Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Chapter 1

UNIX & Shell Scripting

-An Introduction

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

Amol D. Vibhute (PhD)
Assistant Professor

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

Course Goal:

- Understand the fundamental concepts and basic principles of Unix and Shell Scripting
  - Introduction to Unix
  - File Management
  - Directory Management
  - File Permission / Access Modes
  - Shell Prompt, Variables, operators and loop control, Input/output Redirections and Functions.
- Class information:
  - Theory hours:
    - Monday, Thursday and Friday (09:10-10:10, 13:00-14:00, and 08:00-09:00 hrs, (IST) respectively).

Reference Books:

- Required:
  - Unix Shell Programming: Yashwant Kanitkar, BPB Publications, New Delhi.
- Recommended:
  - Classic Shell Scripting: Hidden Commands that Unlock the Power of Unix Book by Stephen P. Robbins.
  - Shell Scripting Tutorial Book by Steve Parker.
  - Linux Command Line and Shell Scripting Bible Book by Richard Blum.

Tentative topics:

- Introduction to Unix
- Directory Management
- File Permission / Access Modes
- Shell Scripting
- Shell Variables, operators and loop control
- Shell Input/output Redirections
- Shell Functions.

Marking Scheme:

- External Exam: 50 Marks
  - Minimum Required: 20
    - Based on syllabus of university (MIT-WPU) (Closed book, comprehensive).
- Internal Exam: 50 Marks
  - Minimum Required: 20
    - Tests/Quizzes (20)
    - Based on syllabus of university (MIT-WPU), homework assignments (10), Oral (10) and Attendance (10).
      - » Homework from textbook
- Evaluation Scheme:
  - Overall Performance,
  - Positive attitude towards subject, lectures and pedagogue,
  - Assignments,
  - Attendance,
  - Internal Points.

Your Best Strategy:

- Eat healthy food, be healthy & happy, enjoy the fun and do hard work (Study)….
  - And next:
    - Come to every lectures (Online mode),
    - Read books, articles and authenticate websites related to Unix and Shell Scripting,
    - Do not wait till last minute to prepare for exam or work on practical's, lectures and assignments.
    - Do practical's by yourself.

Lectures need your help!

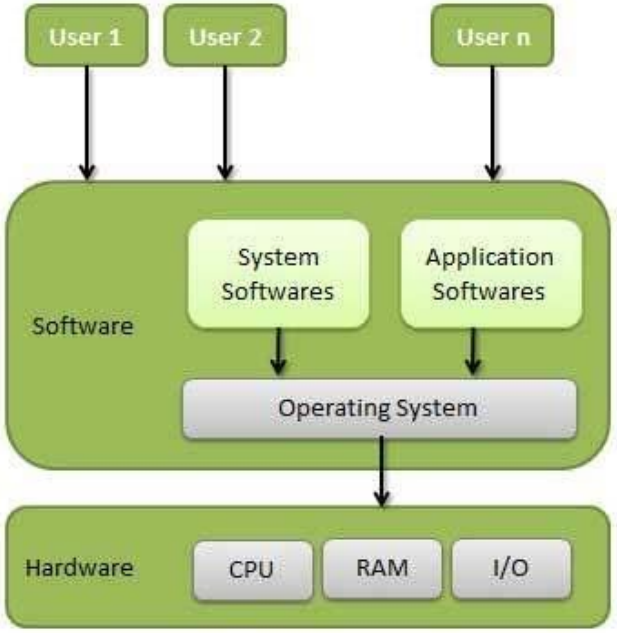
- Ask questions ???
- Tell my errors to me if any!
- Make suggestions!
- Read something interesting and relevant to this course? Announce it in class! Give presentation on it!

Roadmap of Chapter

- Introduction to OS and Unix,
- Brief history of Unix,
- What is Unix?
- Unix features,
- Similarities between Unix, MS-DOS and MS-Windows,
- Differences between Unix, MS-DOS and MS-Windows,
- Different flavors of Unix,
- Unix Architecture,
- Unix Commands and Utilities,
  - Rules of Unix Commands,
  - Types of Unix commands,
  - Some important Commands,
- Files and Directories.

