

# OPERATING SYSTEMS

---

## Storage Management

**Venkatesh Prasad**

Department of Computer Science

# OPERATING SYSTEMS

---

## File System

**Venkatesh Prasad**

Department of Computer Science

# OPERATING SYSTEMS

## Slides Credits for all 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 - 9<sup>th</sup> edition 2013 and some slides from 10<sup>th</sup> edition 2018
  2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9<sup>th</sup> 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

- Contiguous logical address space
- Types:
  - Data
    - ▶ numeric
    - ▶ character
    - ▶ binary
  - Program
- Contents defined by file's creator
  - Many types
    - ▶ Consider **text file, source file, executable file**

- **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
  - Many variations, including extended file attributes such as file checksum information kept in the directory structure

- File is an **abstract data type**
- **Create**
- **Write** – at **write pointer** location
- **Read** – at **read pointer** location
- **Reposition within 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

- Several pieces of data are needed to manage open files:
  - **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
  - Disk location of the file: cache of data access information
  - Access rights: per-process access mode information

- 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
- 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

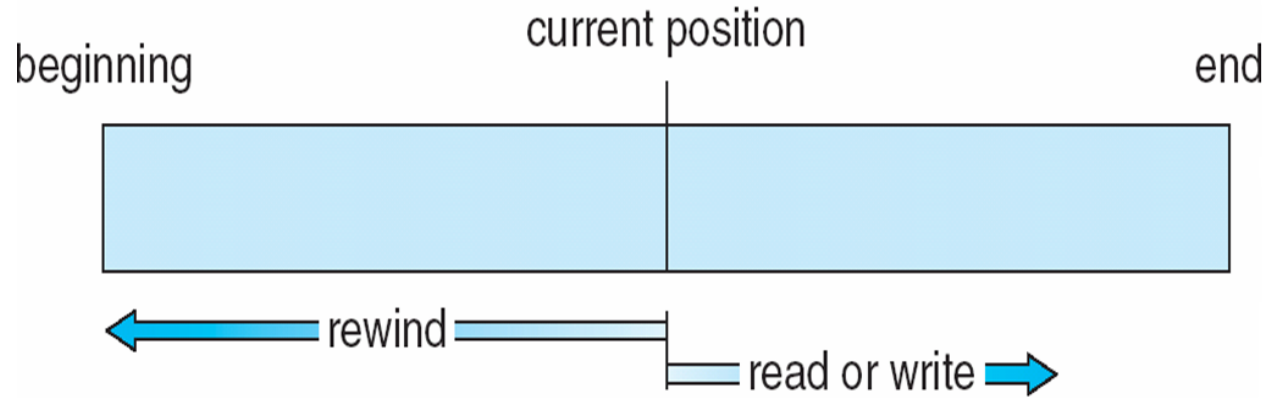
file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual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

- 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

# OPERATING SYSTEMS

## 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

sequential access	implementation for direct access
<i>reset</i>	<i>cp = 0;</i>
<i>read next</i>	<i>read cp;</i> <i>cp = cp + 1;</i>
<i>write next</i>	<i>write cp;</i> <i>cp = cp + 1;</i>

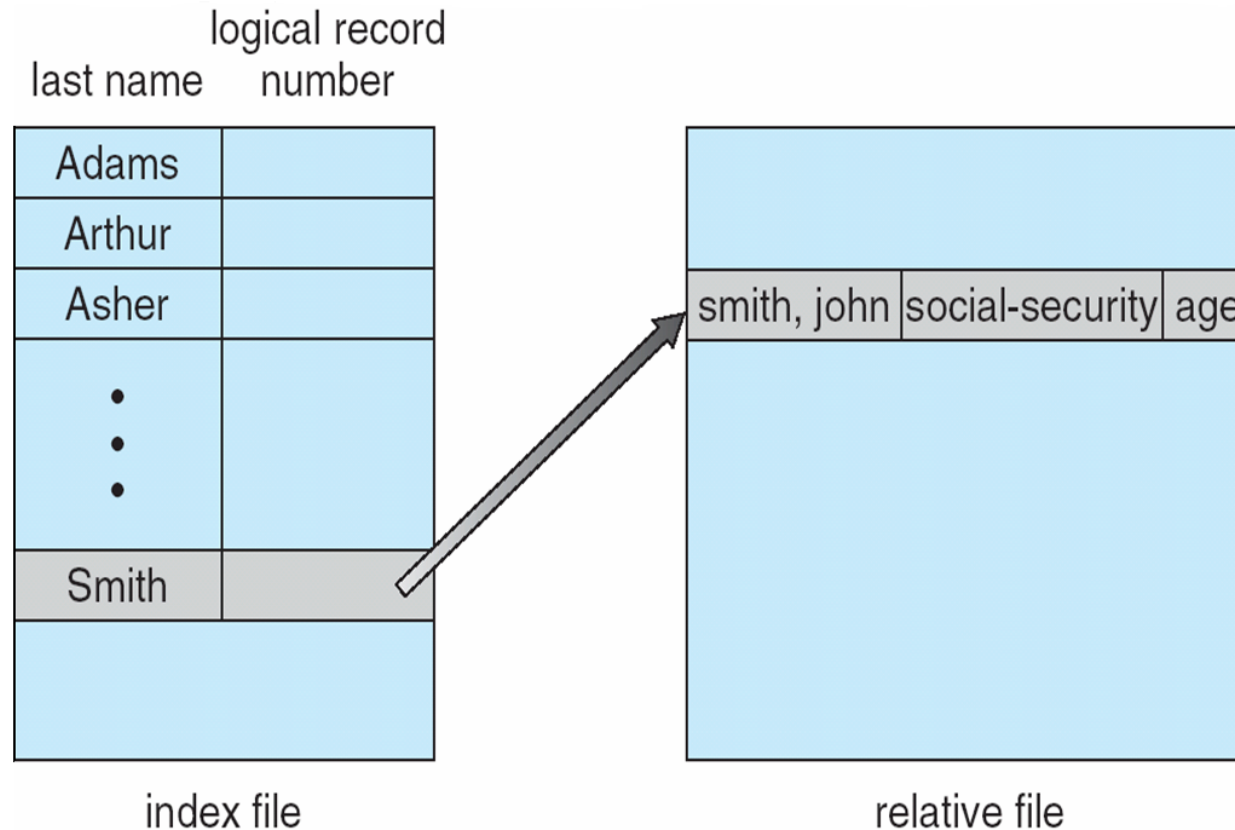
- *cp* = current position
- Some systems allow only sequential file access; others allow only direct access

- 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

# OPERATING SYSTEMS

## Example of Index and Relative Files

- VMS operating system provides index and relative files





# THANK YOU

---

**Venkatesh Prasad**

Department of Computer Science Engineering

**[venkateshprasad@pes.edu](mailto:venkateshprasad@pes.edu)**