LUT University

LAND OF THE CURIOUS

JANUARY 17, 2023

OPERATING SYSTEMS AND SYSTEMS PROGRAMMING (CT30A3370) 6 CREDITS

Venkata Marella

CHAPTER 10: FILE-SYSTEM INTERFACE

- File Concept
- Access Methods
- Directory Structure
- File-System Mounting
- File Sharing
- Protection

OBJECTIVES

- □ To explain the function of file systems
- □ To describe the interfaces to file systems
- □ To discuss file-system design tradeoffs, including access methods, file sharing, file locking, and directory structures
- □ To explore file-system protection

Why tradeoffs?

Too few structures: programming inconvenient; Too many structures: OS bloat & programmer confused.

FILE CONCEPT

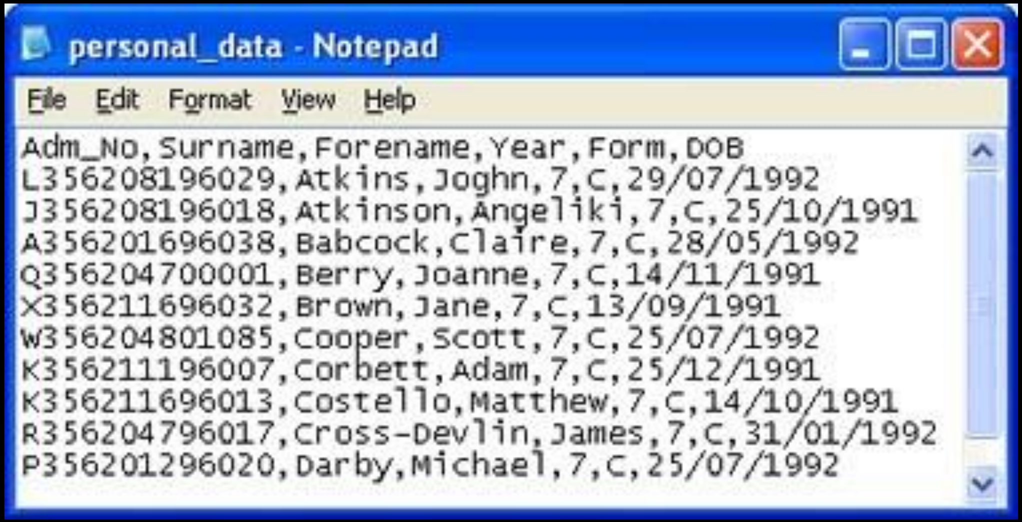
- Contiguous logical address space
- A sequence of bits, bytes, lines, or records. The meaning is defined by the creator and user.
- Types:
  - Data
    - numeric
    - character
    - binary
  - Program
    - Source
    - Object
    - Executable

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

CSV File

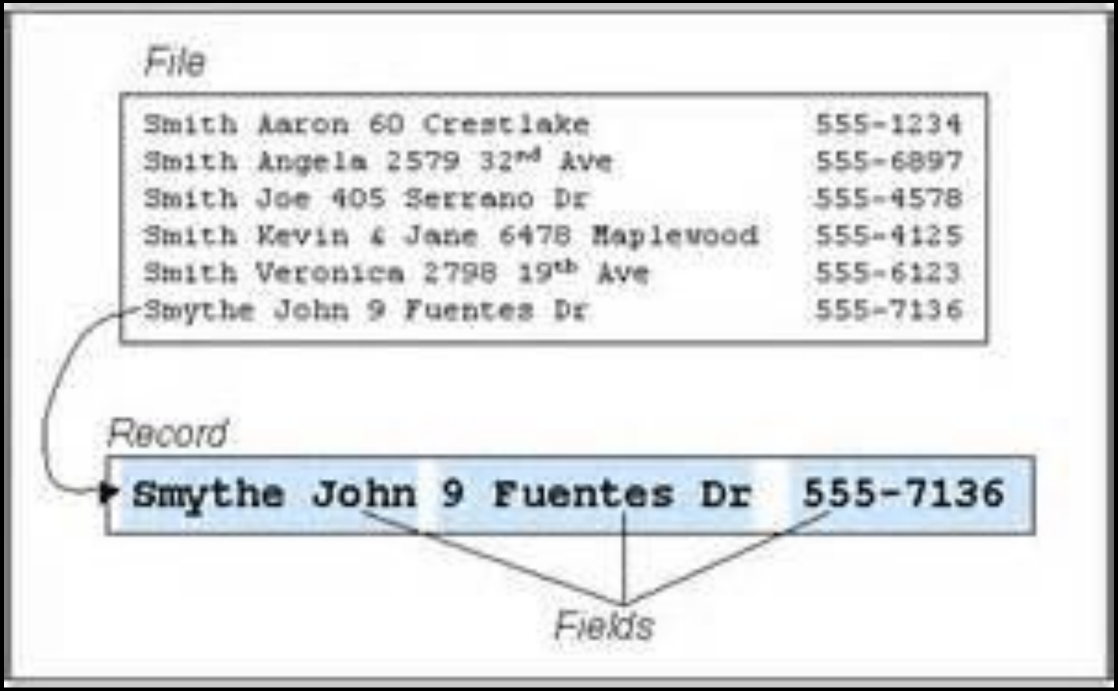
- line-based file structure



```
Adm_No,Surname,Forename,Year,Form,DOB
L356208196029,Atkins,Joghn,7,C,29/07/1992
J356208196018,Atkinson,Angeliki,7,C,25/10/1991
A356201696038,Babcock,Claire,7,C,28/05/1992
Q356204700001,Berry,Joanne,7,C,14/11/1991
X356211696032,Brown,Jane,7,C,13/09/1991
W356204801085,Cooper,Scott,7,C,25/07/1992
K356211196007,Corbett,Adam,7,C,25/12/1991
K356211696013,Costello,Matthew,7,C,14/10/1991
R356204796017,Cross-Devlin,James,7,C,31/01/1992
P356201296020,Darby,Michael,7,C,25/07/1992
```

Database File

- Fixed Length



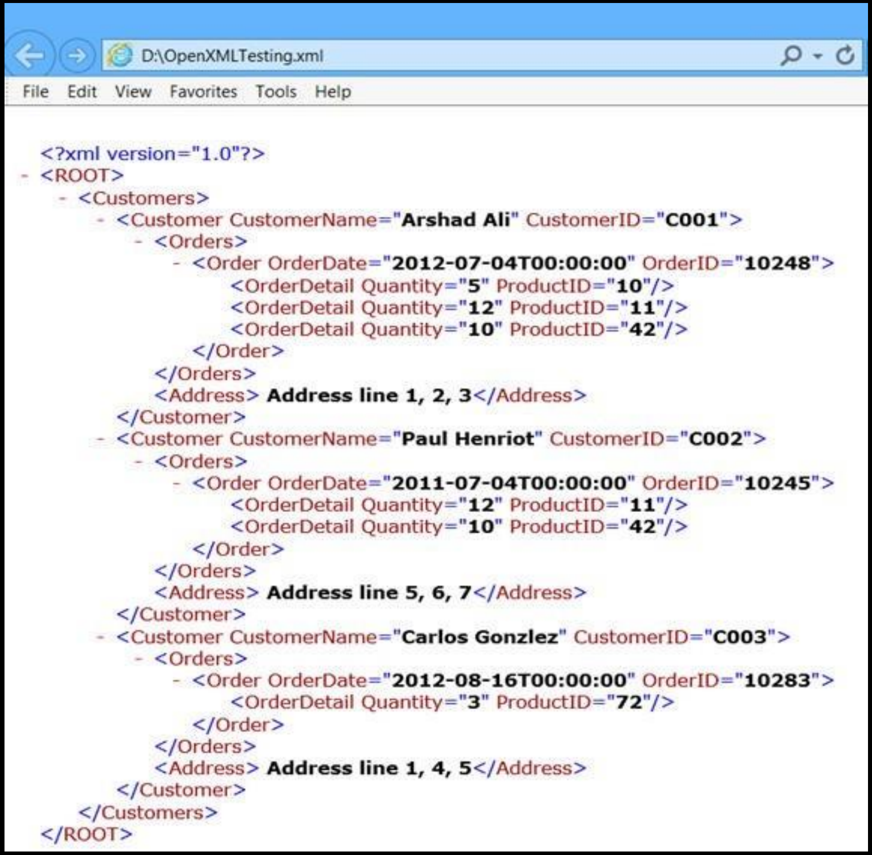
```
Smith Aaron 60 Crestlake                   555-1234
Smith Angela 2579 32nd Ave                 555-6897
Smith Joe 405 Serrano Dr                   555-4578
Smith Kevin & Jane 6478 Maplewood          555-4125
Smith Veronica 2798 19th Ave               555-6123
Smythe John 9 Fuentes Dr                   555-7136
```

```
Smythe John 9 Fuentes Dr   555-7136
```

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

Xml File

- formated structure



```
  <?xml version="1.0"?>
- <ROOT>
  - <Customers>
    - <Customer CustomerName="Arshad Ali" CustomerID="C001">
      - <Orders>
        - <Order OrderDate="2012-07-04T00:00:00" OrderID="10248">
            <OrderDetail Quantity="5" ProductID="10"/>
            <OrderDetail Quantity="12" ProductID="11"/>
            <OrderDetail Quantity="10" ProductID="42"/>
          </Order>
        </Orders>
        <Address> Address line 1, 2, 3</Address>
      </Customer>
    - <Customer CustomerName="Paul Henriot" CustomerID="C002">
      - <Orders>
        - <Order OrderDate="2011-07-04T00:00:00" OrderID="10245">
            <OrderDetail Quantity="12" ProductID="11"/>
            <OrderDetail Quantity="10" ProductID="42"/>
          </Order>
        </Orders>
        <Address> Address line 5, 6, 7</Address>
      </Customer>
    - <Customer CustomerName="Carlos Gonzlez" CustomerID="C003">
      - <Orders>
        - <Order OrderDate="2012-08-16T00:00:00" OrderID="10283">
            <OrderDetail Quantity="3" ProductID="72"/>
          </Order>
        </Orders>
        <Address> Address line 1, 4, 5</Address>
      </Customer>
    </Customers>
  </ROOT>
```

FILE ATTRIBUTES

- □ **Name** – only information kept in human-readable form
- □ **Identifier** – unique tag (number) identifies file within file system
- □ **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

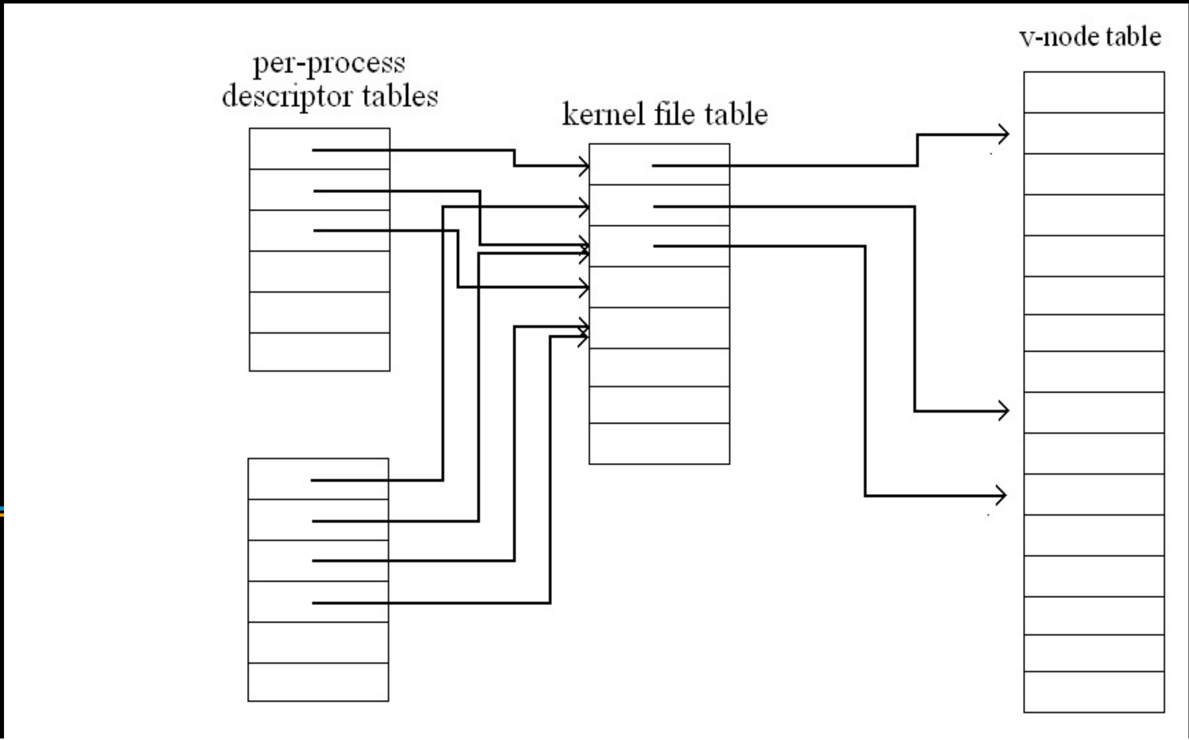
- File is an **abstract data type**
- **Create**
- **Write –** define a pointer
- **Read –** use the same pointer

  Per-process **current file-position pointer**
- **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

```
Class File{
Public:
    Create();
    Write();
    Read();
    Seek();
    ……
}
```

OPEN-FILE TABLE

- Open() system call returns a pointer to an entry in the open-file table
- File Control Block, FCB, (per file) containing details about ownership, size, permissions, dates, etc.
- Per-process table
  - Current file pointer
  - Access rights
  - ...
- System-wide table
  - Open count
  - ... http://cs.oberlin.edu



