The UNIX File System

Introduction

- What is File System?
  - The abstraction used by kernel to represent and organize the storage resources.
- UNIX File System in general
  - File system is organized in tree structure.
  - File tree can be arbitrarily deep.
  - File name must NOT LONGER than 256 chars.
  - Single path name must NOT LONGER than 1023 chars.

Creating File System

- Mounting File System
  - File tree is composed of File System
  - Use *mount* command to map a directory within the existing file tree (mount point) to the root of the new file system.
    - *mount /dev/hda2 /usr*
  - Use *umount* command to detach the file system.
    - Detaching will fail if the file system is busy.

Organizing of The File System

- “The UNIX file system has never been very well organized.” -- Page 58
  - – incompatible naming convention
    - e.g. ATT & BSD startup script naming
    - e.g. log file naming

Organizing of The File System (cont.)

/	
/bin or /sbin	Commands for basic system operation
/dev	Device entries
/etc	Critical startup and configuration files.
/lib	Library for the C compiler
/tmp	Temporary files
/var/adm or /var/log	Accounting file, log files

Types of Files

- Regular Files
  - – binary
    - GIF, JPEG, Executable etc.
  - – text
    - scripts, program source code, documentation
  - – Supports sequential and random access

Types of Files (cont.)

- Directory
  - Can contain ANY kind of files
  - what is “.” and “..”??
- Device File
  - Allows programs to communicate with hardware.
  - Kernel modules handles device management.

Types of Files (cont.)

- Device Files (cont.)
  - Character Device
    - Accepts a stream of characters, without regard to any block structure.
    - It is not addressable, therefore no seek operation
  - Block Device
    - Information stored in fixed-sized block
    - It is addressable, therefore seek operation is possible.

Types of Files (cont.)

- UNIX Domain Sockets (BSD)
  - sockets that are local to a particular host and are referenced through a file system object rather than a network port.
  - X windows
- Named Pipe
  - Allow processes to communicate with each other.

Types of Files (cont.)

- Hard links
  - – Linking files by reference
  - – System maintains a count of the number of links
  - – Does not work across file systems.
- Soft links
  - – Linking files by name
  - – No counter is maintained
  - – Work across file system

File Permissions

- The Setuid and Setgid bits
  - Setuid with octal value 4000
  - Setgid with octal value 2000
  - These bits allow programs to access files that processes that would otherwise off limits to the user that runs them.

Types of Files (cont.)

- Sticky Bit
  - Not very popular in today’s system
  - If a directory has sticky bit set, then only the owner can remove file from the directory.
  - /tmp is a good example.

Types of Files (cont.)

- The Permission Bit
  - – 9 permission bits used to determine 3 types of accesses, READ, WRITE, EXECUTE.
  - – Permission can be set based on GROUP, OWNER, ANYONE ELSE.
  - – Use *chmod* command to change permission
    - Binary 001 for EXECUTE
    - Binary 010 for WRITE
    - Binary 100 for READ

Types of Files (cont.)

- INODES
  - Kernel maintains file information in a structure called *inode.*
    - Creation, modification time stamps
    - Ownership, file size etc.
  - Commonly used INODE information can be found by using *ls* command
  - Group information and be modified by using *chgrp* command.

Summary

- All UNIX file system are very similar.
- All file system have this concept of file tree.
  - Transparent to user even mount point is mapped to a remote file system.
- To communicate with devices, special device files are used.
- More information check out the man pages.

Thank You