Chapter 3

Directory Management

School of Computer Science,
MIT-World Peace University,
Pune, Maharashtra, India.

Amol D. Vibhute (PhD)
Assistant Professor

Email ID:- amol.vibhute@mitwpu.edu.in

Roadmap of Chapter

- Introduction
- Home Directory
- Absolute/Relative Pathnames
- Listing Directories
- Creating Directories
- Creating Parent Directories
- Removing Directories
- Changing Directories
- Renaming Directories

Introduction:

- A directory is a file the only job of which is to store the file names and the related information.
- All the files, whether ordinary, special, or directory, are contained in directories.
- Unix uses a **hierarchical structure** for organizing **files** and **directories**. This structure is often referred to as a **directory tree**.
- The tree has a **single root node**, the **slash character (*/*)**, and all other directories are contained below it.

Home Directory:

- The directory in which you find yourself when you first login is called your **home directory**.
- You will be doing much of your work in your home directory and subdirectories that you'll be creating to organize your files.
- We can go in your home directory anytime using the following command −
  - $cd ~
  - Here ~ (Tilde) indicates the home directory.
- Suppose we want go in any other user's home directory, use the following command –
  - $cd ~username
- To go in our last directory, we can use the following command –
  - $cd -

Absolute/Relative Pathnames:

- Directories are arranged in a hierarchy with **root (*/*)** at the top.
- The position of any file within the hierarchy is described by its **pathname**.
- Elements of a pathname are separated by a */*.
- A pathname is absolute, if it is described in relation to root, thus absolute pathnames always begin with a */*.

Absolute Pathnames:

- To traverse a location of file in the file system, we can start the searching from the root of the directory (*/*). Then follow the branches up to the desired file name.
- So, if we want to locate the file date.c in the figure, the path name will be /usr/src/cmd/date.c
- Above path name is full path name and it is beginning with root.

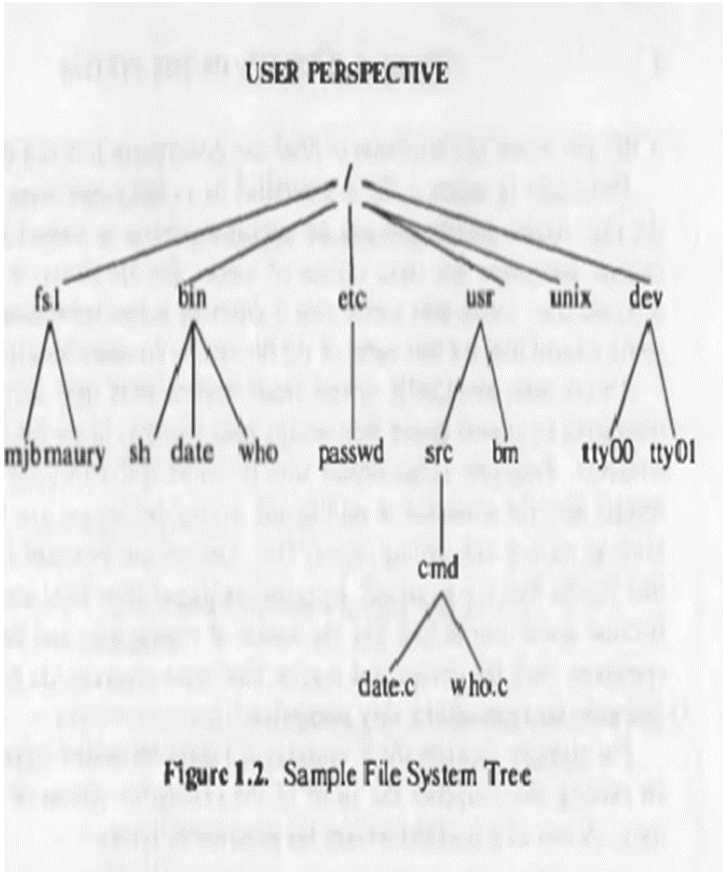

Figure 1.2. Sample File System Tree

Relative Pathnames:

- If we are already switched in **/usr /src** then path of **date.c** will be just **cmd/date.c**.
- This is because of path of a file is relative to current directory in which you are switched such path name is termed as relative pathname
- For Ex: $pwd
- **Note**: pwd is print working directory. The word print in print working directory means “print to the screen,” not “send to printer.”