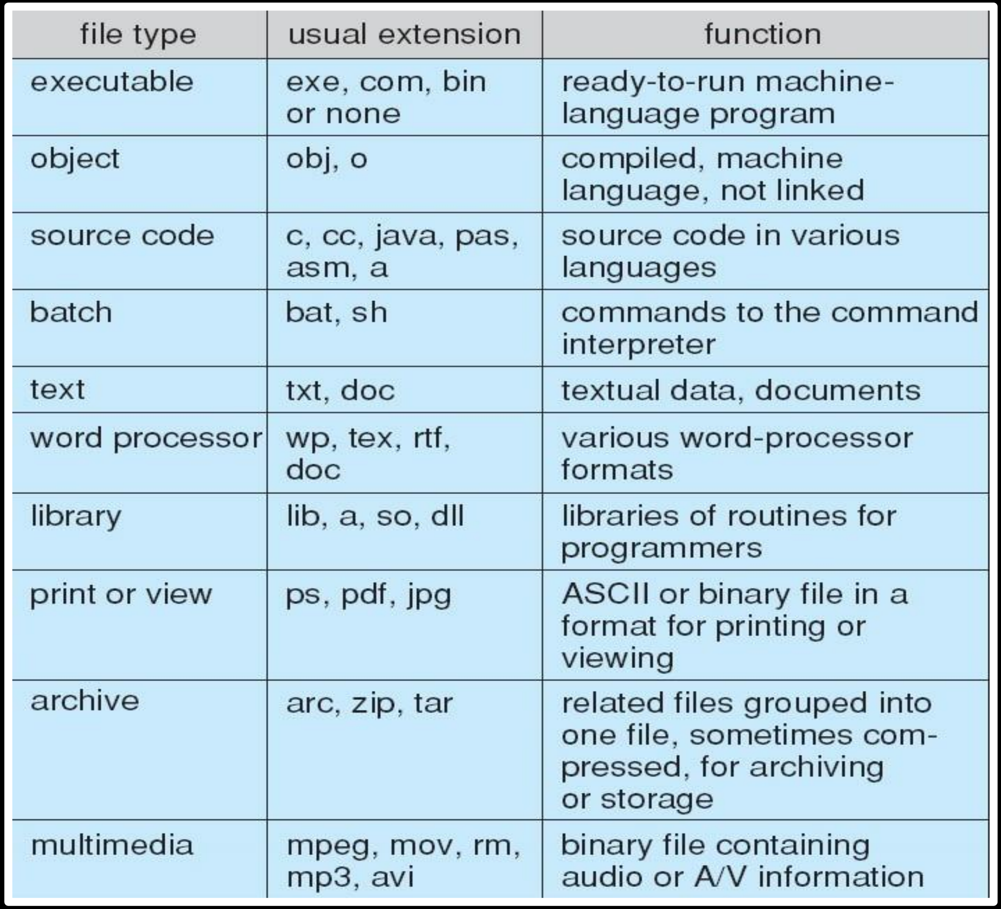
OPEN FILES

- Several pieces of data are needed to manage open files:
  - File pointer: pointer to last read/write location, per process that has the file open
  - File-open count: counter of number of times a file is open – to allow removal of data from open-file table when last processes closes it
  - Disk location of the file: cache of data access information – system doesn’t need to read it from disk for every operation.
  - Access rights: per-process access mode information

OPEN FILE LOCKING

- Provided by some operating systems and file systems
- Mediates access to a file (by multiple processes)
- File locks are similar to reader-writer locks
  - Shared lock (reader)
  - Exclusive lock (writer)
- Mandatory or advisory:
  - **Mandatory** – access is denied depending on locks held and requested
  - **Advisory** – processes can find status of locks and decide what to do

FILE TYPES – NAME, EXTENSION

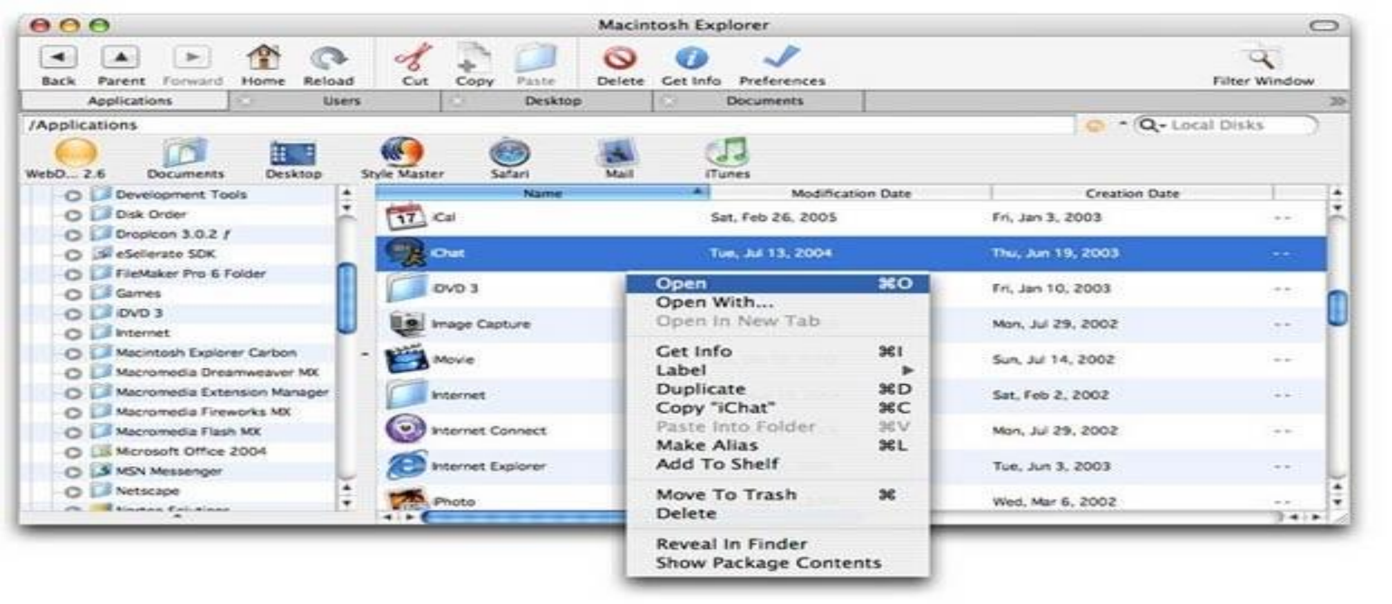


file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

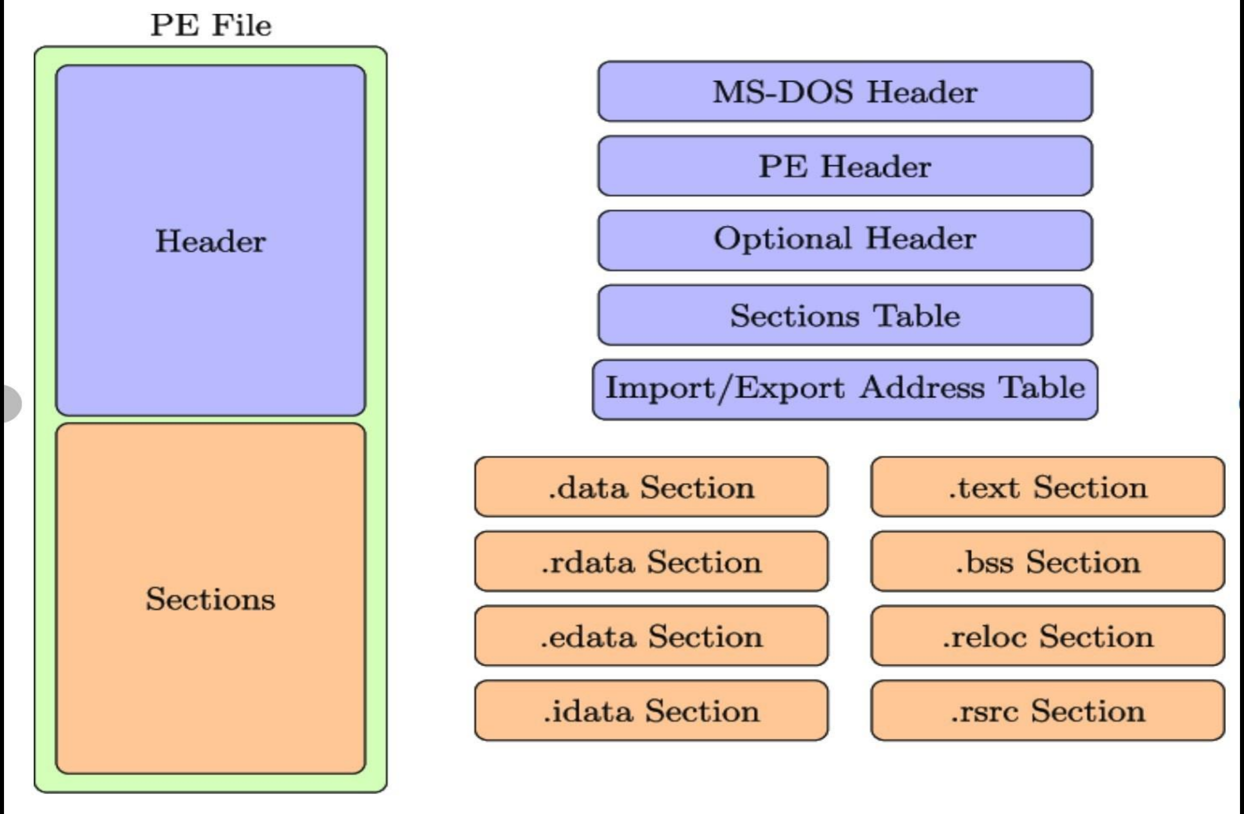
FILE TYPES

- MS-DOS
- MAC OS X
  - Each file has a creator attribute containing the name of the program that created it.
- UNIX
  - Magic number (executable, shell script, postscript, …)



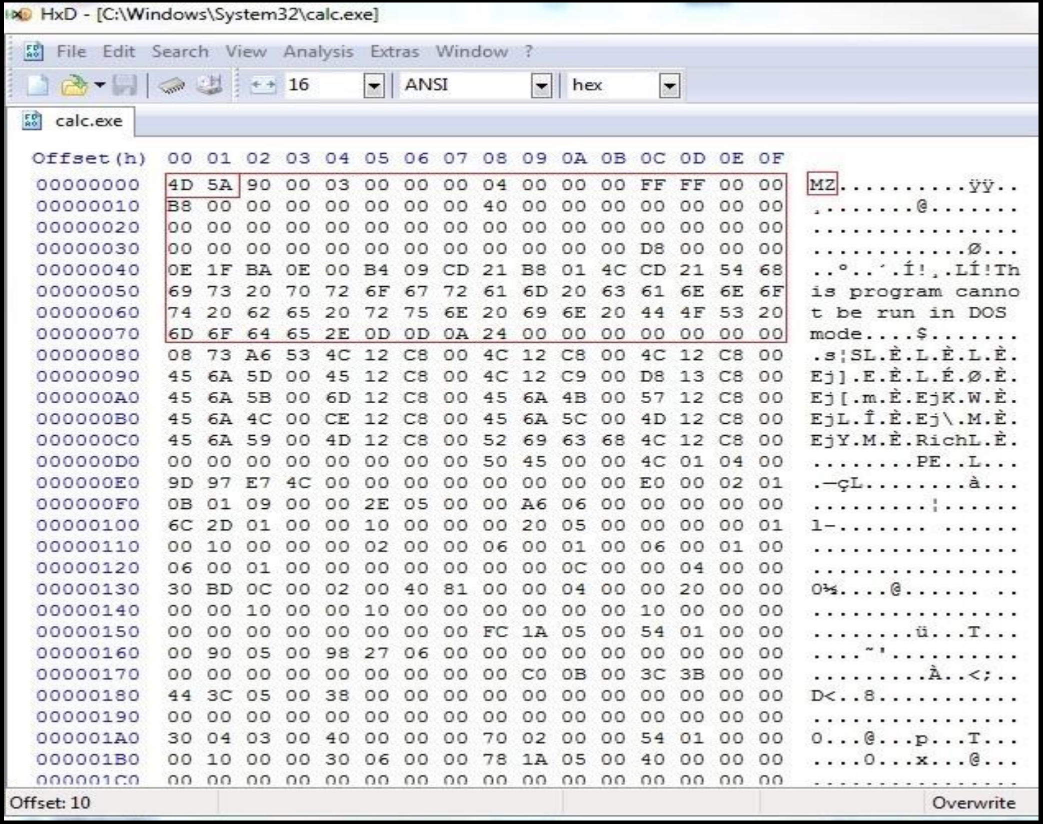
Name	Modification Date	Creation Date	
iCal	Sat, Feb 26, 2005	Fri, Jan 3, 2003	- -
iChat	Tue, Jul 13, 2004	Thu, Jun 19, 2003	- -
iDVD 3		Fri, Jan 10, 2003	- -
Image Capture		Mon, Jul 29, 2002	- -
iMovie		Sun, Jul 14, 2002	- -
Internet		Sat, Feb 2, 2002	- -
Internet Connect		Mon, Jul 29, 2002	- -
Internet Explorer		Tue, Jun 3, 2003	- -
iPhoto		Wed, Mar 6, 2002	- -

FILE TYPES: PORTABLE EXECUTABLE FILE FORMAT



Portable executable file format.

