Introduction to the File System

- Ch. 10 - Silberschatz, Galvin, Gagnes , "Operating System Concepts"
- Ch. 11 - William Stallings, "Operating System-Internals and Design Principles "

Contents

- File concepts
- File Attributes and operations
- File Types and sharing
- File Structure
- File system mounting and un-mounting
- Directory Overview and types
- Types of users
- Access modes/Permissions
- Case Study – Unix File Structure

Files

- Computers can store information on various storage media, such as magnetic disks, magnetic tapes, and optical disks.
- The operating system provides a uniform logical view of information storage.
- The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, ***the file.***
- A file is a named collection of related information.
- Collections of files are grouped into directories (i.e. Folders).
- A directory is itself a file

File

- It represent programs and data.
- Its a sequence of bits, bytes, lines, or records, the meaning of which is defined by the file's creator and user.
- A file is named, for convenience, and is referred to by its name.
- A file has a certain defined structure which depends on its type.
  - *Text file* is a sequence of characters,
  - *Source file* is a sequence of subroutines and Functions,
  - *Object file* is a sequence of bytes,
  - *Executable file* is a series of code

File Concepts

- Contiguous logical address space
- Types:
  - Data
    - Numeric
    - Character
    - Binary
  - Program

File Structure

- None - sequence of words, bytes
- Simple record structure
  - Lines
  - Fixed length
  - Variable length
- Complex Structures
  - Formatted document
  - Relocatable load file
- Can simulate last two with first method by inserting appropriate control characters.
- Who decides: Operating system / Program

File Attributes

- File specific information maintained by the operating system.
- File attributes are meta-data associated with computer files that define file system behavior.
- Each attribute can have one of two states: set and cleared.
- Information about files are kept in the directory structure, which is maintained on the disk.

File Attributes

- **Name** – only information kept in human-readable form.
- **Type** – needed for systems that support different types.
- **Location** – pointer to file location on device.
- **Size** – current file size.
- **Protection** – controls who can do reading, writing, executing.
- **Time, date, and user identification** – data for protection, security, and usage monitoring.
- Information about files are kept in the directory structure, which is maintained on the disk.

File Operations

- Create
- Write
- Read
- reposition within file – file seek
- Delete
- Truncate
- open(F_i) – search the directory structure on disk for entry F_i, and move the content of entry to memory.
- close (F_i) – move the content of entry F_i in memory to directory structure on disk.

File Operations

- Open, Close
  - Gain or relinquish access to a file
  - OS returns a file handle – an internal data structure letting it cache internal information needed for efficient file access
- Read, Write, Truncate
  - Read: return a sequence of n bytes from file
  - Write: replace n bytes in file, and/or append to end
  - Truncate: throw away all but the first n bytes of file
- Seek, Tell
  - Seek: reposition file pointer for subsequent reads and writes
  - Tell: get current file pointer
- Create, Delete:
  - Conjure up a new file; or blow away an existing one

File Types – name, extension

- Most operating systems recognize file types
  - Filename extension
  - I.e. resume.doc, server.java, readerthread.c
- Most support them
  - Automatically open a type of file via a specific application (.doc)
  - Only execute files of a given extension (.exe, .com)
  - Run files of a given type via a scripting language (.bat)
- Can get more advanced
  - If source code modified since executable compiled, if attempt made to execute, recompile and then execute (TOPS-20)
  - Mac OS encodes creating program's name in file attributes
    - Double clicking on file passes the file name to appropriate application
  - Unix has magic number stored in file at first byte indicating file type

File Types – name, extension

File Type	Extension	Function
Executable	exe, com, bin or none	ready-to-run machine-language program
Object	obj, o	complied, machine language, not linked
Source code	c, p, pas, 177, asm, a	source code in various languages
Batch	bat, sh	commands to the command interpreter
Text	txt, doc	textual data documents
Word processor	wp, tex, rrf, etc.	various word-processor formats
Library	lib, a	libraries of routines
Print or view	ps, dvi, gif	ASCII or binary file
Archive	arc, zip, tar	related files grouped into one file, sometimes compressed.

[Chhaya@localhost ~]$ cat a.c

```
#include<stdio.h>
int main()
{
 int a[10],b,c,i;
 b= 10;

 for(i=0;i<10;i++)
 {
   a[i]= b+i;;

 }
 b= c+b;
}
```

```
[Chhaya@localhost ~]$ xxd -b a.c
0000000: 00100011 01101001 01101110 01100011 01101100 01110101  #inclu
0000006: 01100100 01100101 00111100 01110011 01110100 01100100  de<std
000000c: 01101001 01101111 00101110 01101000 00111110 00001010  io.h>.
0000012: 01101001 01101110 01110100 00100000 01101101 01100001  int ma
0000018: 01101001 01101110 00101000 00101001 00001010 01111011  in().{
000001e: 00001010 00100000 01101001 01101110 01110100 00100000  . int
0000024: 01100001 01011011 00110001 00110000 01011101 00101100  a[10],
000002a: 01100010 00101100 01100011 00101100 01101001 00111011  b,c,i;
0000030: 00001010 00100000 01100010 00111101 00100000 00110001  . b= 1
0000036: 00110000 00111011 00001010 00100000 00001010 00100000  0;. .
000003c: 01100110 01101111 01110010 00101000 01101001 00111101  for(i=
0000042: 00110000 00111011 01101001 00111100 00110001 00110000  0;i<10
0000048: 00111011 01101001 00101011 00101011 00101001 00001010  ;i++).
000004e: 00100000 01111011 00001010 00100000 00100000 00100000   {.
0000054: 01100001 01011011 01101001 01011101 00111101 00100000  a[i]=
000005a: 01100010 00101011 01101001 00111011 00111011 00001010  b+i;;.
0000060: 00100000 00100000 00100000 00001010 00100000 01111101     . }
0000066: 00001010 00100000 01100010 00111101 00100000 01100011  . b= c
000006c: 00101011 01100010 00111011 00100000 00001010 01111101  +b; .}
0000072: 00001010 00100000 00100000 00100000 00001010           .   .
[Chhaya@localhost ~]$
```