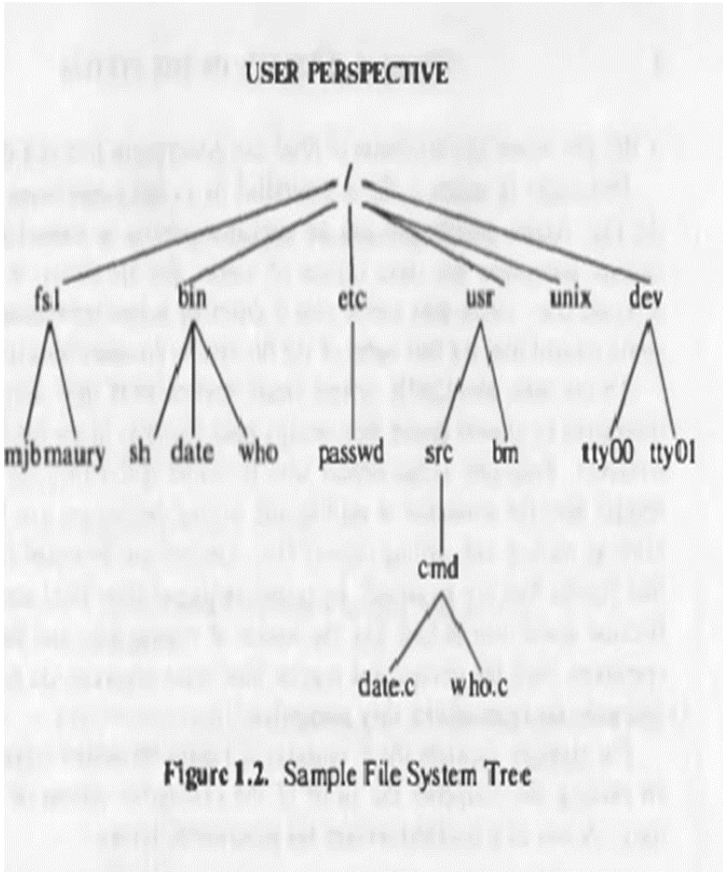

Figure 1.2. Sample File System Tree

Listing Directories:

- To list the files in a directory, we can use the following syntax −
  - $ls dirname
- Following is the example to list all the files contained in **/usr/local** directory –
- Syntax: $ls /usr/local
  - Output: bin etc lib

- Syntax: $ ls -R /home
  - We can see the tree like structure using **-R** option of **ls** command.

Creating Directories:

- Directories are created by the following command −
  - $mkdir dirname
- Here, directory is the **absolute or relative** pathname of the directory we want to create. For example, the command –
  - $mkdir demo
  - Creates the directory demo in the current directory.
- For Ex: $mkdir /tmp/test-dir
  - This command creates the directory **test-dir** in the **/tmp** directory. The mkdir command produces no output if it successfully creates the requested directory.
- If you give more than one directory on the command line, mkdir creates each of the directories.
  - For example, − $mkdir hello unix world
  - Creates the directories hello unix and world under the current directory.

Creating Parent Directories:

- Sometimes when we want to create a directory, its parent directory or directories might not exist. In this case, mkdir issues an error message as follows −
  - $ mkdir /mydir/new
  - mkdir: cannot create directory '/mydir/new': No such file or directory
- In such cases, we can specify the **-p** option to the **mkdir** command. It creates all the necessary directories for you.
  - For example –$ mkdir -p /home/amol.vibhute/newunix
- The above command creates all the required parent directories.

Removing Directories:

- Directories can be deleted using the **rmdir** command as follows −
- **Syntax**: $rmdir dirname
- **Note** − To remove a directory, make sure it is empty which means there should not be any file or sub-directory inside this directory.
- We can remove multiple directories at a time as follows −
- **Syntax**: $rmdir dirname1 dirname2 dirname3
- The above command removes the directories dirname1, dirname2, and dirname3, if they are empty. The rmdir command produces no output if it is successful.

Changing Directories:

- We can use the **cd** command to do more than just change to a home directory.
- We can use it to change to any directory by specifying a valid absolute or relative path.
- **Syntax**: $cd dirname
- Here, **dirname** is the name of the directory that you want to change to.
- **Syntax**: $cd /usr/local/bin
- Changes to the directory /usr/local/bin. From this directory, we can **cd** to the directory /home/amol.vibhute using the following relative path,
  - $cd ..

Renaming Directories:

- The **mv** (move) command can also be used to rename a directory.
- **Syntax**: $mv olddir newdir
- You can rename a directory mydir to yourdir as follows –
- $mv mydir yourdir

References:

- Unix Shell Programming: Yashwant Kanitkar, BPB Publications, New Delhi.

Thank You !!!