FILE TYPES: MAGIC BYTES



16 ANSI hex

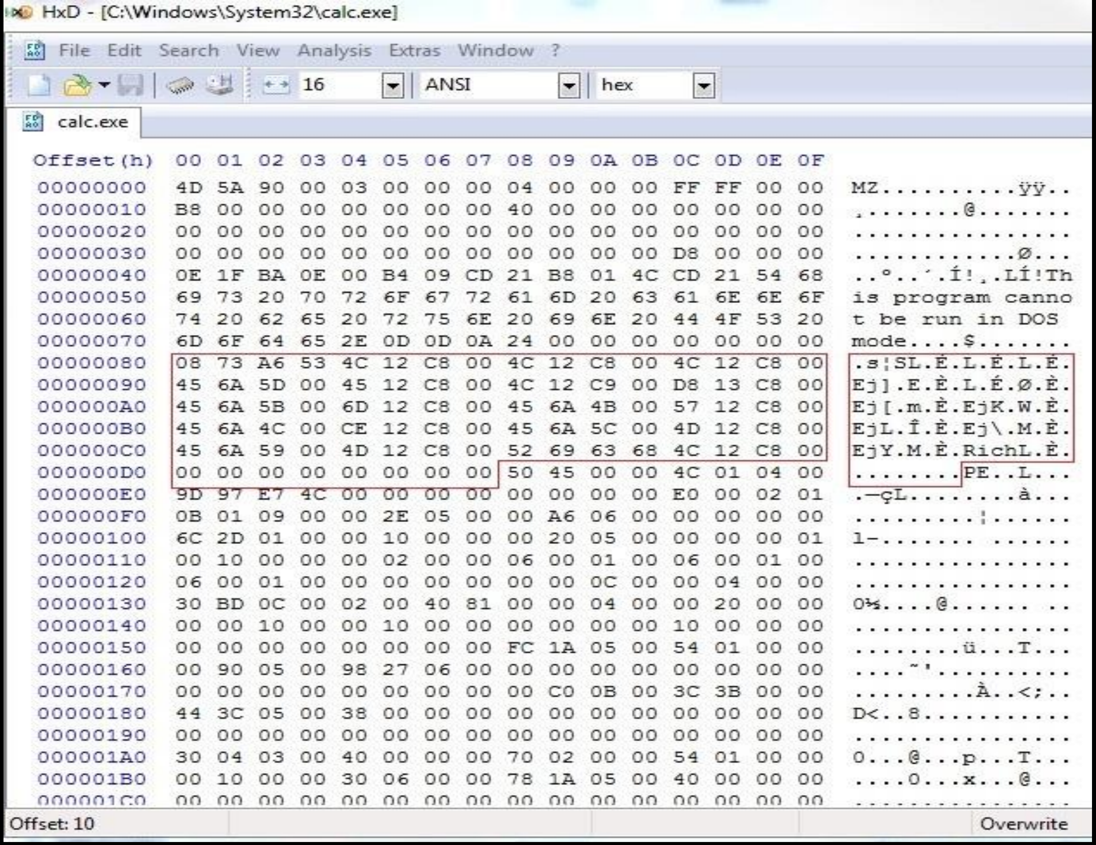
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 D8 00 00 00  ............Ø...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  08 73 A6 53 4C 12 C8 00 4C 12 C8 00 4C 12 C8 00  .s¦SL.È.L.È.L.È.
00000090  45 6A 5D 00 45 12 C8 00 4C 12 C9 00 D8 13 C8 00  Ej].E.È.L.É.Ø.È.
000000A0  45 6A 5B 00 6D 12 C8 00 45 6A 4B 00 57 12 C8 00  Ej[.m.È.EjK.W.È.
000000B0  45 6A 4C 00 CE 12 C8 00 45 6A 5C 00 4D 12 C8 00  EjL.Î.È.Ej\.M.È.
000000C0  45 6A 59 00 4D 12 C8 00 52 69 63 68 4C 12 C8 00  EjY.M.È.RichL.È.
000000D0  00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00  ........PE..L...
000000E0  9D 97 E7 4C 00 00 00 00 00 00 00 00 E0 00 02 01  .—çL........à...
000000F0  0B 01 09 00 00 2E 05 00 00 A6 06 00 00 00 00 00  .........¦......
00000100  6C 2D 01 00 00 10 00 00 00 20 05 00 00 00 00 01  l-....... ......
00000110  00 10 00 00 00 02 00 00 06 00 01 00 06 00 01 00  ................
00000120  06 00 01 00 00 00 00 00 00 00 0C 00 00 04 00 00  ................
00000130  30 BD 0C 00 02 00 40 81 00 00 04 00 00 20 00 00  0½....@...... ..
00000140  00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00  ................
00000150  00 00 00 00 00 00 00 00 FC 1A 05 00 54 01 00 00  ........ü...T...
00000160  00 90 05 00 98 27 06 00 00 00 00 00 00 00 00 00  ....˜'..........
00000170  00 00 00 00 00 00 00 00 00 C0 0B 00 3C 3B 00 00  .........À..<;..
00000180  44 3C 05 00 38 00 00 00 00 00 00 00 00 00 00 00  D<..8...........
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  30 04 03 00 40 00 00 00 70 02 00 00 54 01 00 00  0...@...p...T...
000001B0  00 10 00 00 30 06 00 00 78 1A 05 00 40 00 00 00  ....0...x...@...
000001C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

FILE TYPES: MAGIC BYTES

Executable Binaries	**Mnemonic**	**Signature**
DOS Executable	"MZ"	0x4D 0x5A
PE32 Executable	"MZ"...."PE.."	0x4D 0x5A ... 0x50 0x45 0x00 0x00
Mach-O Executable (32 bit)	"FEEDFACE"	0xFE 0xED 0xFA 0xCE
Mach-O Executable (64 bit)	"FEEDFACF"	0xFE 0xED 0xFA 0xCF
ELF Executable	".ELF"	0x7F 0x45 0x4C 0x46
Compressed Archives	**Mnemonic**	**Signature**
Zip Archive	"PK.."	0x50 0x4B 0x03 0x04
Rar Archive	"Rar!...."	0x52 0x61 0x72 0x21 0x1A 0x07 0x01 0x00
Ogg Container	"OggS"	0x4F 0x67 0x67 0x53
Matroska/EBML Container	N/A	0x45 0x1A 0xA3 0xDF
Image File Formats	**Mnemonic**	**Signature**
PNG Image	".PNG...."	0x89 0x50 0x4E 0x47 0x0D 0x0A 0x1A 0x0A
BMP Image	"BM"	0x42 0x4D
GIF Image	"GIF87a"	0x47 0x49 0x46 0x38 0x37 0x61
	"GIF89a"	0x47 0x49 0x46 0x38 0x39 0x61

FILE TYPES: MAGIC BYTES


HxD - [C:\Windows\System32\calc.exe]
File Edit Search View Analysis Extras Window ?
16
ANSI
hex
calc.exe

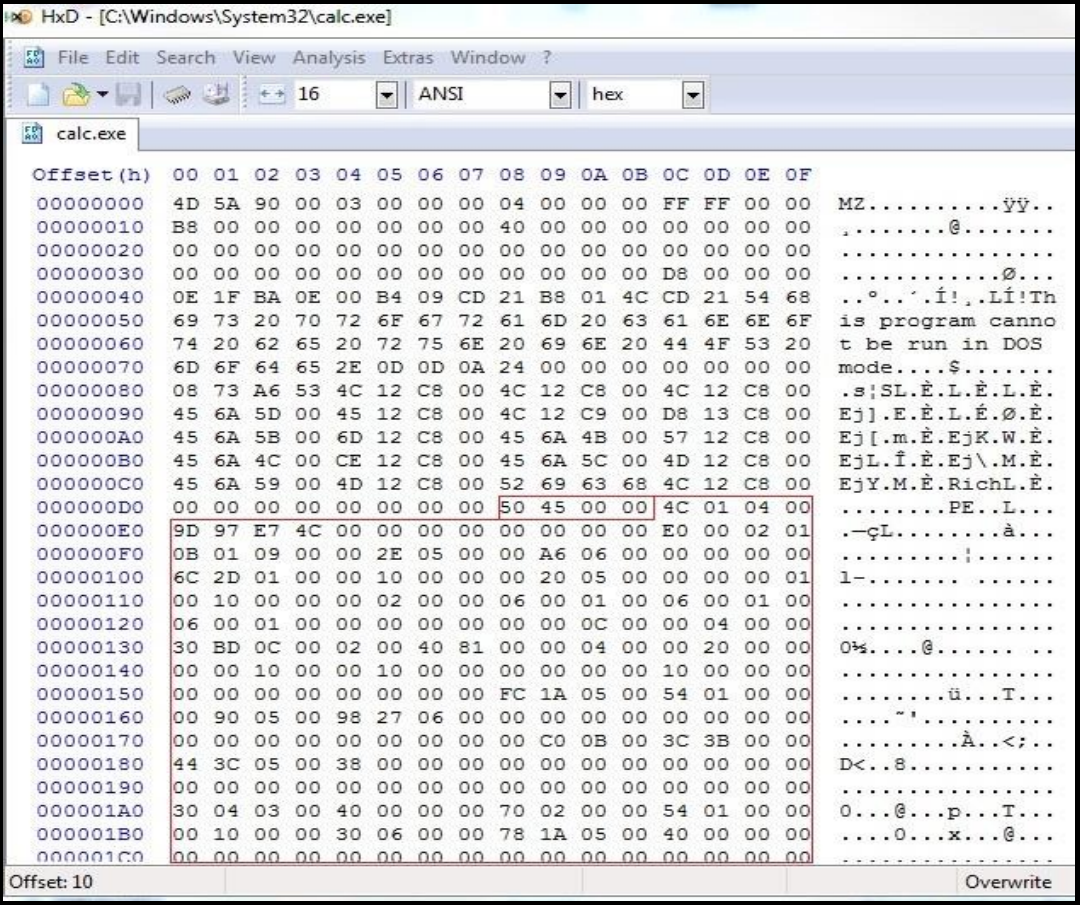
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 D8 00 00 00  ............Ø...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  08 73 A6 53 4C 12 C8 00 4C 12 C8 00 4C 12 C8 00  .s¦SL.É.L.É.L.É.
00000090  45 6A 5D 00 45 12 C8 00 4C 12 C9 00 D8 13 C8 00  Ej].E.È.L.É.Ø.È.
000000A0  45 6A 5B 00 6D 12 C8 00 45 6A 4B 00 57 12 C8 00  Ej[.m.È.EjK.W.È.
000000B0  45 6A 4C 00 CE 12 C8 00 45 6A 5C 00 4D 12 C8 00  EjL.Î.È.Ej\.M.È.
000000C0  45 6A 59 00 4D 12 C8 00 52 69 63 68 4C 12 C8 00  EjY.M.È.RichL.È.
000000D0  00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00  ........PE..L...
000000E0  9D 97 E7 4C 00 00 00 00 00 00 00 00 E0 00 02 01  .—çL........à...
000000F0  0B 01 09 00 00 2E 05 00 00 A6 06 00 00 00 00 00  ..........¦.....
00000100  6C 2D 01 00 00 10 00 00 00 20 05 00 00 00 00 01  l-....... ......
00000110  00 10 00 00 00 02 00 00 06 00 01 00 06 00 01 00  ................
00000120  06 00 01 00 00 00 00 00 00 00 0C 00 00 04 00 00  ................
00000130  30 BD 0C 00 02 00 40 81 00 00 04 00 00 20 00 00  0½....@...... ..
00000140  00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00  ................
00000150  00 00 00 00 00 00 00 00 FC 1A 05 00 54 01 00 00  ........ü...T...
00000160  00 90 05 00 98 27 06 00 00 00 00 00 00 00 00 00  ....˜'..........
00000170  00 00 00 00 00 00 00 00 00 C0 0B 00 3C 3B 00 00  .........À..<;..
00000180  44 3C 05 00 38 00 00 00 00 00 00 00 00 00 00 00  D<..8...........
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  30 04 03 00 40 00 00 00 70 02 00 00 54 01 00 00  0...@...p...T...
000001B0  00 10 00 00 30 06 00 00 78 1A 05 00 40 00 00 00  ....0...x...@...
000001C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Offset: 10

Overwrite

FILE TYPES: MAGIC BYTES



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 D8 00 00 00  ............Ø...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  08 73 A6 53 4C 12 C8 00 4C 12 C8 00 4C 12 C8 00  .s¦SL.È.L.È.L.È.
00000090  45 6A 5D 00 45 12 C8 00 4C 12 C9 00 D8 13 C8 00  Ej].E.È.L.É.Ø.È.
000000A0  45 6A 5B 00 6D 12 C8 00 45 6A 4B 00 57 12 C8 00  Ej[.m.È.EjK.W.È.
000000B0  45 6A 4C 00 CE 12 C8 00 45 6A 5C 00 4D 12 C8 00  EjL.Î.È.Ej\.M.È.
000000C0  45 6A 59 00 4D 12 C8 00 52 69 63 68 4C 12 C8 00  EjY.M.È.RichL.È.
000000D0  00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00  ........PE..L...
000000E0  9D 97 E7 4C 00 00 00 00 00 00 00 00 E0 00 02 01  .—çL........à...
000000F0  0B 01 09 00 00 2E 05 00 00 A6 06 00 00 00 00 00  .........¦......
00000100  6C 2D 01 00 00 10 00 00 00 20 05 00 00 00 00 01  l-....... ......
00000110  00 10 00 00 00 02 00 00 06 00 01 00 06 00 01 00  ................
00000120  06 00 01 00 00 00 00 00 00 00 0C 00 00 04 00 00  ................
00000130  30 BD 0C 00 02 00 40 81 00 00 04 00 00 20 00 00  0½....@...... ..
00000140  00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00  ................
00000150  00 00 00 00 00 00 00 00 FC 1A 05 00 54 01 00 00  ........ü...T...
00000160  00 90 05 00 98 27 06 00 00 00 00 00 00 00 00 00  ....˜'..........
00000170  00 00 00 00 00 00 00 00 00 C0 0B 00 3C 3B 00 00  .........À..<;..
00000180  44 3C 05 00 38 00 00 00 00 00 00 00 00 00 00 00  D<..8...........
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  30 04 03 00 40 00 00 00 70 02 00 00 54 01 00 00  0...@...p...T...
000001B0  00 10 00 00 30 06 00 00 78 1A 05 00 40 00 00 00  ....0...x...@...
000001C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

ACCESS METHODS

- □ **Sequential Access**

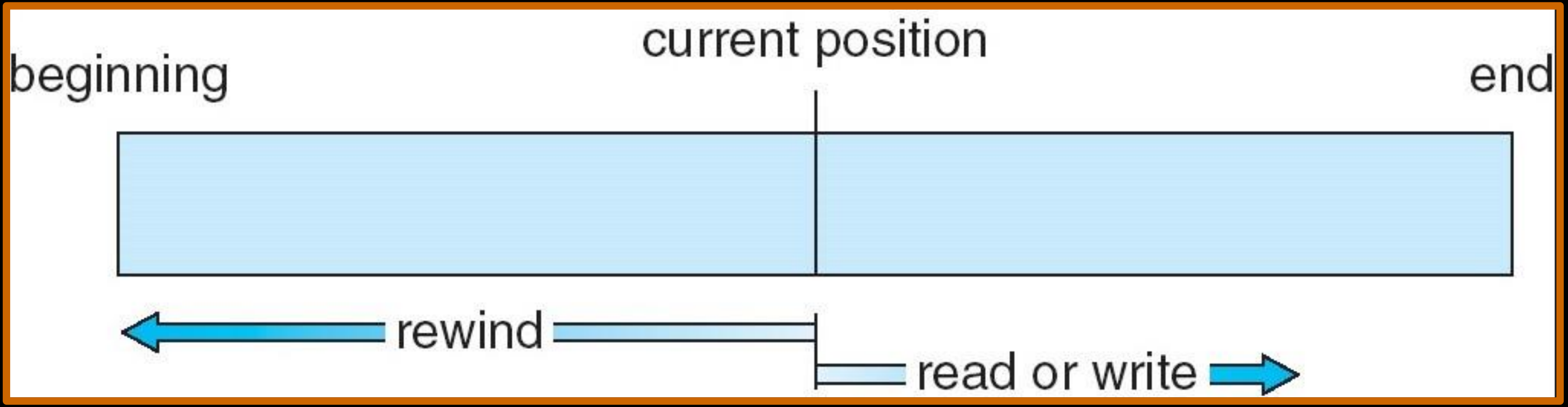
read next
write next
reset
no read after last write
(rewrite)

