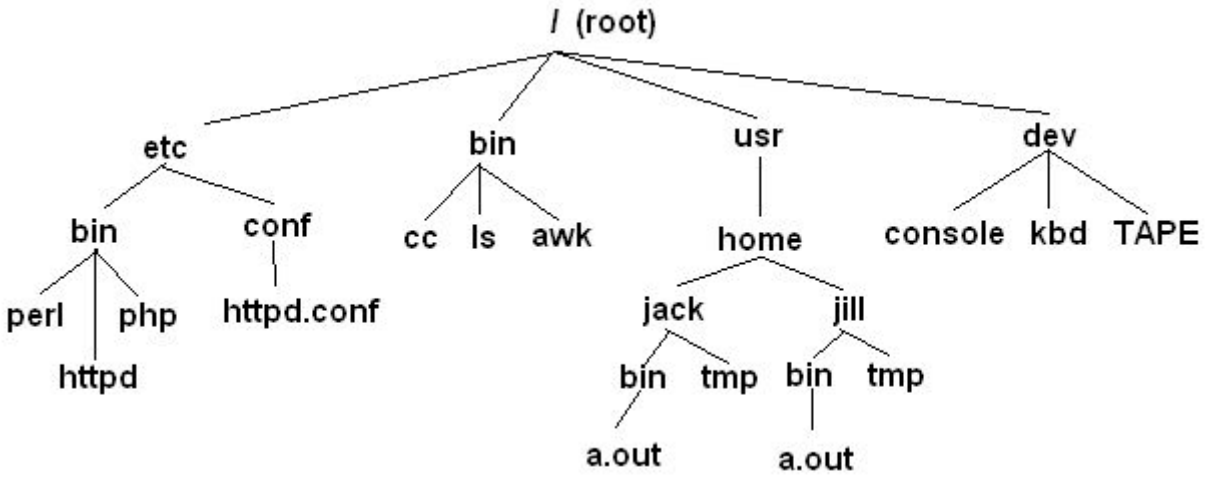
2.6 Special Characters in Files

- When DOS files are opened in *text* mode, the internal separator ('\n') is translated to the the external separator (<CR><LF>) during read and write.CR-carriage return, LF-Line feed
- When DOS files are opened in *binary* mode, the internal separator ('\n') is **not** translated to the the external separator (<CR><LF>) during read and write.
- In DOS (Windows) files, end-of-file can be marked by a "control-Z" character (ASCII *SUB*).
- In C++ implementations for DOS, a control-Z in a file is interpreted as end-of-file.

2.7 The Unix Directory Structure

- The Unix file system is a tree-structured organization of directories,with the root of the tree signified by the character /.
- In UNIX, the directory structure is a single tree for the entire file system.



- In UNIX, separate disks appear as subdirectories of the root (/).
- In UNIX, the subdirectories of a pathname are separated by the forward slash character (/).
- Example: /usr/bin/perl
- The directory structure of UNIX is actually a graph, since symbolic links allow entries to appear at more than one location in the directory structure.

2.8 Physical Devices and Logical Files

2.8.1 Physical devices as files

In unix, devices like keyboard and console are also files. The keyboard produces a sequence of bytes that are sent to the computer when keys are pressed. The console accepts a sequence of bytes and displays the symbols on screen.

A Unix file is represented logically by an integer-the file descriptor

A keyboard, a disk file, and a magnetic tape are all represented by integers.

This view of a file in Unix makes it possible to do with a very few operations compared to other OS.

2.8.2 The console, the keyboard and standard error

In C streams, the keyboard is called stdin(standard input),console is called stdout(standard output) error file is called stderr(standard error).

Handle	FILE	iostream	Description
0	stdin	Cin	Standard Input
1	stdout	Cout	Standard Output
2	stderr	Cerr	Standard Error

2.8.3 I/O redirection and Pipes

Operating systems provide shortcuts for switching between standard I/O(stdin and stdout) and regular file I/O

I/O redirection is used to change a program so it writes its output to a regular file rather than to stdout.

- In both DOS and UNIX, the standard output of a program can be redirected to a file with the > symbol.
- In both DOS and UNIX, the standard input of a program can be redirected to a file with the < symbol.

The notations for input and output redirection on the command line in Unix are

```
< file                (redirect stdin to "file")
> file                (redirect stdout to "file")
```

Example:

```
list.exe > myfile
```

The output of the executable file is redirected to a file called “myfile”

pipe

Piping: using the output of one program as input to another program.

A connection between standard output of one process and standard input of a second process.

- In both DOS and UNIX, the standard output of one program can be piped (connected) to the standard input of another program with the | symbol.
- Example:

```
program1 | program2
```

Output of program1 is used as input for program2

2.9 File-Related Header Files

- Header files can vary with the C++ implementation.

  Stdio.h, iostream.h, fstream.h, fcntl.h and file.h are some of the header files used in different operating systems

2.10 Unix File System Commands

UNIX	**Description**
cat *filename*	Type the contents of a file
tail *filename*	Type the last ten lines of a file
cp *file1 file2*	Copy file1 to file2
mv *file1 file2*	Move(rename) file1 to file2
rm *filenames*	Delete files
chmod *mode filename*	Change the protection mode
ls	List contents of a directory
mkdir	Create directory
rmdir	Remove directory