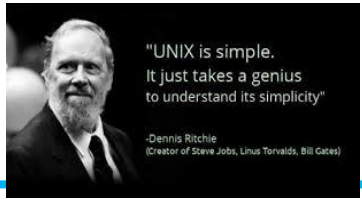
Introduction:

- **Operating System:**
  - A set of programs which acts as an interface between user interface and computer hardware and controls the execution of all kinds of programs..
  - A software which performs all the basic tasks like **file management, memory management, process management, handling input and output, and controlling peripheral devices** such as disk drives and printers.
  - Important functions of an operating System:
    - Memory Management,
    - Processor Management,
    - Device Management,
    - File Management,
    - Security,
    - Control over system performance,
    - Job accounting,
    - Error detecting aids,
    - Coordination between other software and users.



Unix: A brief History

- **1964 (Five members)**–AT & T Bell Lab *GE(General Electric company) & Project MAC of MIT (Massachusetts Institute of technology): MULTICS–**Multi**plexed **I**nformation & **C**omputing **S**ervices.
- **1969**– Due to failure of achievement as the goal & due to over–budget, Bell Lab withdrew its participation. But some of members of the above project from bell like **ken Thompson** & **Dennis Ritchie** took the idea Multics and did a paper design of new operating system in 1969. They implemented this design on **DEC PDP–7** machine and was named as UNICS–where "MULTI" is replaced by **"UNI"** by another member **Brain Kernighan**. Later in UNICS is made shorter to UNIX.
- In **1971** UNIX was tried on more powerful machine PDP–11. System was very small, requiring 16 kb fore system, 8kb for user programs and maximum limit at 64 kb for file. Thus disk of 512 KB was enough to hold the complete system.



Cont….

- Initially the whole system was written in the **assembly language** and thus not easy to become portable on another machine. In 1972, **Dennis Ritchie** developed **"C"** language and in 1973 the whole UNIX operating system was re-written in c (keeping very small assembly code in it).
- The operating system was still in use only for Bell Laboratory. But in 1977, it was first ported on NON-PDP machine called **Interdata 8/32**..
- Realizing its commercial value, AT & T released UNIX commercially in the world with its source code. After that many institutes made their own Unix standards by adding more and more functionality to it.
- From 1977 to 1982, AT & T itself created **UNIX System III** (for commercial purpose),**UNIX System IV** (Only for internal use) and in Jan1983, the **UNIX System V** was sold commercially and become the most popular one.
- Due to availability of source code, University California in Berkeley developed their own UNIX standard for VAX machine, which was called as **UNIX 4.3 BSD** (**B**erkeley **S**oftware **D**istribution)
- **Unix versions** (1975-80): IBM-AIX, Sun Solaris, Mac OS, HP-UX, **Linux (1991) MINIX**- Andrew Tanenbum.
- Some Unix are open source and some or not.

What is Unix?

- A set of programs that act as a link between the computer and the user.
- The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the **operating system** or **the kernel**.
- Users communicate with the **kernel** through a program known as the **shell**. The shell is a **command line interpreter**; it translates commands entered by the **user** and converts them into a language that is understood by the kernel.

Unix Features:

- **Multiuser**: Several people can use a Unix computer at the same time.
- **Multitasking**: A user can also run multiple programs at the same time.
- **Multi-process**: Each user can execute several processes simultaneously.
- **Hierarchical Structure**: Unix directories are present like a tree structure to support the organization and maintenance of files.
- **Open system**: Some Unix OS are open source. Hence, users can modify as per requirement.
- **Portability**: Allows users to transfer data from one machine to another.
- **Programming facility**: can be used as a programming or scripting language.

Cont….

- **Communication facility:** allows communication between different users by providing some information.
- **Security:** system and file level security controlled by admin and owner respectively.
- **Tools and utilities:** supports several tools, libraries and utilities to help software development.
- **Help Facility:** 'man' command is used to view help content on any command.
- **Modularity:** Unix consists of multiple number of independent modules or programs which perform different elementary tasks.

Similarities Between Unix, MS-DOS and MS-Windows:

- **Unix and MS-DOS:**
  - Command Line Interface (CLI),
  - Hierarchical directory structure with the root directory at the top,
  - Read, write and execute permissions on files.
- **Unix and MS-Windows:**
  - Multitasking,
  - Built-in networking wit TCP/IP as the standard protocol.