- □ **Direct Access**

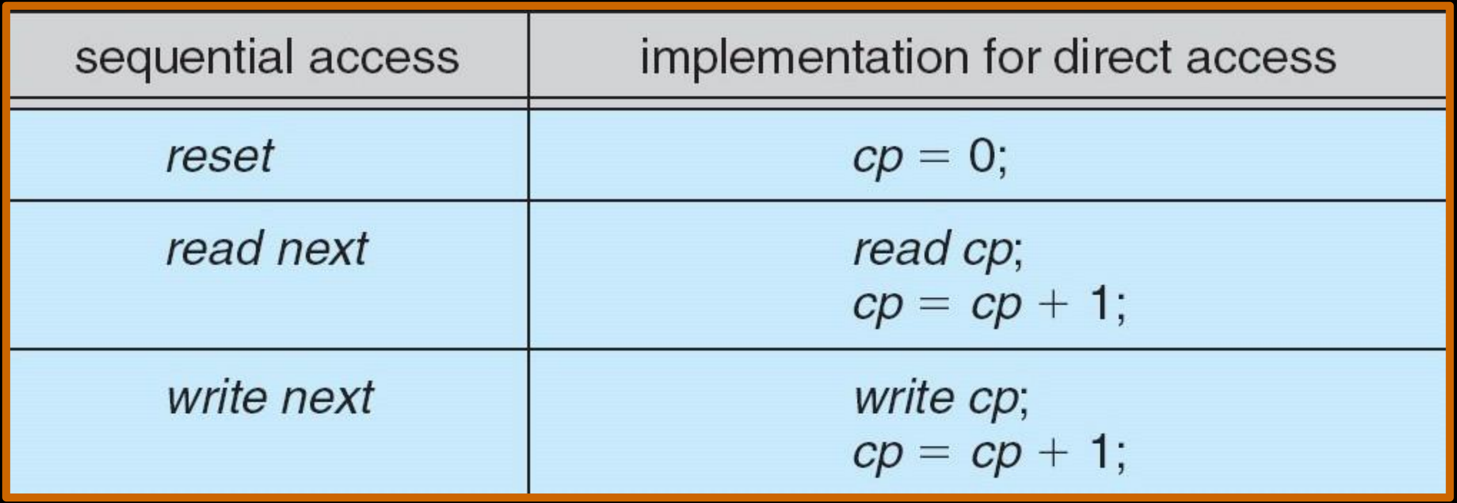
read *n*
write *n*
position to *n*
read next
write next
rewrite *n*

n = relative block number

SEQUENTIAL-ACCESS FILE



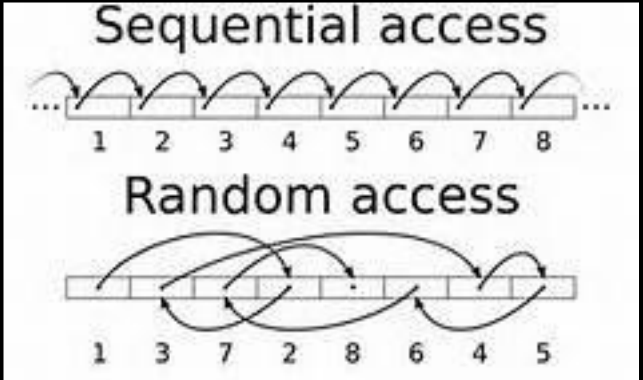
SIMULATION OF SEQUENTIAL ACCESS ON A DIRECT-ACCESS FILE

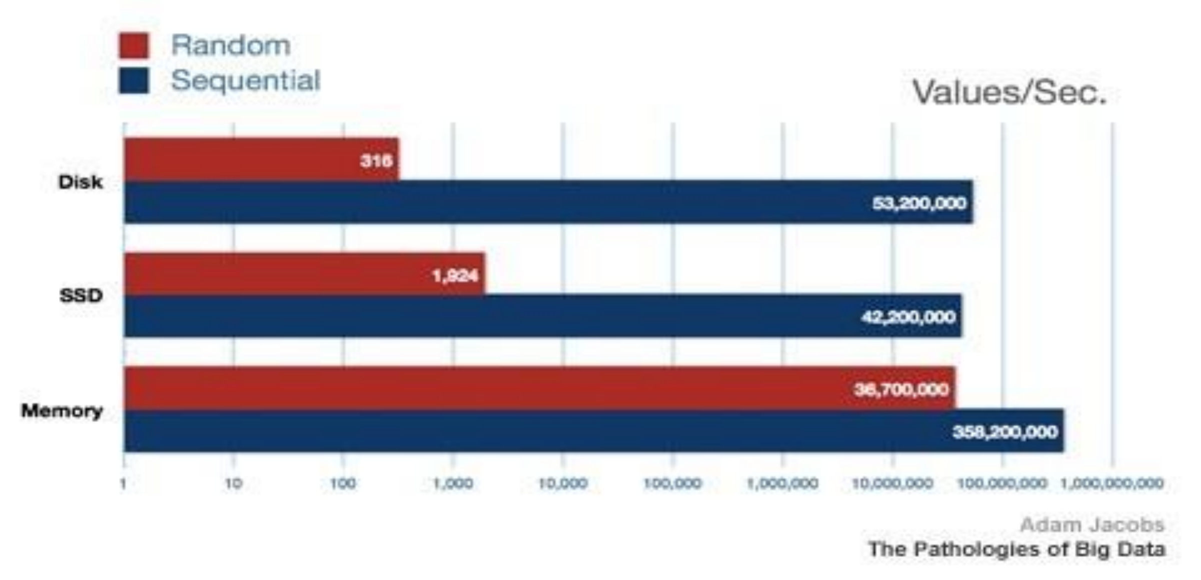


sequential access	implementation for direct access
reset	$cp = 0;$
read next	*read* cp; $cp = cp + 1;$
write next	*write* cp; $cp = cp + 1;$

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

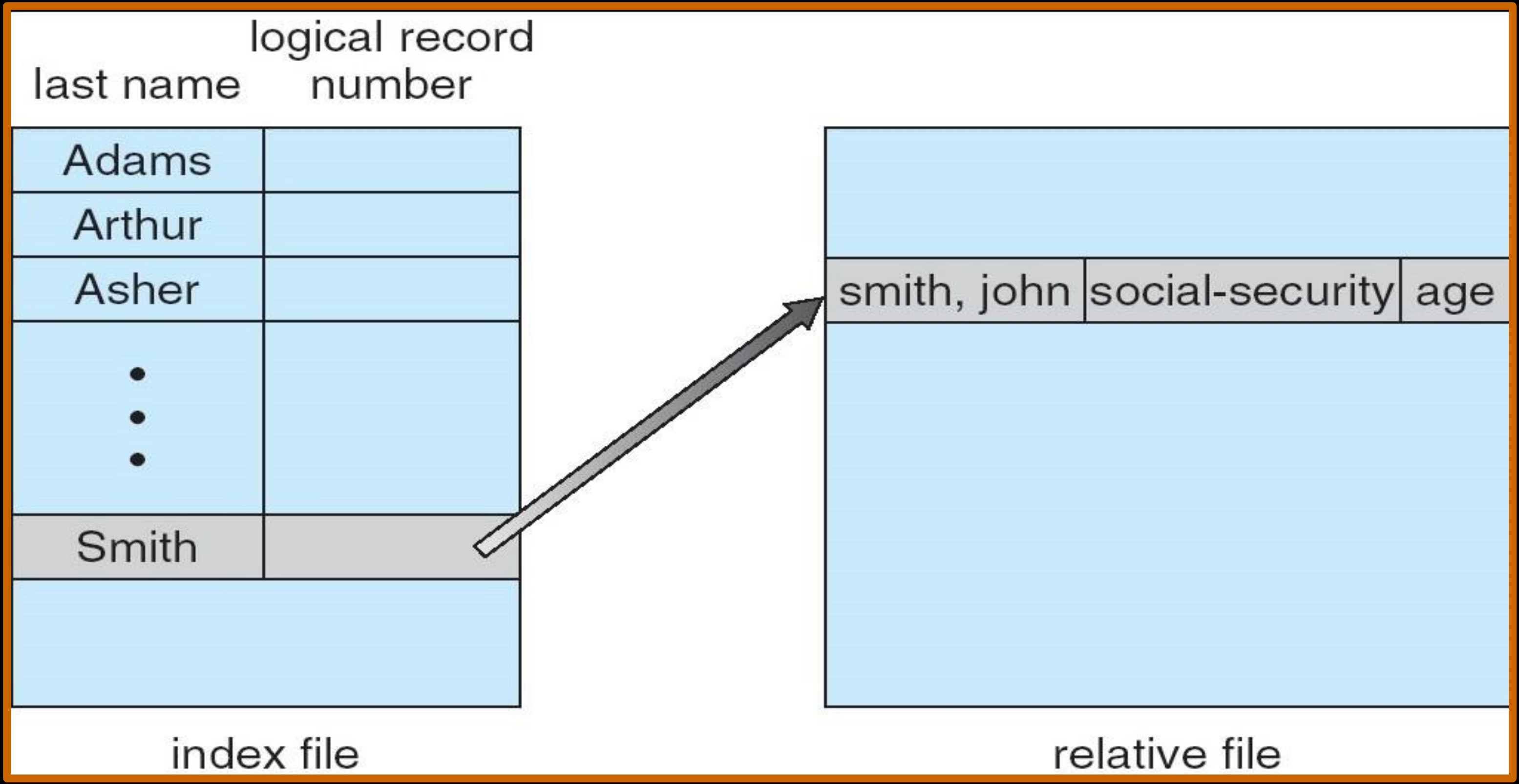
SEQUENTIAL ACCESS VS DIRECT ACCESS





Adam Jacobs
The Pathologies of Big Data

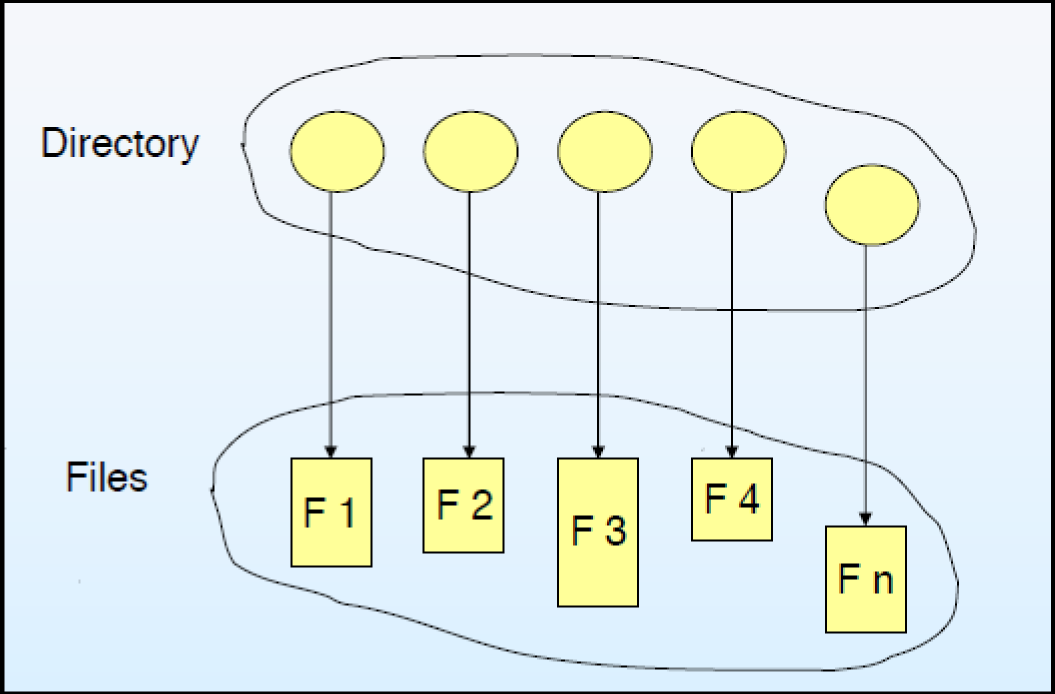
EXAMPLE OF INDEX AND RELATIVE FILES



Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

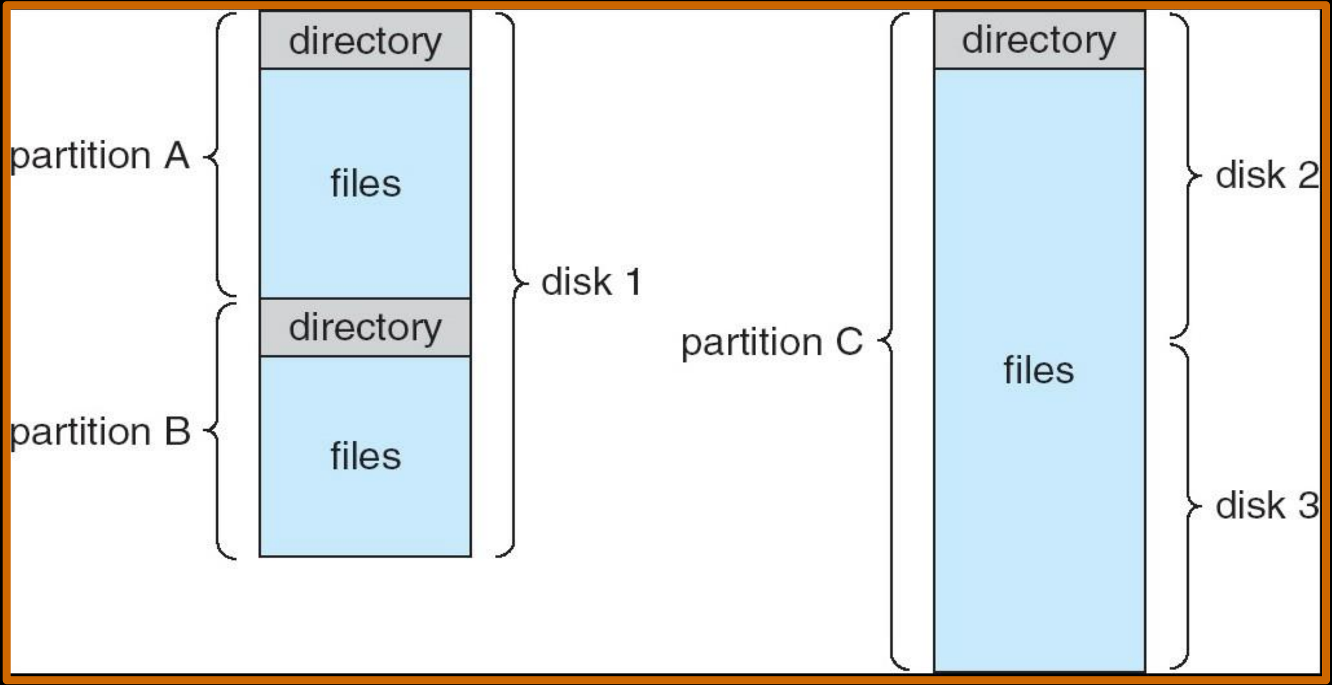
DIRECTORY STRUCTURE

- A collection of nodes containing (management) information about all files



Both the directory structure and the files reside on disk Backups of these two structures are kept on tapes

A TYPICAL FILE-SYSTEM ORGANIZATION



The directory records information about the files in the system – such as name, location, size and type.

