device filenames are generally found inside a single directory structure, **/dev**
- A device file is indeed special; its not really a stream of characters. In fact, it doesn’t contain anything at all.

What’s in a (FILE) Name ??/ Naming Convention

On most of Unix systems today, a file name can consist of up to 255 characters, though this figure normally not reached.

Files may or may not have extensions & can consist of practically any ASCII characters except the / and the NULL character (ASCII value 0).

It is recommended that only the following characters should be used in filenames:

- Alphabetic characters & numerals.
- The period (.), hyphen(-), & underscore (_)

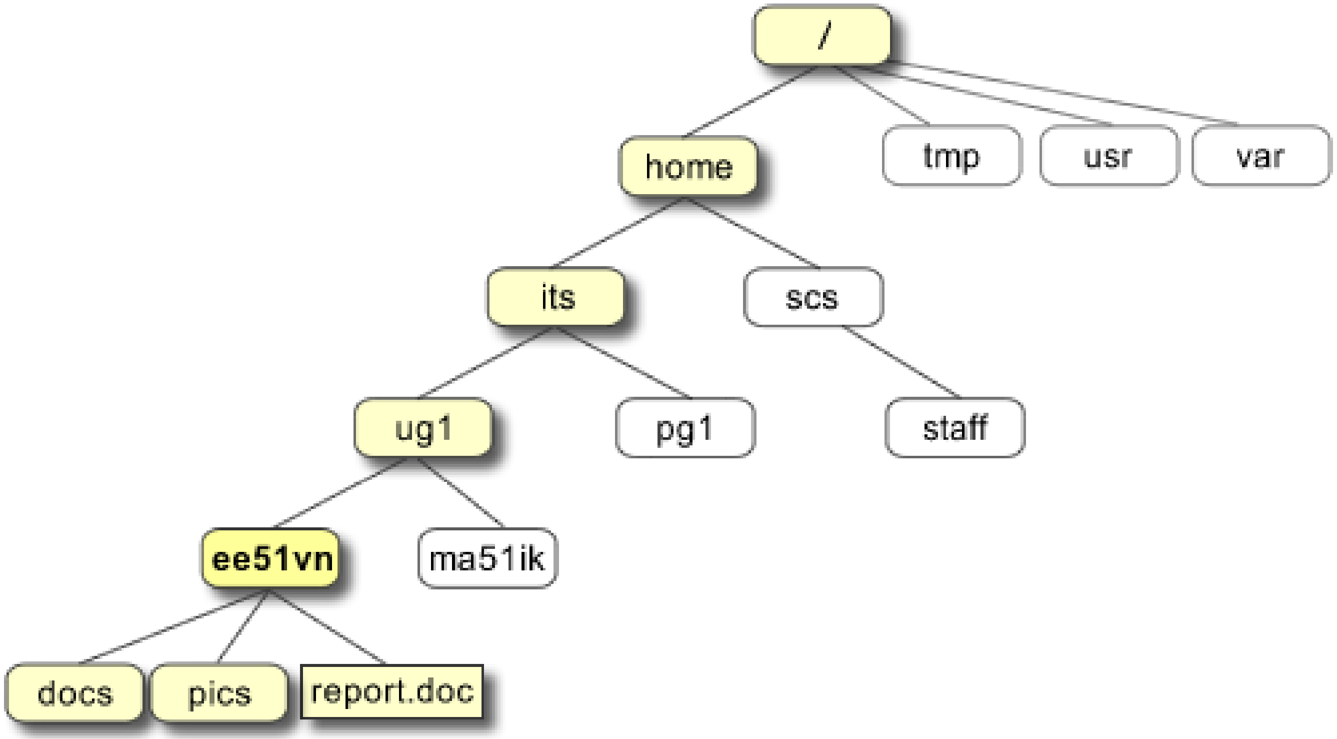
✓ A file can have as many dots embedded in its name; a.b.c.d.e is a perfectly valid file name.

✓ A file name can also begin with (.) & can also end with another dot.

✓ Unix is sensitive to case; chap01, Chap01and CHAP01 are three different filenames, and it’s possible for them to coexist in same directory.

The Parent-child relationship:

-



The Unix file system tree

/home/its/ug1/ee51vn/docs

- All files in Unix are “related” to one another. The file system in Unix is a collection of all of these related files (ordinary, directory & device files) organized in an inverted tree structure.
- The implicit feature of every Unix system is that there is a top, which serves as the reference point for all files.
- This top is called “root” & is represented by / front slash. Root is actually a “directory”.
- The root directory has a number of sub-directory under it.
- These subdirectories in turn have more sub-directories and other files under them.
- For instance, home and usr are two directories directly under /, while its and scs are sub directories under home.

Applications

- Vendors of some proprietary Unix systems, such as SunOS/Solaris, System V Release 4, HP-UX, and Tru64 UNIX, have adopted UFS. Most of them adapted UFS to their own uses, adding proprietary extensions that may not be recognized by other vendors' versions of Unix. Surprisingly, many have continued to use the original block size and data field widths as the original UFS, so some degree of (read) compatibility remains across platforms. Compatibility between implementations as a whole is spotty at best and should be researched before using it across multiple platforms where shared data is a primary intent.
- As of Solaris 7, Sun Microsystems included UFS Logging, which brought filesystem journaling to UFS. Solaris UFS also has extensions for large files and large disks and other features.

Research

- **radmind** : A suite of Unix command-line tools and a server designed to remotely administer the file systems of multiple Unix machines. For Mac OS X, there's also a graphical interface.
- At its core, radmind operates as a tripwire. It is able to detect changes to any managed filesystem object, e.g. files, directories, links, etc. However, radmind goes further than just integrity checking: once a change is detected, radmind can optionally reverse the change.
- Each managed machine may have its own loadset composed of multiple, layered overloads. This allows, for example, the operating system to be described separately from applications.
- Loadsets are stored on a remote server. By updating a loadset on the server, changes can be pushed to managed machines.