Differences Between Unix and MS-DOS:

Unix	MS-DOS
Have a GUI	Does not have GUI
Multi-user, multi-tasking and multi-process	Single-user, single-tasking and single-process
Uses forward slashes (/) to separate directories	Uses back slashes (\) to separate directories
Mainly used in servers	Used in embedded systems
Has shell script	Has batch files
An OS with concepts like process priorities	An OS without concepts like process priorities
Case sensitive	Not a case sensitive

Differences Between Unix and MS-Windows:

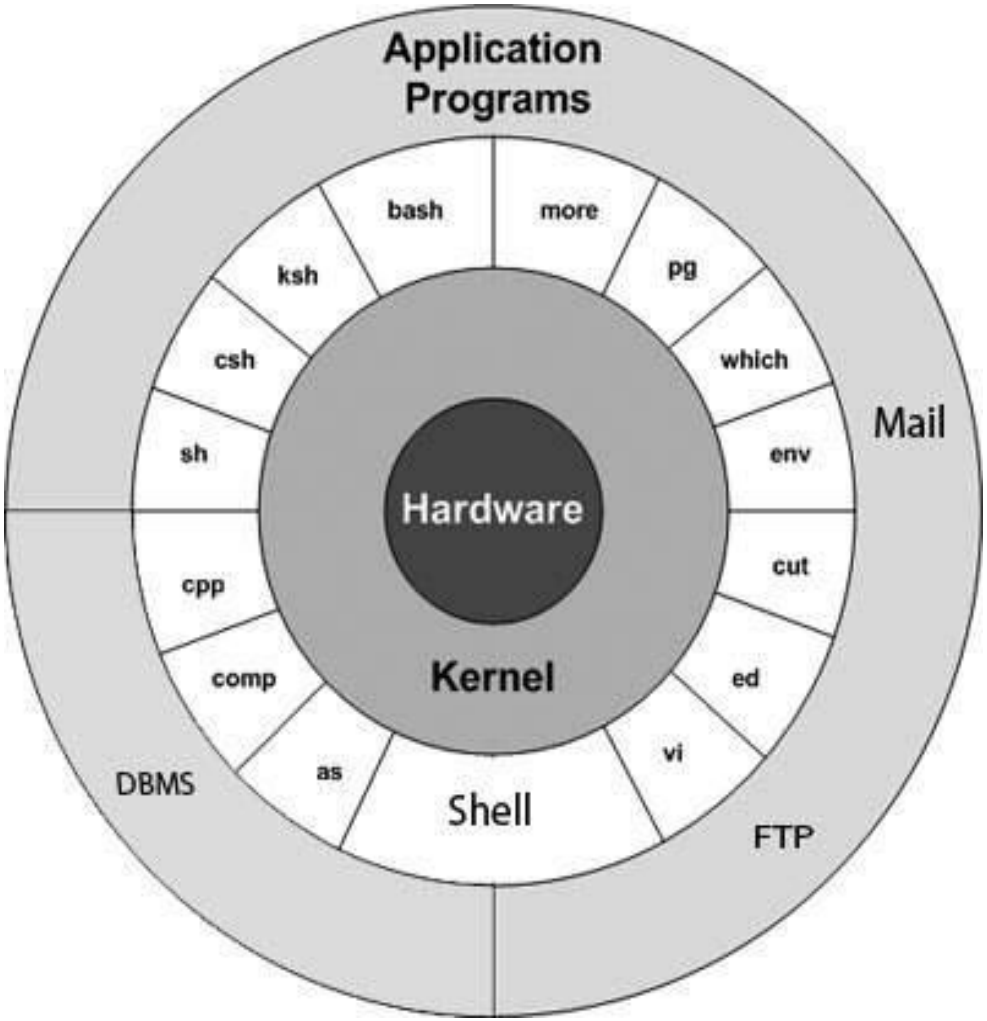
Unix	MS-Windows
Unix file system a hierarchical model	Windows file system is a flat model
Command Line Interface (CUI)	Graphical User Interface (GUI)
A multi-user and multi-tasking	Single-user and multi-tasking
Free open source	Licensed
Has dumpy terminals (without hard disks)	Do not support dumpy terminals
Case sensitive	Not a case sensitive
Not user friendly	User friendly
Supports programming facility	Do not support programming facility
Have multiple vendors	Have only one vendor, i.e. Microsoft.