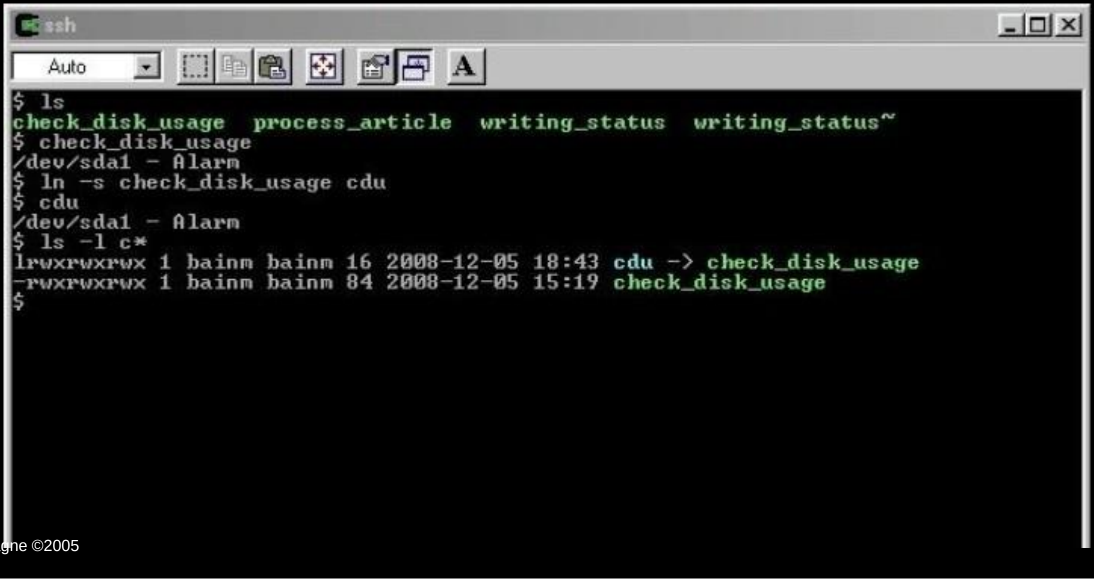
Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

OPERATIONS PERFORMED ON DIRECTORY

- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file
- Traverse the file system – access every dir and file for backing up.

ORGANIZE THE DIRECTORY (LOGICALLY) TO OBTAIN

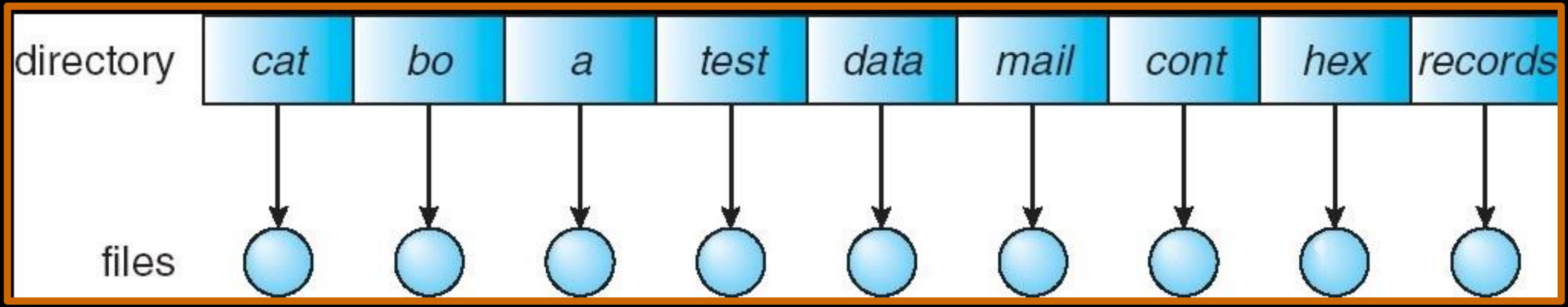
- Efficiency – locating a file quickly
- Naming – convenient to users
  - Two users can have same name for different files
  - The same file can have several different names
- Grouping – logical grouping of files by properties, (e.g., all Java programs, all games, …)



```
$ ls
check_disk_usage   process_article   writing_status   writing_status~
$ check_disk_usage
/dev/sda1 - Alarm
$ ln -s check_disk_usage cdu
$ cdu
/dev/sda1 - Alarm
$ ls -l c*
lrwxrwxrwx 1 bainm bainm 16 2008-12-05 18:43 cdu -> check_disk_usage
-rwxrwxrwx 1 bainm bainm 84 2008-12-05 15:19 check_disk_usage
$
```

SINGLE-LEVEL DIRECTORY

- A single directory for all users



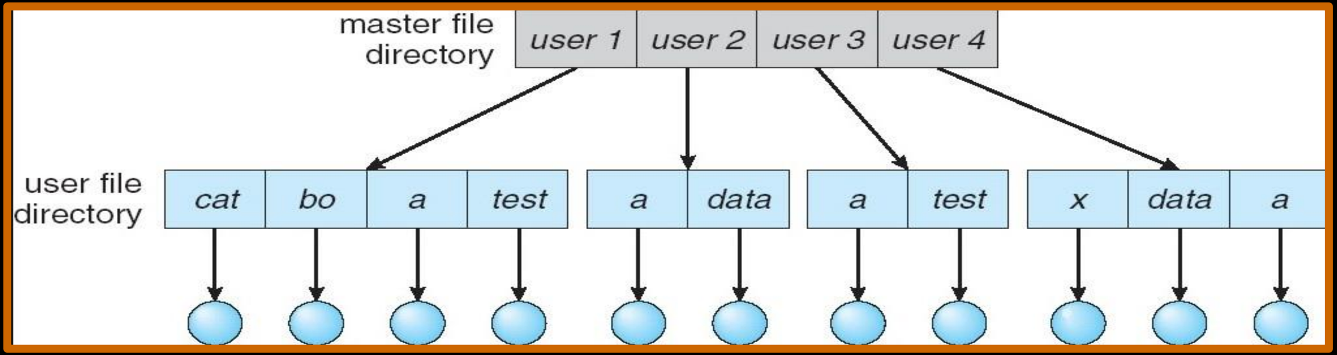
Naming problem

Grouping problem

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

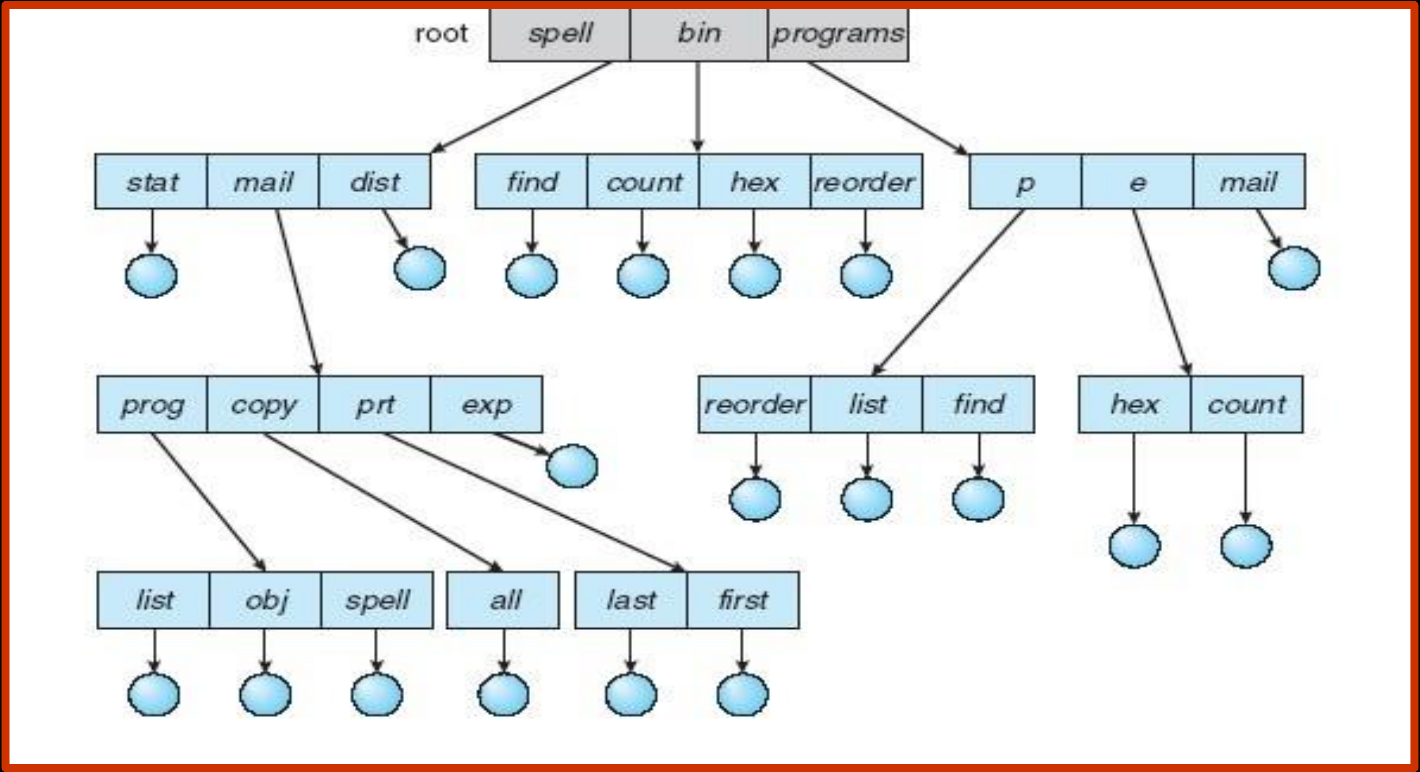
TWO-LEVEL DIRECTORY

- □ Separate directory for each user



- □ Path name
- □ Can have the same file name for different user
- □ Efficient searching
- □ No grouping capability

TREE-STRUCTURED DIRECTORIES



TREE-STRUCTURED DIRECTORIES (CONT)

- Each directory entry contains a bit defining the entry as file(0) or directory(1).
- Efficient searching
- Grouping Capability
- Current directory (working directory)
  - cd /spell/mail/prog
  - type list

TREE-STRUCTURED DIRECTORIES (CONT)

- **Absolute** or **relative** path name
- Creating a new file is done in current directory
- Delete a file

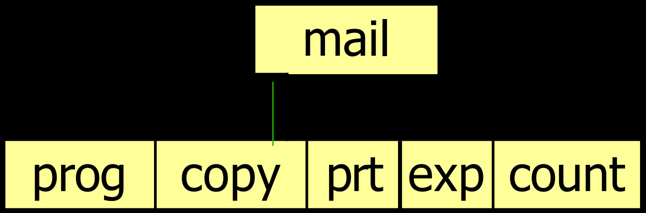
```
rm <file-name>
```

- Creating a new subdirectory is done in current directory

```
mkdir <dir-name>
```

Example: if in current directory /mail

```
mkdir count
```



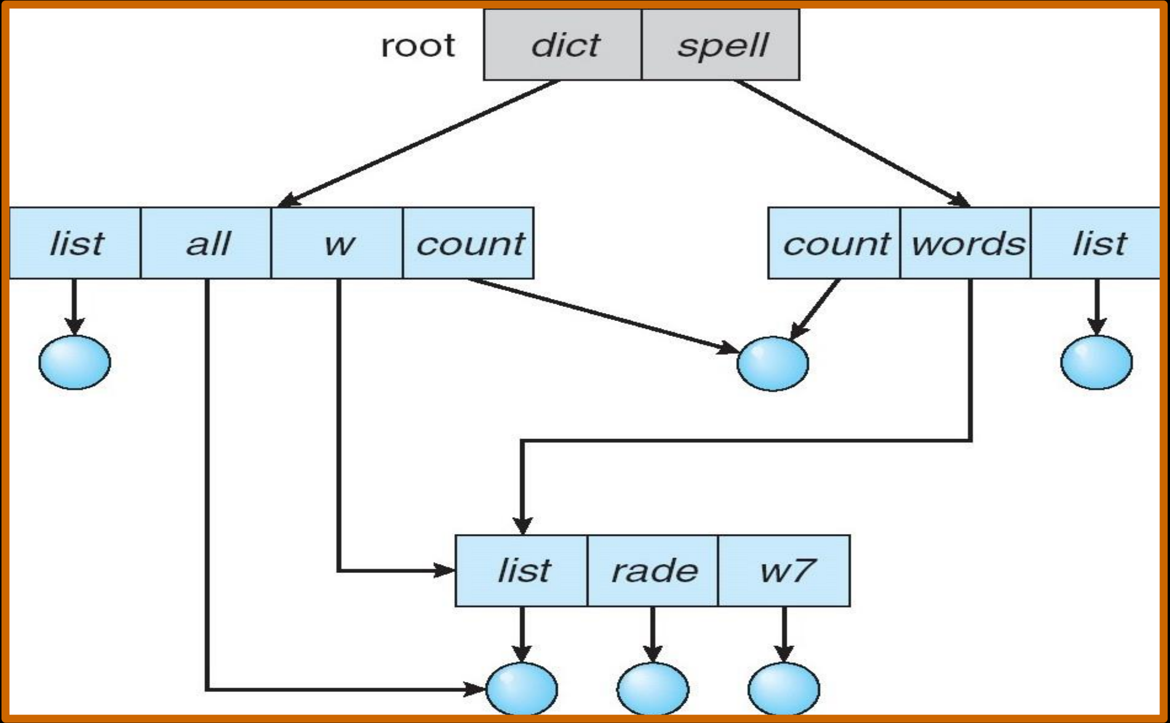
mail

prog copy prt exp count

Deleting "mail" ⇒ deleting the entire subtree rooted by "mail"

ACYCLIC-GRAPH DIRECTORIES

- Requirement for file sharing
- Have shared subdirectories and files



ACYCLIC-GRAPH DIRECTORIES (CONT.)

