UNIX

Lesson 02 : UNIX file system

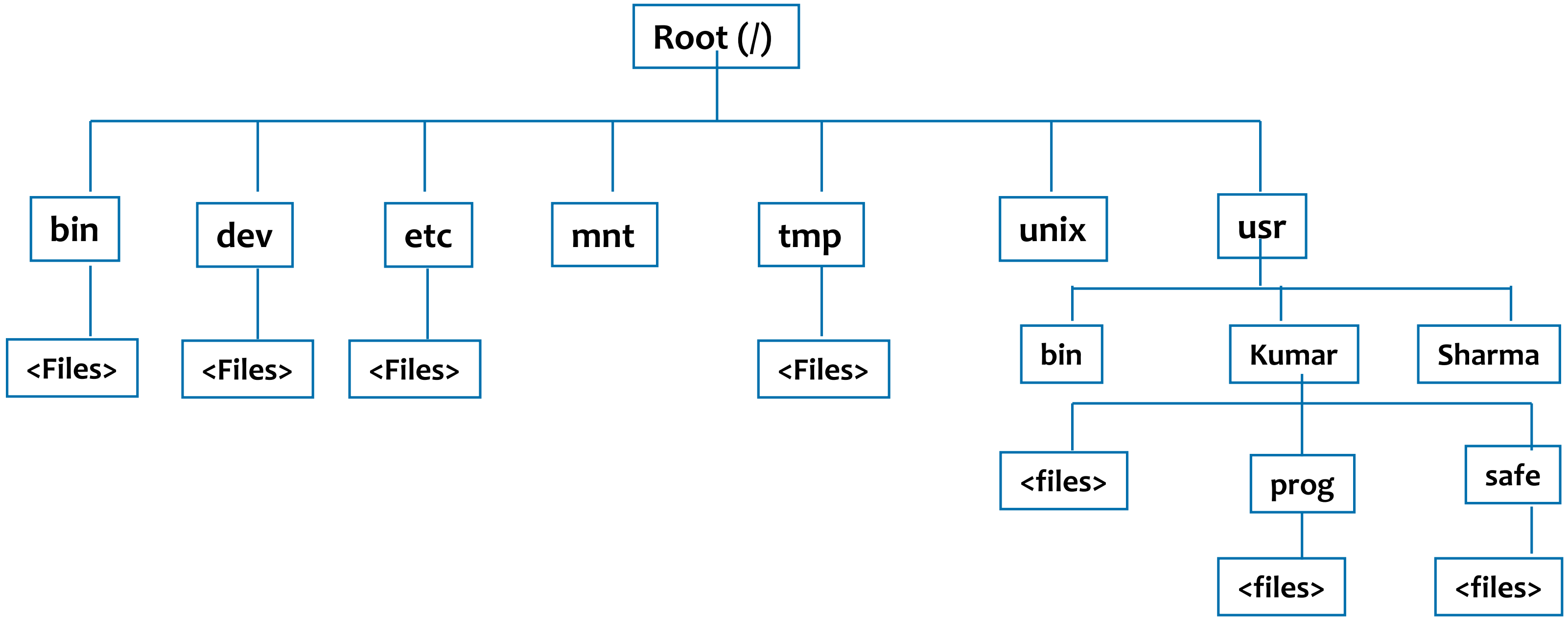
Lesson Objectives

- In this lesson, you will learn:
  - File System
  - Basic UNIX Commands

2.1: File System
Features of UNIX File system

- All of the files in the UNIX file system are organized into a multi-leveled hierarchy called a directory tree.
- A family tree is an example of a hierarchical structure that represents how the UNIX file system is organized. The UNIX file system might also be envisioned as an inverted tree or the root system of plant.
- At the very top of the file system is single directory called "root" which is represented by a / (slash). All other files are "descendents" of root.
- The number of levels is largely arbitrary, although most UNIX systems share some organizational similarities.

