OPERATING SYSTEMS

Storage Management

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

File System

Suresh Jamadagni
Department of Computer Science

OPERATING SYSTEMS

Slides Credits for all the PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

OPERATING SYSTEMS

File Concept

- Computers can store information on various storage media, such as magnetic disks, magnetic tapes, and optical disks.
- File is logical storage where as disks are physical storage unit
- What is a File?
  - collection of related information
  - Data cannot to be written to disk unless they are in files
  - Data can be Numeric, alphabetic, alphanumeric, binary
- A file is a sequence of bits, bytes, lines, or records, the meaning of which is defined by the file's creator and user
- Contents of file can be source or executable programs, numeric or text data, photos, music, video, and so on.

File Attributes

Files are named and are referred by their names by the human users.

Attributes of files

- **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.

OPERATING SYSTEMS

File Attributes

- The information about all files is kept in the directory structure, which also resides on secondary storage.
- Directory entry consists of the file's name and its uniqueand its unique identifier.
- Identifier in turn locates the file attributes.

```
File: g.c
  Size: 218           Blocks: 0          IO Block: 4096   regular file
Device: 2h/2d   Inode: 337699720577240  Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2020-09-15 15:18:57.338585500 +0530
Modify: 2021-03-20 10:40:37.416576700 +0530
Change: 2021-03-20 10:40:37.416576700 +0530
```

File Operations

File is an abstract data type.

File Operations

- **Create**
- **Write** – Use the system call to write to a file. A write pointer points to the location in the file, where the next write is to take place.
- **Read** – – Use the system call to write to a file. A read pointer points to the location in the file from where the next read is to take place
- **Reposition within file -** Repositioning within a file need not involve any actual I/O. It is done with system call lseek. Repositioning is also called seek to location in a file
- **Delete:** Use the system call to delete/remove a file.
  - File to be deleted is searched in the directory.
  - Release the memory allocated after deletion

File Operations

- **Truncate:** The user may want to erase the contents of a file but keep its attributes.
  - Instead of deleting the file and then recreating it, use the function that sets the size of file to zero.
- **Appending** to the end of a existing file
- ***Open a file*** – use a system call open()
  - **Open-file table**: tracks 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
- ***Close:*** *When a file is actively not used close it.*

OPERATING SYSTEMS

File Operations

- Several pieces of information are associated with an open file.
  - **File pointer**
  - **File-open count.**
  - **Disk location of the file.**
  - **Access rights**

Open Files

- Provided by some operating systems and file systems
  - Similar to reader-writer locks
  - **Shared lock** similar to reader lock – several processes can acquire concurrently
  - **Exclusive lock** similar to writer lock
- OS mediates access to a file
- 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

OPERATING SYSTEMS

File Types – Name, Extension

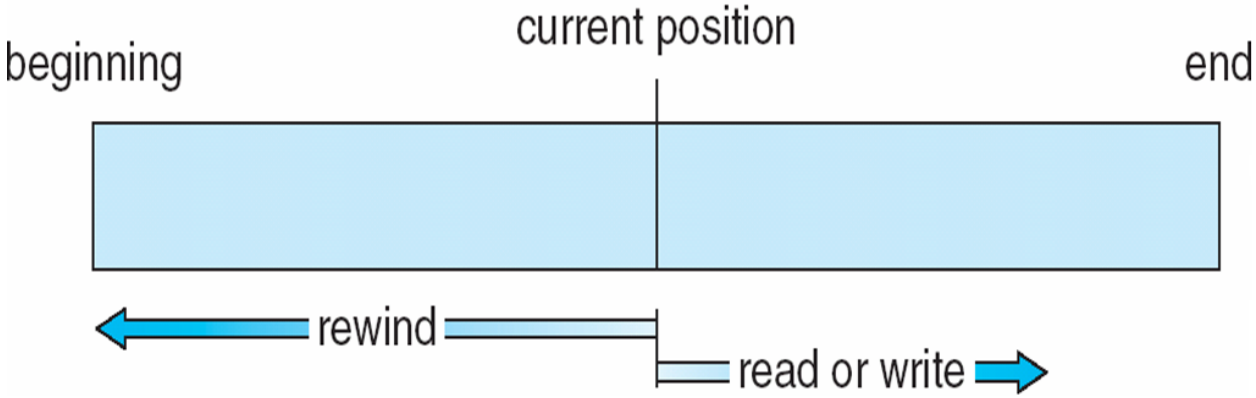
file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual 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

OPERATING SYSTEMS

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

Access Methods



Sequential-access File

- **Sequential Access**

  read next
  write next
  reset

- **Direct Access (or relative access)** – file is fixed length logical records

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

  n = relative block number (i.e. index relative to the beginning of the file)

- Relative block numbers allow OS to decide where file should be placed

Simulation of Sequential Access on Direct-access File

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

- cp = current position
- Some systems allow only sequential file access; others allow only direct access

Other Access Methods

- Can be built on top of base methods
- General involve creation of an index for the file
- Keep index in memory for fast determination of location of data to be operated on (consider UPC code plus record of data about that item with associated prices)
- If index file becomes too large, create index for the index file
  - Primary index file contains pointers to the secondary index files, which point to the actual data items
- IBM indexed sequential-access method (ISAM)
  - Small master index, points to disk blocks of secondary index (which points to the actual file blocks)
  - File kept sorted on a defined key
  - Binary search of the master index provides the block number of the secondary index and another binary search to find the block containing the desired record

THANK YOU

Suresh Jamadagni
Department of Computer Science Engineering

sureshjamadagni@pes.edu