Different Flavors of Unix:

Unix Flavors/Variant	Vendors/Companies/Org
AIX	IBM
HP Ux	HP
Solaris	Sun
Tru64Unix	Compaq
TRIX	SGI (Silicon Graphics)
SCO Unix	SCO (Santa Cruz Operation)
Xenix	Microsoft
UTS	Amdahl Corporation
Linux	Red Hat

Unix Architecture:

- **Hardware/Physical:**
  - Controls the use of physical system resources, such as memory manager, process manager, disk drivers, devices and so on.
  - Hardware consists of all peripheral devices (RAM, HDD, CPU and so on).
- **Kernel:**
  - Heart of Unix.
  - It acts as an interface between hardware and shell layer.
  - Most of the tasks like memory management, task scheduling and file management, and so on are performed by the kernel.
  - It manages external commands in Unix.



Cont….

- **Shell:**
  - An interface between a user application and the kernel.
  - This layer processes your requests. When we type a command at your terminal, the shell interprets the command and calls the program that we want.
  - It uses standard syntax for all commands.
  - Various shells available with most of the Unix variants/flavors are: C Shell (csh), Bourne Shell (sh) and Korn Shell (ksh) and Bourne Again Shell (bash).
  - A file "/etc/shells" contains a list of all the shells supported and available in the system.

Cont….

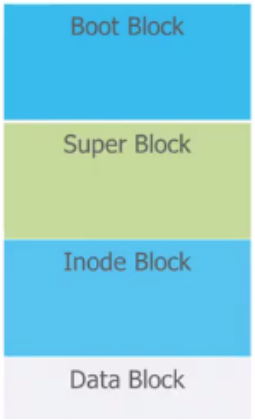
- **User application program:**
  - Utility programs and applications given by the user are handled in this layer.
  - Data in Unix are organized into files and files are organized into directories which are further organized into a tree-like structure called the file system.
- A file system consists a sequence of four blocks:

❑ **Boot Block:**

- It contains the boot strap code that is required to boot the system like Restart, Shutdown, Logoff, Switch user, etc.
- It occupies the beginning (first sector) of the Unix file system.

Cont….

- **Super Block:**
  - It contains administrative information about the entire Unix file system.
  - For instance–state of the file system, free data blocks, number of inodes, number of disk blocks, next available free blocks, next available index node, etc.
- **Inode Block:**
  - It stores the list of inodes for all the files in the file system.
  - Characteristics of the files are pointed here.
- **Data Block:**
  - All the files and administrative data are stored here.



Cont….

- **Unix Commands and Utilities:**
  - In Unix, a command is a program that can run.
  - To run a command in Unix, type its name and then press ***Enter*** key.
  - In MAC and Windows OS, point to the program that needs to run by clicking on it.
- **Rules of Unix Commands:**
  - Unix command may or may not have arguments. An argument can be an option or a filename.
  - The general format for a Unix command is **Command options(s) filename(s)**.
  - The option is usually preceded by a **minus (-)** sign.
  - Two or more options available with a command can be combined. **For instance: ls –l –a (or) ls –la**.
  - All Unix commands must be entered in small case letters.
  - There must be a space or tab between the Command name and options.
  - If there exists a typo error, use **Back Space** to erase the characters.

Cont….

- Types of Unix commands:
  - Simplex commands.
  - Complex commands.
  - Compound commands.
- Simplex commands:
  - A simple command is a command that can be executed by providing the command name in the command prompt.
  - For instance: $date, $cal, etc.
- Complex Commands:
  - Complex commands consists of a compound name with a list of arguments that together acts as a single command.
  - Ex: $who am i.
- Compound Commands:
  - A compound commands consists of a list of simple and/or complex commands separated by command operator, semi colon (;) which indicates where does a command end begin.
  - Ex: $date; whoamI; cal.
  - If the commands are executed without a command separator, an error message will be displayed.
  - Ex of an error message: Bad conversion.

Some important Commands:

- **Cal**: It displays a calendar. If arguments are not specified, the current month is displayed.
  - Syntax: $ cal [option]
  - -1: Displays the current month output only (default). Ex: cal -1,
  - -3: Displays previous, current and next month output,
  - -s: Display Sunday as the first day of the week (default),
  - -m: Display Monday as the first day of the week,
  - -j: Displays Julian dates (days one-based, numbered from January 1),
  - -y: Displays a calendar for the current year.

Cont….

- **date**: Used to display the date in different formats.
  - It also allows to set the system date and time.
- Various date command formats:
  - To display the date in various formats, multiple options can be used.
- Syntax: $ date +% [option].