- Two different names (aliasing)

- If *dict* deletes *count* ⇒ dangling pointer

  Solutions:

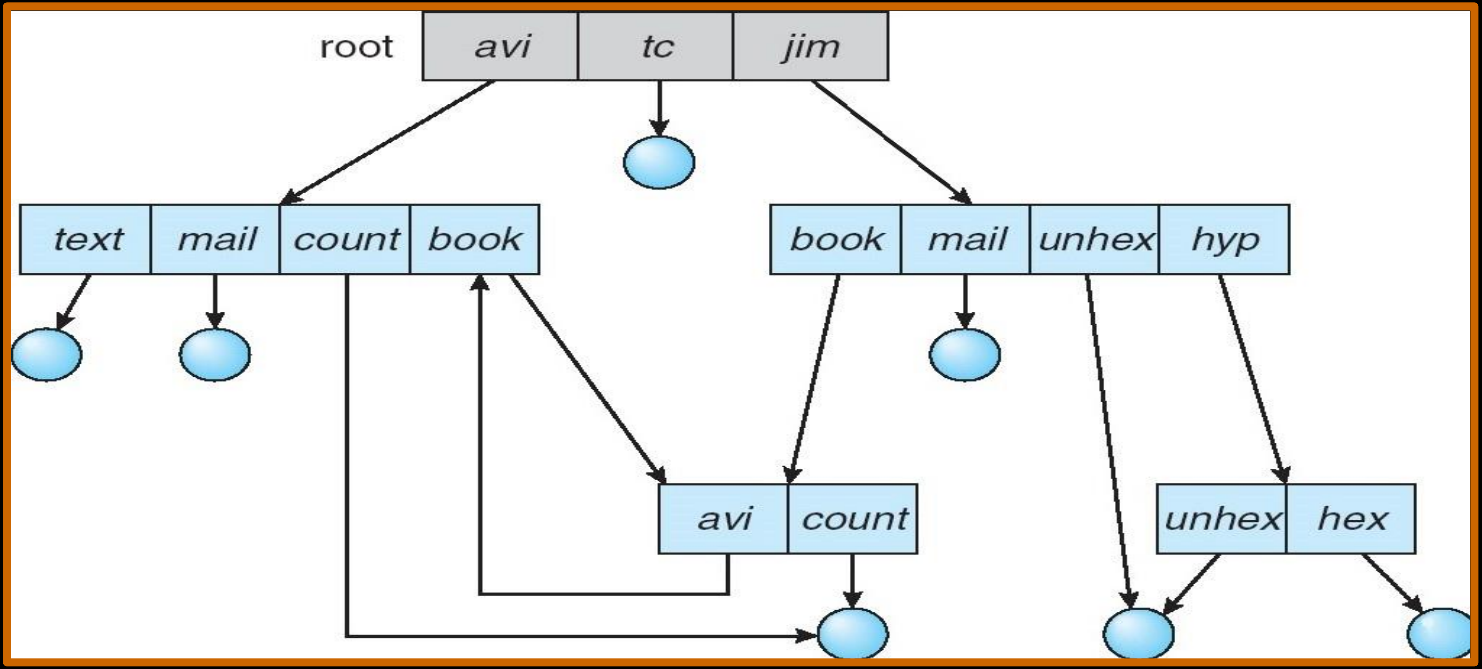
  - Backpointers (keep a list of references to a file), so we can delete all pointers

    But: Large, variable size reference list is a problem

  - Entry-hold-count solution
- New directory entry type
  - **Link** – another name (pointer) to an existing file
  - **Resolve the link** – follow pointer to locate the file

GENERAL GRAPH DIRECTORY

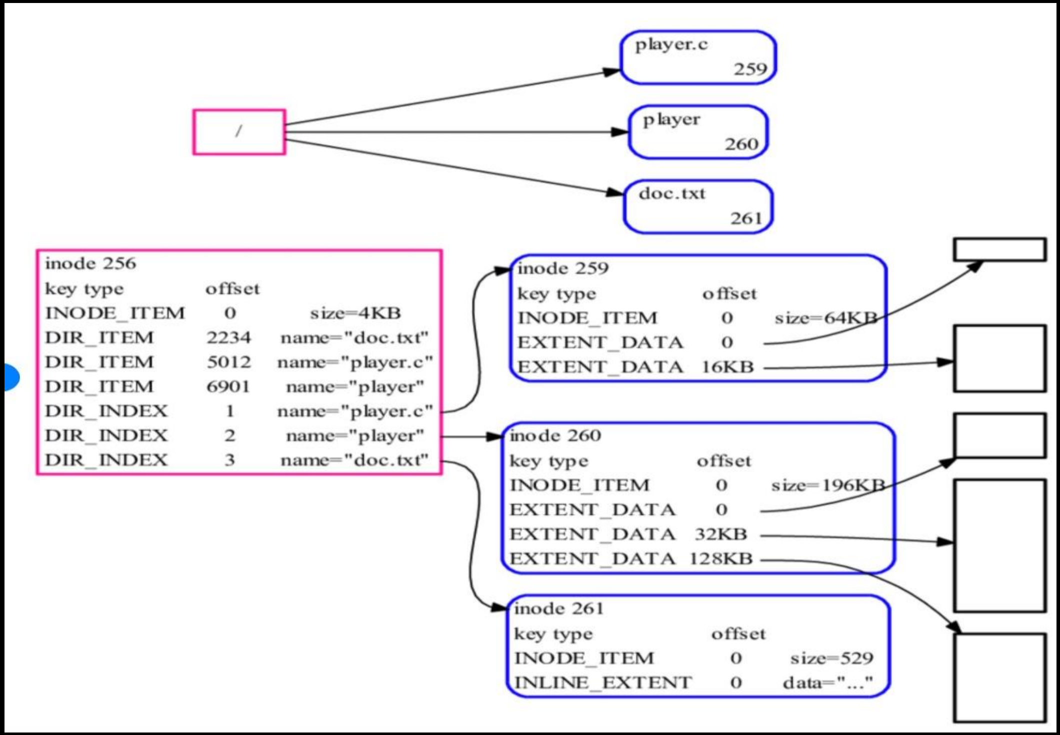
A serious problem with acyclic-graph is to ensure no cycles.



GENERAL GRAPH DIRECTORY (CONT.)

- If cycles allowed
  - Repeated search the same object
  - File deletion problem (count <>0 even if unused)
- How do we guarantee no cycles?
  - Allow only links to file not subdirectories
  - Garbage collection
  - Every time a new link is added use a cycle detection algorithm to determine whether it is OK

STRUCTURE OF DIRECTORY FILE



/

player.c 259

player 260

doc.txt 261

inode 256		
key type	offset	
INODE_ITEM	0	size=4KB
DIR_ITEM	2234	name="doc.txt"
DIR_ITEM	5012	name="player.c"
DIR_ITEM	6901	name="player"
DIR_INDEX	1	name="player.c"
DIR_INDEX	2	name="player"
DIR_INDEX	3	name="doc.txt"

inode 259		
key type	offset	
INODE_ITEM	0	size=64KB
EXTENT_DATA	0	
EXTENT_DATA	16KB	

inode 260		
key type	offset	
INODE_ITEM	0	size=196KB
EXTENT_DATA	0	
EXTENT_DATA	32KB	
EXTENT_DATA	128KB	

inode 261		
key type	offset	
INODE_ITEM	0	size=529
INLINE_EXTENT	0	data="..."

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

LINUX DISK PARTITIONING

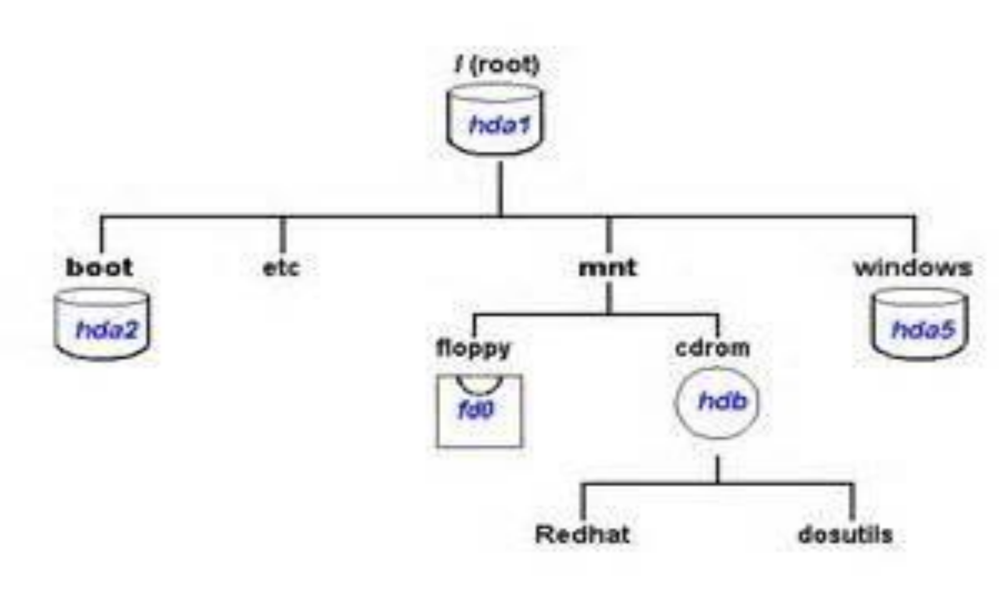
	Linux
IDE	/dev/hd[a-d]
SCSI/SATA/USB	/dev/sd[a-p]
USB	/dev/sd[a-p](SATA)
	/dev/fd[0-1]
	25: /dev/lp[0-2] USB: /dev/usb/lp[0-15]
	PS2: /dev/psaux USB: /dev/usb/mouse[0-15]
CDROM/DVDROM	/dev/cdrom
	/dev/mouse

LINUX DISK PARTITIONING

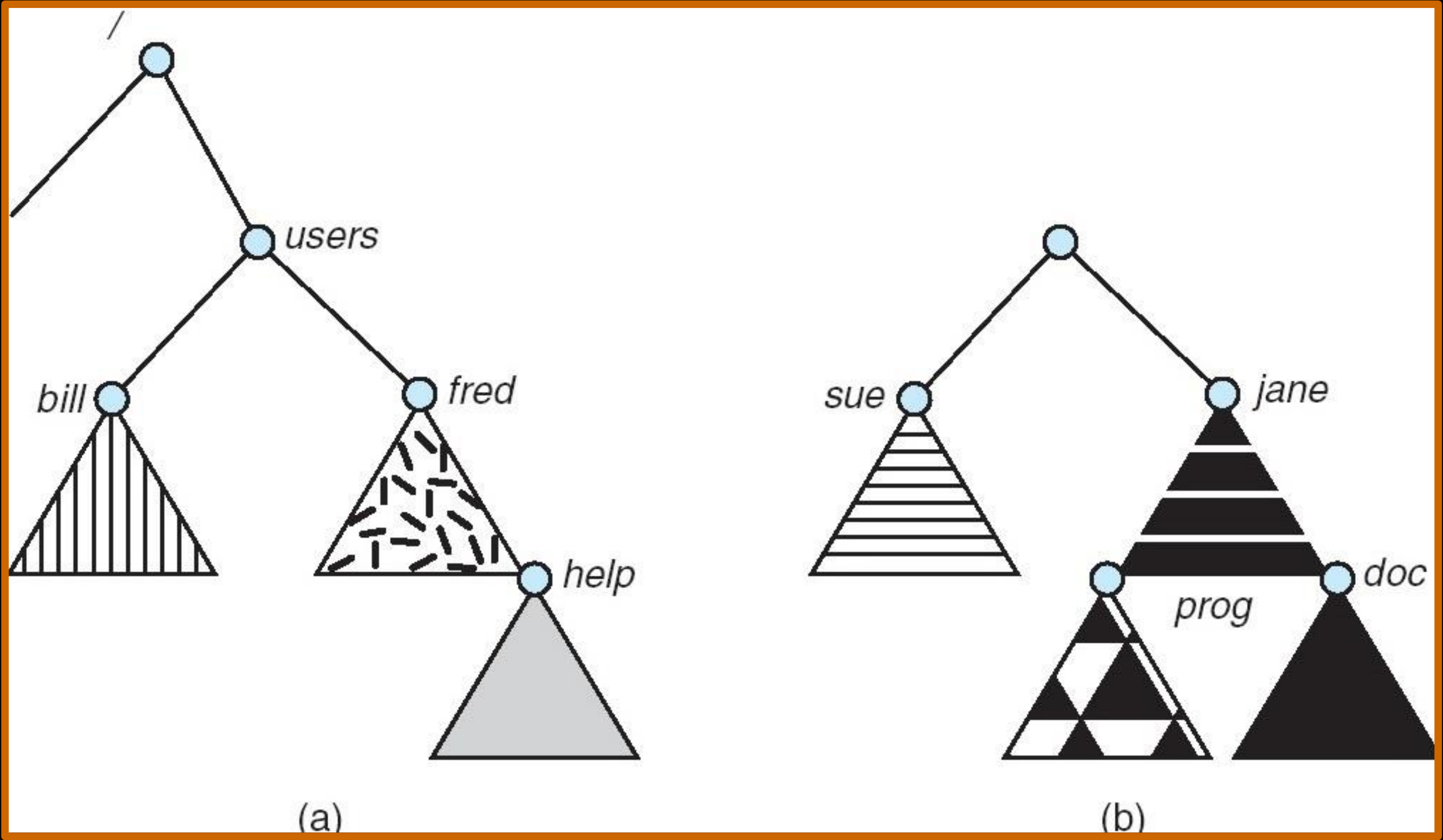
- MBR
  - □ P1:/dev/hda1
  - □ P2:/dev/hda2
  - □ L1:/dev/hda5
  - □ L2:/dev/hda6
  - □ L3:/dev/hda7
  - □ L4:/dev/hda8
  - □ L5:/dev/hda9