File System Structure



File System Structure

/ bin : commonly used UNIX Commands like who, ls

/usr/bin : cat, wc etc. are stored here

/dev : contains device files of all hardware devices

/etc : contains those utilities mostly used by system administrator

- Example: passwd, chmod, chown

File System

/tmp : used by some UNIX utilities especially vi and by user to store temporary files

/usr : contains all the files created by user, including login directory

/unix : kernel

Release V:

- It does not contain / bin.
- It contains / home instead of /usr.

2.2: File Types
File types

- Ordinary Files
- Directory
- Special Files
- Pipes

File types

- Ordinary Files
  - Used to store your information, such as some text you have written or an image you have drawn. This is the type of file that you usually work with.
  - Always located within/under a directory file
  - Do not contain other files

File types

- Directories
  - Branching points in the hierarchical tree
  - Used to organize groups of files
  - May contain ordinary files, special files or other directories
  - Never contain "real" information which you would work with (such as text). Basically, just used for orgaanizing files.
  - All files are descendants of the root directory, (named /) located at the top of the tree.

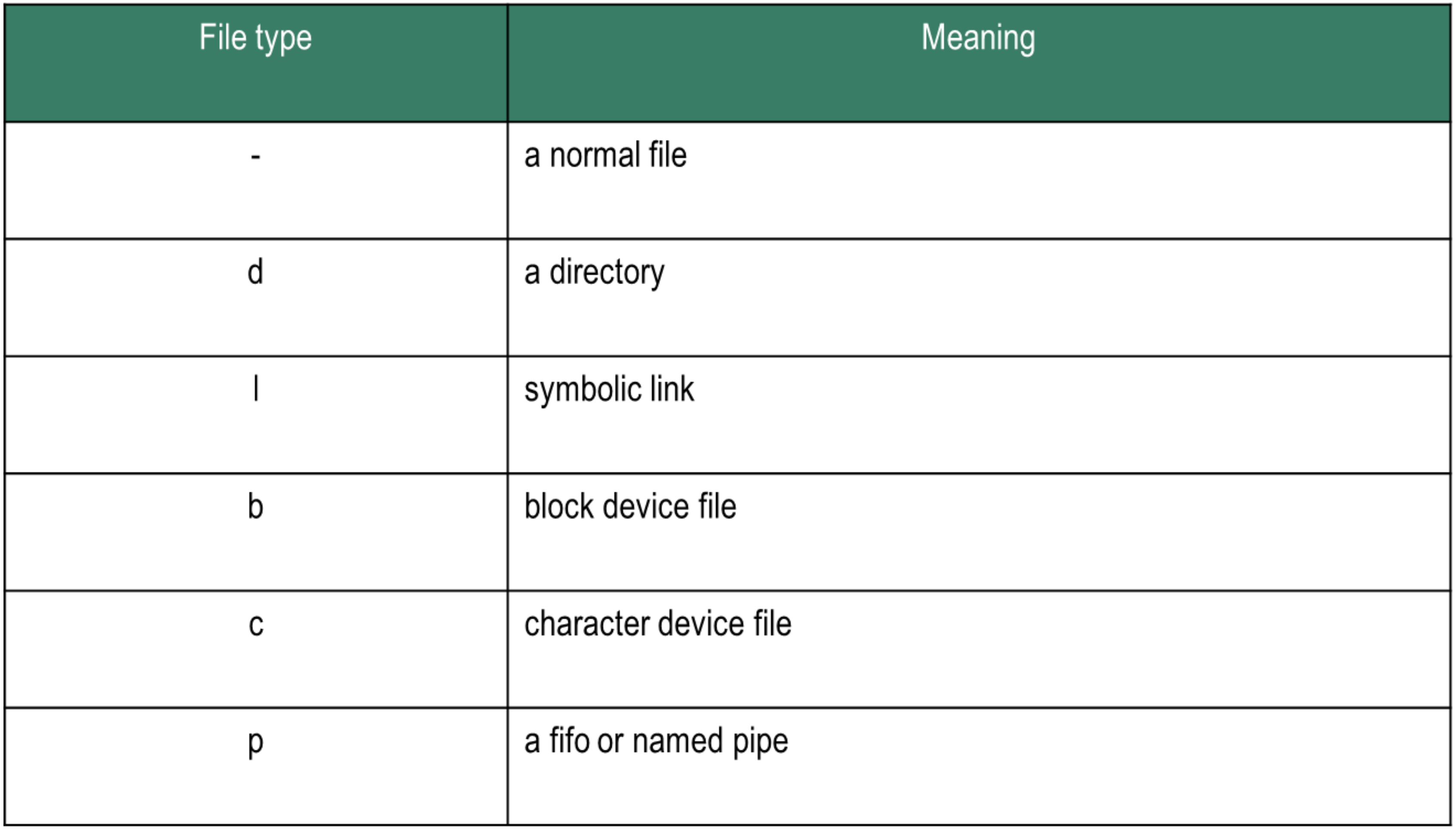
File types

- Special Files
  - Used to represent a real physical device such as a printer, tape drive or terminal, used for Input/Ouput (I/O) operations
  - Unix considers any device attached to the system to be a file - including your terminal:
    - By default, a command treats your terminal as the standard input file (stdin) from which to read its input
    - Your terminal is also treated as the standard output file (stdout) to which a command's output is sent
    - Stdin and stdout
  - Two types of I/O: character and block
  - Usually only found under directories named /dev

File types

- Pipes
  - UNIX allows you to link commands together using a pipe. The pipe acts a temporary file which only exists to hold data from one command until it is read by another

File types



File type	Meaning
-	a normal file
d	a directory
l	symbolic link
b	block device file