Options	Significance/Description	Output
date +%d	Displays day of month (e.g. 01)	07
date +%D	Displays current date in MM/DD/YY format	02/07/13
date +%F	Displays current date in YYYY-MM-DD format	2013-02-07
date +%H	Displays hour in (00..23) format	23
date +%I	Displays hour (01..12) format	11
date +%j	Displays day of year (001..366)	038
date +%m	Displays month (01..12)	02
date +%M	Displays minute (00..59)	44
date +%S	Displays second (00..60)	17
date +%N	Displays nanoseconds (000000000..999999999)	573587606
date +%Y	Displays full year (e.g. YYYY)	2013
date +%Z	Displays alphabetic time zone abbreviation (e.g., PDT, IST, EDT)	IS

Cont….

- **history**: displays the history list of the commands with line numbers.
- **sleep**: It pauses for a period of time defined by a number.
- Ex: sleep 10: displays for 10 seconds to go to the next command or display the command prompt.
- Cal –1; sleep 10; cal –y.
- **passwd**: changes the password of the user.

Cont….

- **Clear**: Used to clear the contents of screen.
- **Ls**: Used to list the contents of the directories. For instance, $ ls [option].
    - ls –1: lists one file per line (default), Ex: ls –1.
    - –r: Lists file in reverse order, Ex: ls –r.
    - –s: Displays the file size in blocks, Ex: ls –s.
    - –u: Lists files according to file access time, Ex: ls–u.
    - –x: Displays files stored in rows across the screen, Ex: ls–x.
    - –R: Lists subdirectories recurvely.
- df: used to see how much disk is being used and what part of it lies free.
- du: Used to specify the disk space used by the specified files and directories in it.
- **Note**: There is space between ls and options in all commands.

Files and Directories:

- All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the **file system**.
- The File System: Unix file system is characterized by.
  - **A hierarchical (i.e. tree like) structure:**
    - The Unix file system is created like a tree structure having single root node, written as "/". The branches either end with leaf (i.e. no further division) or with non-leaf node (i.e. which further may divide or re-divide). Every non-leaf node is considered as directory. These directories further may contain sub-directories(non-leaf) or just files (leaf). The leaf nodes indicate either empty directories (which further in future may become non-leaf node by containing files in it, but at present they are empty) or regular files or special device files).
  - **Path Name:**
    - Path name is a sequence of component names, each separated by slash character. Using the path name, location a file in the file system hierarchy is found out.

Cont….

- To traverse a location of file in the file system, we can start the searching from the root of the directory (say /). 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 begin with root.
- **Relative path name** –
  - If you 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.