FILE SYSTEM MOUNTING

- A file system must be **mounted** before it can be accessed
- An un-mounted file system (i.e. Fig. 10-12(b)) is mounted at a **mount point**



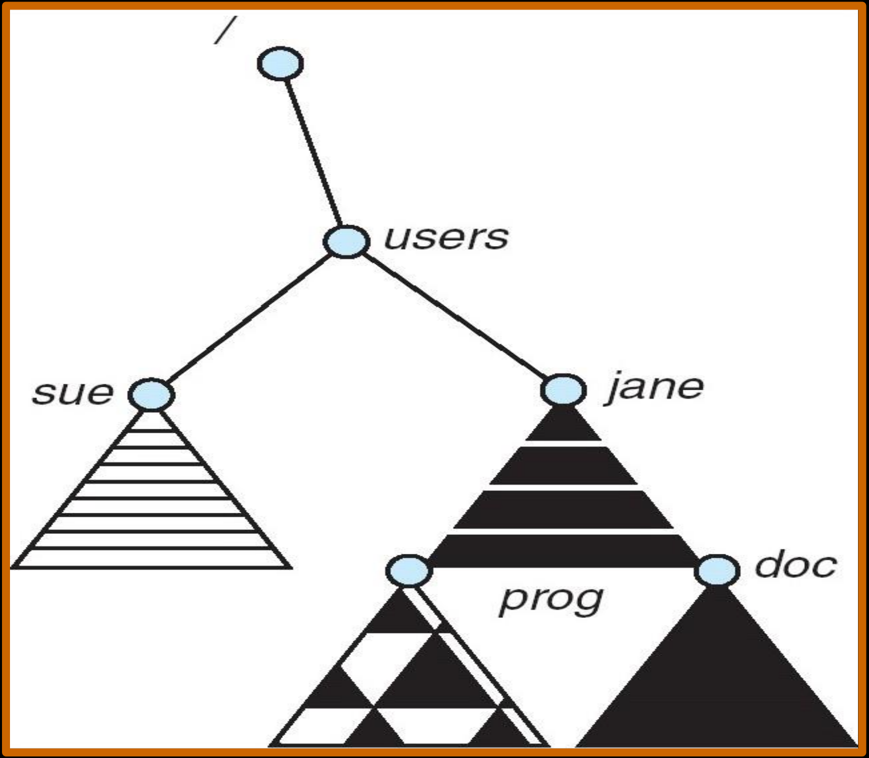
(A) EXISTING. (B) UNMOUNTED PARTITION



Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

MOUNT POINT

```
$ mount        /dev/dsk       /users
```



Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

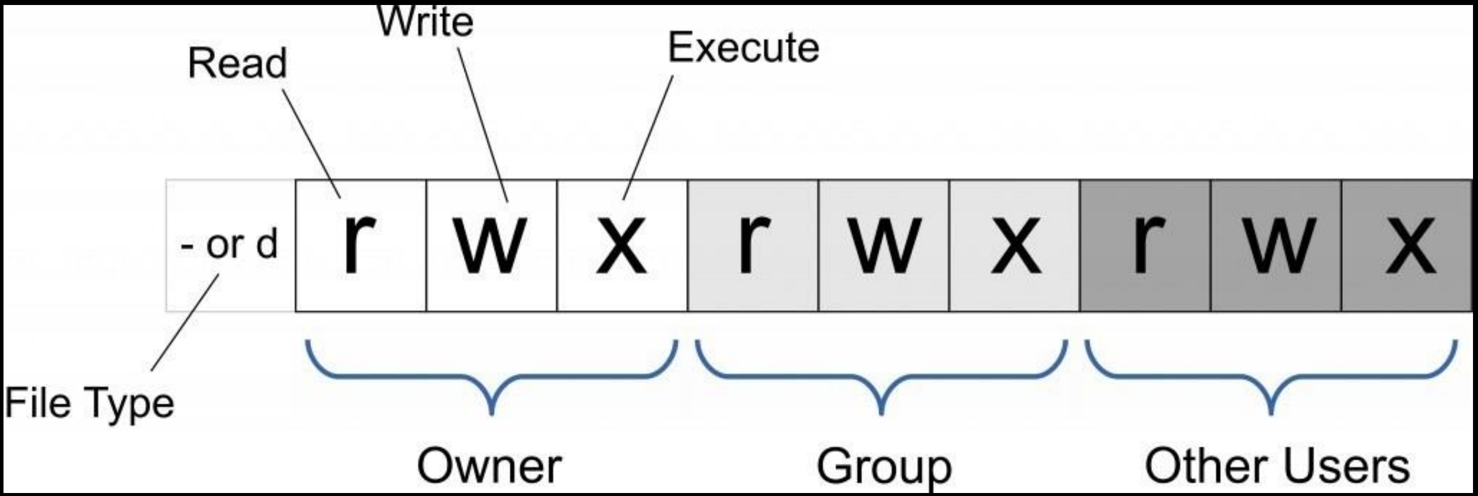
FILE SHARING

- Sharing of files on multi-user systems is desirable
- Sharing may be done through a **protection** scheme
- On distributed systems, files may be shared across a network
- Network File System (NFS) is a common distributed file-sharing method

FILE SHARING – MULTIPLE USERS

- **User IDs** identify users, allowing permissions and protections to be per-user

- **Group IDs** allow users to be in groups, permitting group access rights

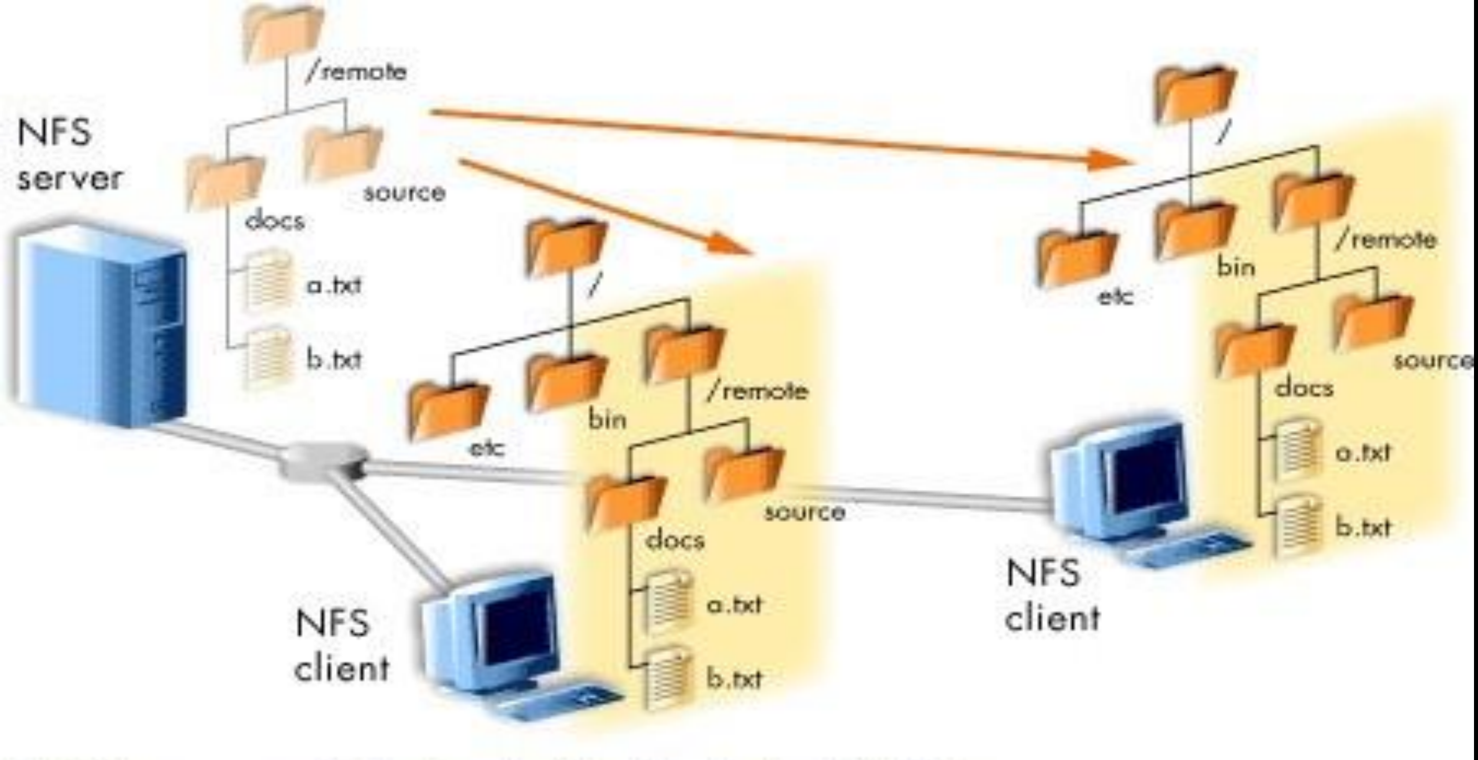


Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

FILE SHARING – REMOTE FILE SYSTEMS

- Uses networking to allow file system access between systems
  - Manually via programs like FTP
  - Automatically, seamlessly using **distributed file systems**
  - Semi automatically via the **world wide web**
- **Client-server** model allows clients to mount remote file systems from servers
  - Server can serve multiple clients
  - Client and user-on-client identification is insecure or complicated
  - **NFS** is standard UNIX client-server file sharing protocol
  - **CIFS** is standard Windows protocol
  - Standard operating system file calls are translated into remote calls
- Distributed Information Systems **(distributed naming services)** such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote computing

NFS SYSTEM



The NFS server exports the /remote filesystem to two NFS clients.

FILE SHARING – FAILURE MODES

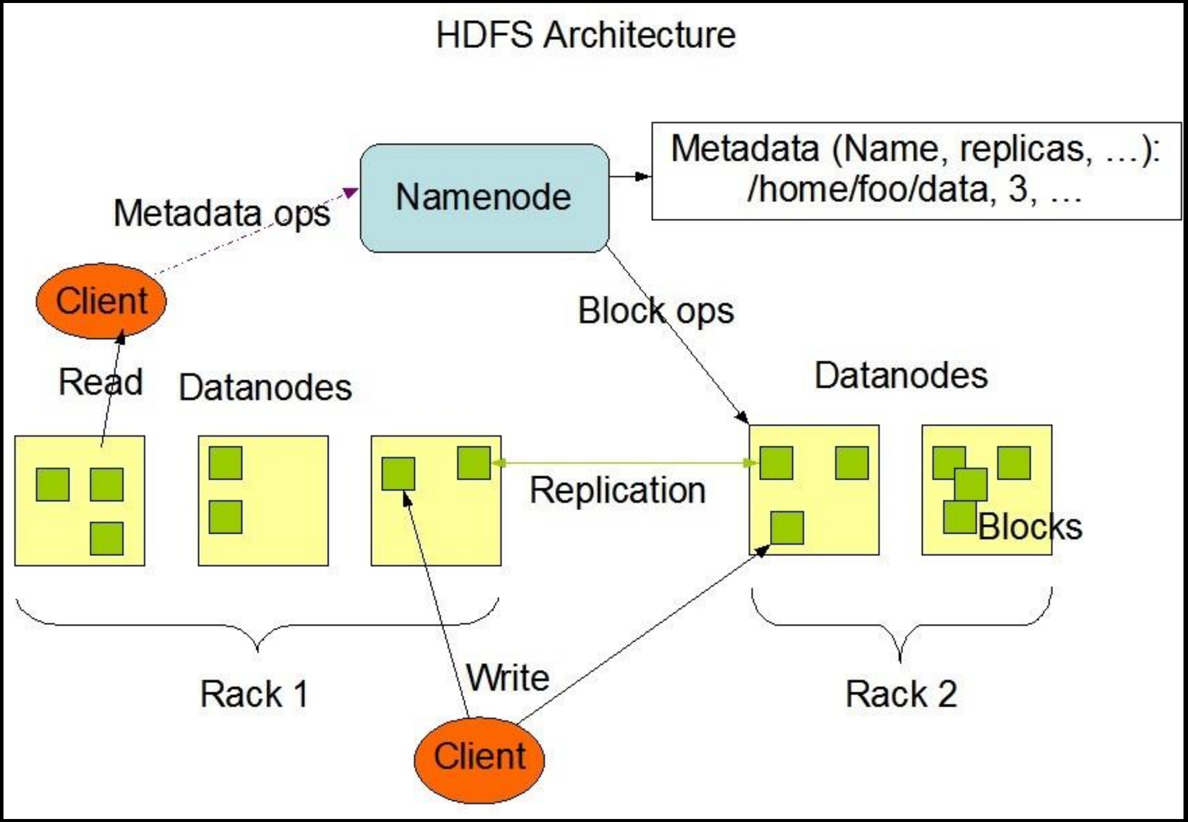
- Remote file systems add new failure modes, due to network failure, server failure
- Recovery from failure can involve state information about status of each remote request
- Stateless protocols such as NFS include all information in each request, allowing easy recovery but less security

FILE SHARING – CONSISTENCY SEMANTICS

- **Consistency semantics** specify how multiple users are to access a shared file simultaneously
  - Similar to Ch 6 process synchronization algorithms
    - Tend to be less complex due to disk I/O and network latency (for remote file systems) – slow speed
  - Andrew File System (AFS) implemented complex remote file sharing semantics
  - Unix file system (UFS) implements:
    - Writes to an open file visible immediately to other users of the same open file
    - Sharing file pointer to allow multiple users to read and write concurrently
  - AFS has session semantics
    - Writes only visible to sessions starting after the file is closed