c	character device file
p	a fifo or named pipe

File Names

- UNIX permits file names to use most characters, but avoid spaces, tabs and characters that have a special meaning to the shell, such as: **& ; ( ) | ? \ ' " ` [ ] { } < > $ - ! /**
- Case Sensitivity: uppercase and lowercase are not the same! These are three different files: **NOVEMBER November november**
- Length: can be up to 256 characters
- Extensions: may be used to identify types of files
  - **libc.a - *archive, library file***
  - **program.c - *C language source file***
  - **alpha2.f - *Fortran source file***
  - **xwd2ps.o - *Object/executable code***
  - **mygames.Z - *Compressed file***

File Names

- Hidden Files: have names that begin with a dot (.) For example: **.cshrc .login .mailrc .mwmrc**
- Uniqueness: as children in a family, no two files with the same parent directory can have the same name. Files located in separate directories can have identical names.
- Reserved Filenames:
  - / - the root directory (slash)
  - . - current directory (period)
  - .. - parent directory (double period)
  - ~ - your home directory (tilde)

2.3: File Path

Pathnames

- Specify where a file is located in the hierarchically organized file system
- **Must know how to use pathnames to navigate the UNIX file system**
- Absolute Pathname: tells how to reach a file begining from the root; always begins with / (slash). For example: /usr/local/doc/training/sample.f
- **Relative Pathname:** tells how to reach a file from the directory you are currently in (current or working directory); never begins with / (slash). For example:
  - training/sample.f
  - ../bin
  - ~/projects/report.001
- For example, if your current directory is /usr/home/johnson and you wanted to change to the directory /usr/home/quattro, you could use either of these commands:
  - cd ../quattro - *relative pathname*
  - cd /usr/home/quattro - *absolute pathname*

SUMMARY

- In this lesson, you have learnt:
  - UNIX organizes files in hierarchical manner.
  - File types
  - Relative and absolute path

Review Questions

❖What is relative path?

❖State True or False:

- **/bin stores temporary file?**
- **/etc will have system configuration files**