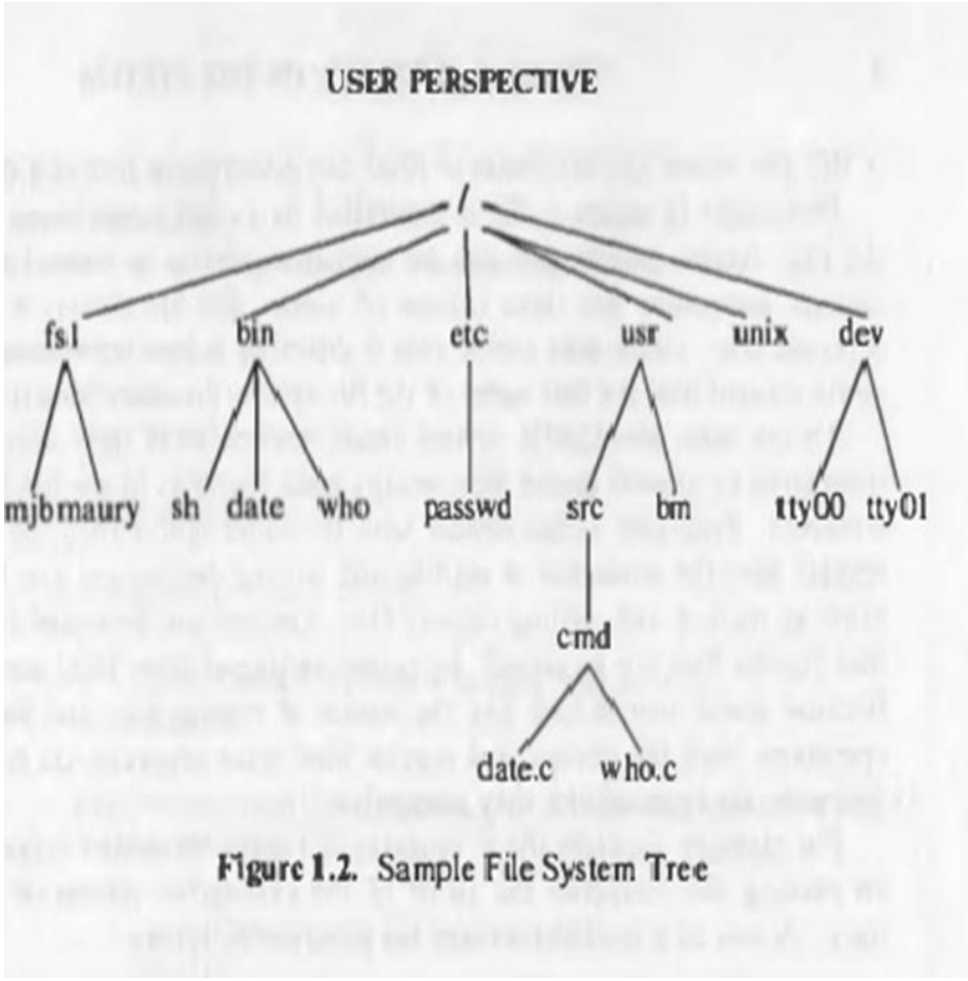
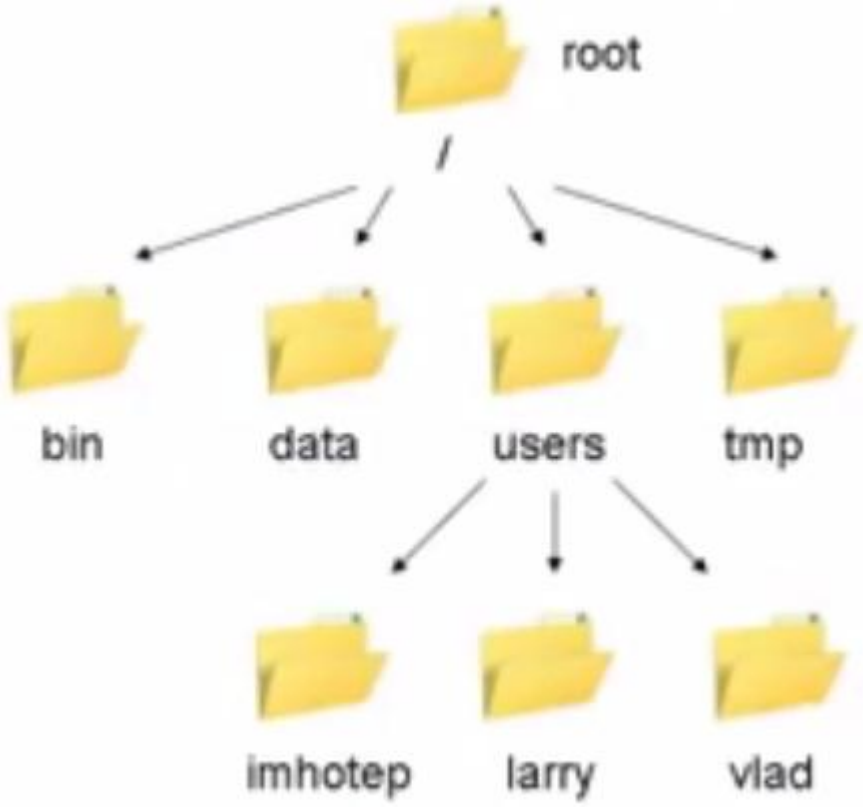


Figure 1.2. Sample File System Tree

Cont….

- **Predefined directories:**
  - **bin:** This directory contains executable files for most of the Unix commands. Unix commands can be either C programs or Shell programs.
  - **lib:** It contains all the library functions provided by Unix for users.
  - **dev:** It contains files that control various input/output devices like terminals, printer, disk drivers etc. for each device there is a separate file.
  - **usr:** In the usr directory there are several directories, each associated with a particular user. Within the usr directory there is another **bin** directory which contains additional Unix command files.
  - **tmp:** The tmp directory contains the directory files created by Unix or by the users. The **etc** contains binary executable files usually required for system administration.



Cont….

MS-Windows	Unix
Folder	Directory
Administrator	Root user/super user
File	File
Software	Package

References:

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

Thank You !!!