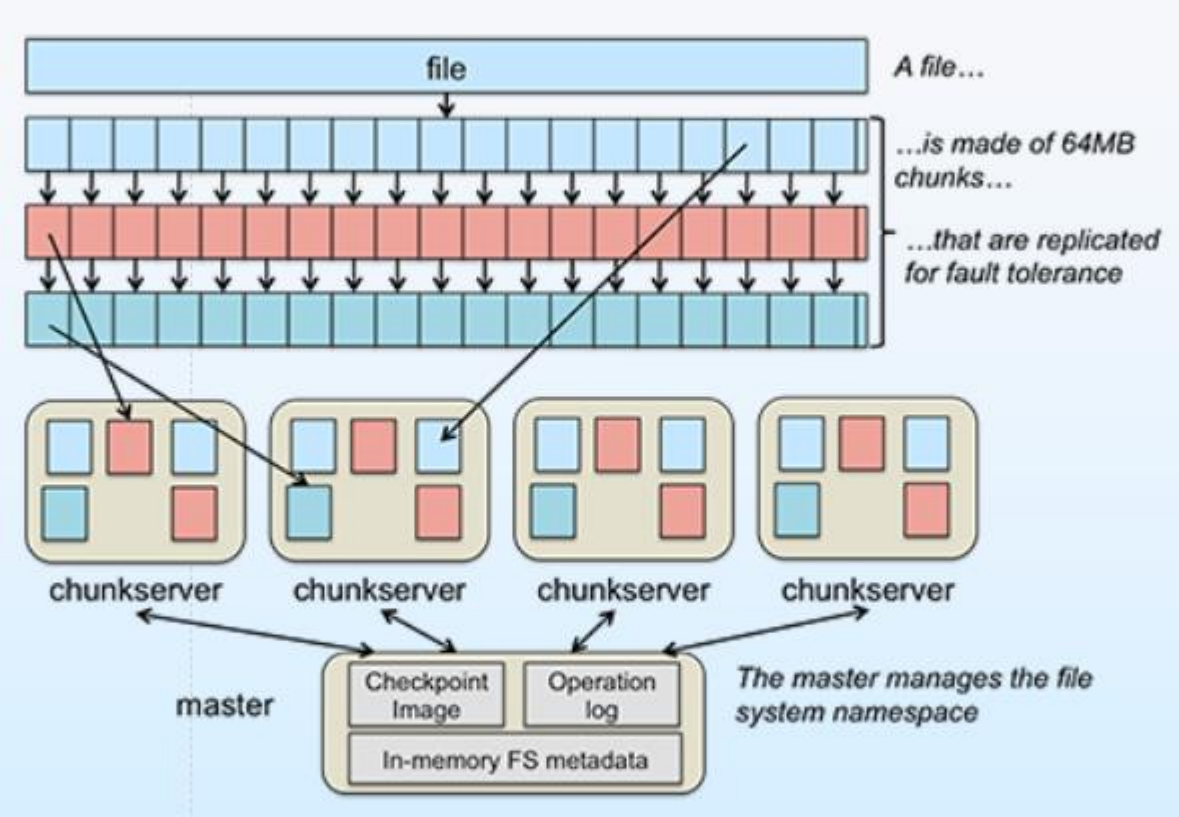
GOOGLE FILE SYSTEM/HDFS

- namenode：datanode » second namenode
- datanode



Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

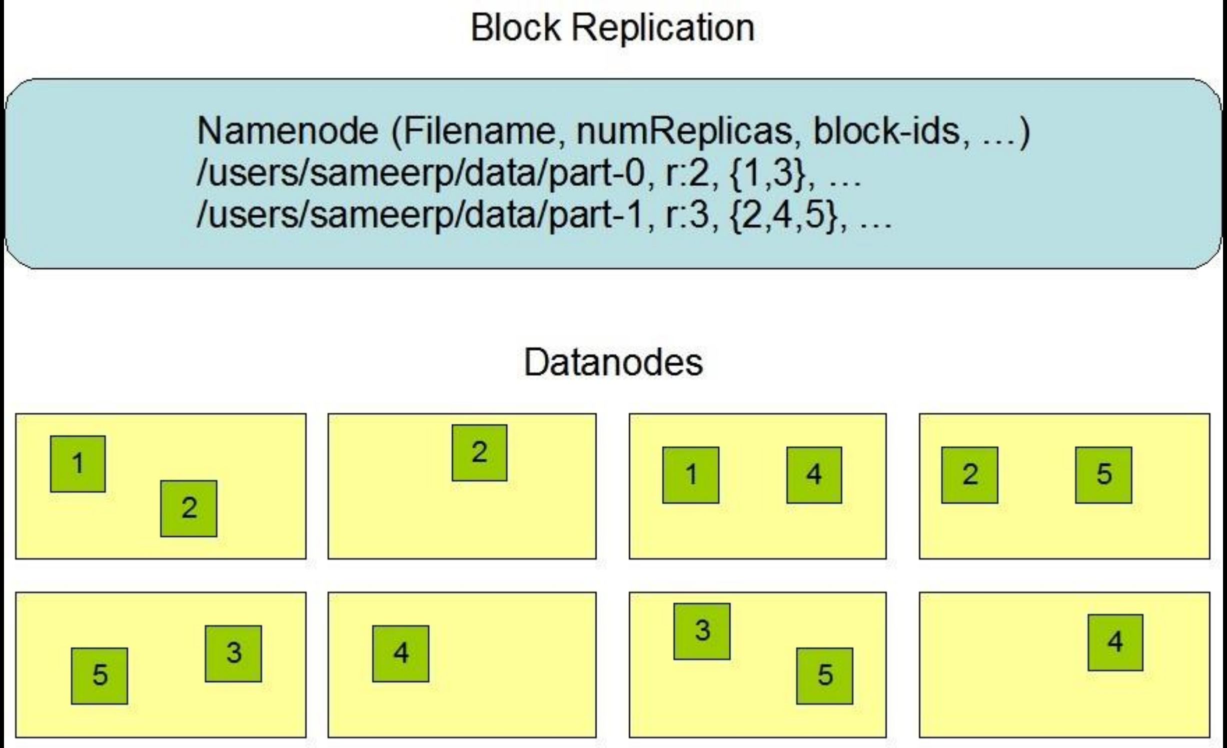
GOOGLE FILE SYSTEM/HDFS


file
A file...
...is made of 64MB chunks...
...that are replicated for fault tolerance
chunkserver
chunkserver
chunkserver
chunkserver
master
Checkpoint Image
Operation log
In-memory FS metadata
The master manages the file system namespace

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

REPLICATION IN HDFS

- replication=3
- 64M



Block Replication

Namenode (Filename, numReplicas, block-ids, …)
/users/sameerp/data/part-0, r:2, {1,3}, …
/users/sameerp/data/part-1, r:3, {2,4,5}, …

Datanodes

PROTECTION

- File owner/creator should be able to control:
  - what can be done
  - by whom

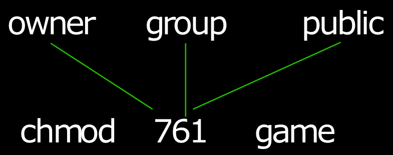
- Types of access
  - **Read**
  - **Write**
  - **Execute**
  - **Append**
  - **Delete**
  - **List**

ACCESS LISTS AND GROUPS

- Mode of access: read, write, execute
- Three classes of users

a) **owner access**	7	⇒	RWX 1 1 1
b) **group access**	6	⇒	RWX 1 1 0
c) **public access**	1	⇒	RWX 0 0 1

- Ask manager to create a group (unique name), say G, and add some users to the group.
- For a particular file (say *game*) or subdirectory, define an appropriate access.

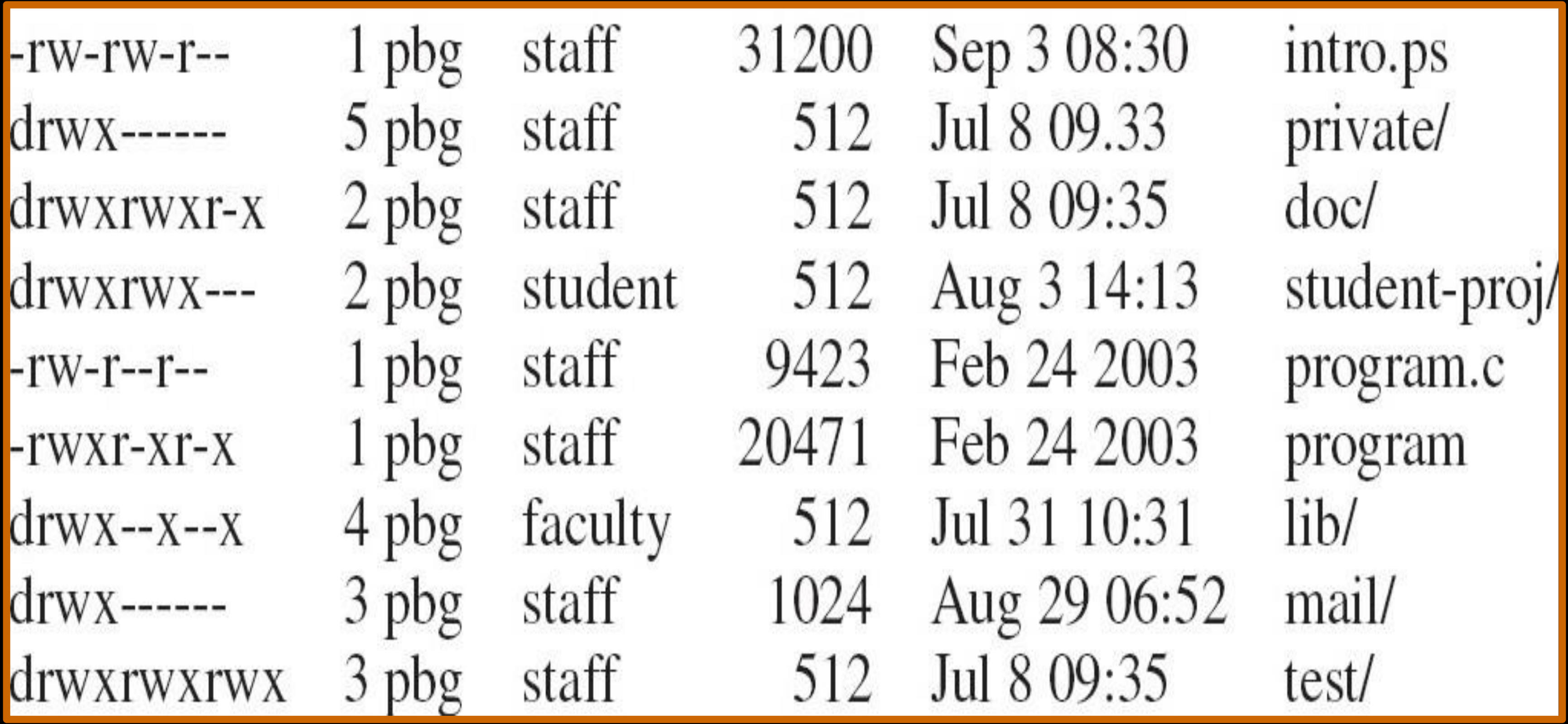


```
chmod   761   game
```

Attach a group to a file

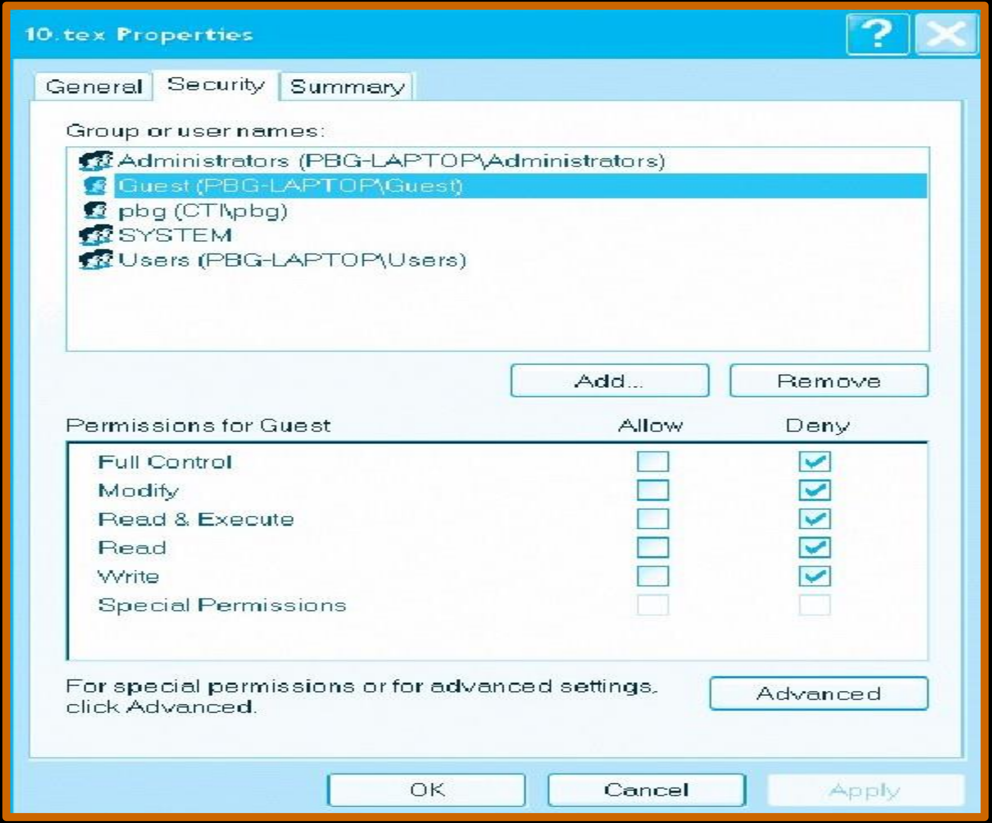
```
chgrp   G   game
```

A SAMPLE UNIX DIRECTORY LISTING



```
-rw-rw-r--    1 pbg  staff     31200  Sep 3 08:30   intro.ps
drwx------    5 pbg  staff       512  Jul 8 09.33   private/
drwxrwxr-x    2 pbg  staff       512  Jul 8 09:35   doc/
drwxrwx---    2 pbg  student     512  Aug 3 14:13   student-proj/
-rw-r--r--    1 pbg  staff      9423  Feb 24 2003   program.c
-rwxr-xr-x    1 pbg  staff     20471  Feb 24 2003   program
drwx--x--x    4 pbg  faculty     512  Jul 31 10:31  lib/
drwx------    3 pbg  staff      1024  Aug 29 06:52  mail/
drwxrwxrwx    3 pbg  staff       512  Jul 8 09:35   test/
```

WINDOWS XP ACCESS-CONTROL LIST MANAGEMENT



10.tex Properties

Group or user names:

- Administrators (PBG-LAPTOP\Administrators)
- Guest (PBG-LAPTOP\Guest)
- pbg (CTI\pbg)
- SYSTEM
- Users (PBG-LAPTOP\Users)

Permissions for Guest	Allow	Deny
Full Control	□	☑
Modify	□	☑
Read & Execute	□	☑
Read	□	☑
Write	□	☑
Special Permissions	□	□

For special permissions or for advanced settings, click Advanced.

Silberschatz, Galvin Operating System Concepts – 7th Edition and Gagne ©2005

LUT
University