Chapter 11

File Systems and Directories

Chapter Goals

- Describe the purpose of files, file systems, and directories
- Distinguish between text and binary files
- Identify various file types by their extensions
- Explain how file types improve file usage
- Define the basic operations on a file

Chapter Goals

- Compare and contrast sequential and direct file access
- Discuss the issues related to file protection
- Describe a directory tree
- Create absolute and relative paths for a directory tree
- Describe several disk-scheduling algorithms

File Systems

- **File** A named collection of related data
- **File system** The logical view that an operating system provides so that users can manage information as a collection of files
- **Directory** A named group of files

Text and Binary Files

- **Text file** A file in which the bytes of data are organized as characters from the ASCII or Unicode character sets
- **Binary file** A file that contains data in a specific format, requiring interpretation

Text and Binary Files

- The terms ***text file*** and ***binary file*** are somewhat misleading
- They seem to imply that the information in a text file is not stored as binary data
- Ultimately, all information on a computer is stored as binary digits
- These terms refer to how those bits are formatted: as chunks of 8 or 16 bits, interpreted as characters, or in some other special format

File Types

- Most files, whether they are in text or binary format, contain a specific type of information
  - For example, a file may contain a Java program, a JPEG image, or an MP3 audio clip
- The kind of information contained in a document is called the **file type**
  - Most operating systems recognize a list of specific file types

File Types

Extensions	File type
txt	text data file
mp3, au, wav	audio file
gif, tiff, jpg	image file
doc, wp3	word processing document
java, c, cpp	program source files

Figure 11.1 Some common file types and their extensions

- File names are often separated, usually by a period, into two parts
  - Main name
  - File extension
- The **file extension** indicates the type of the file

File Operations

- Create a file
- Delete a file
- Open a file
- Close a file
- Read data from a file
- Write data to a file
- Reposition the current file pointer in a file
- Append data to the end of a file
- Truncate a file (delete its contents)
- Rename a file
- Copy a file

File Access

- **Sequential access** Information in the file is processed in order, and read and write operations move the current file pointer as far as needed to read or write the data

  The most common file access technique, and the simplest to implement

File Access

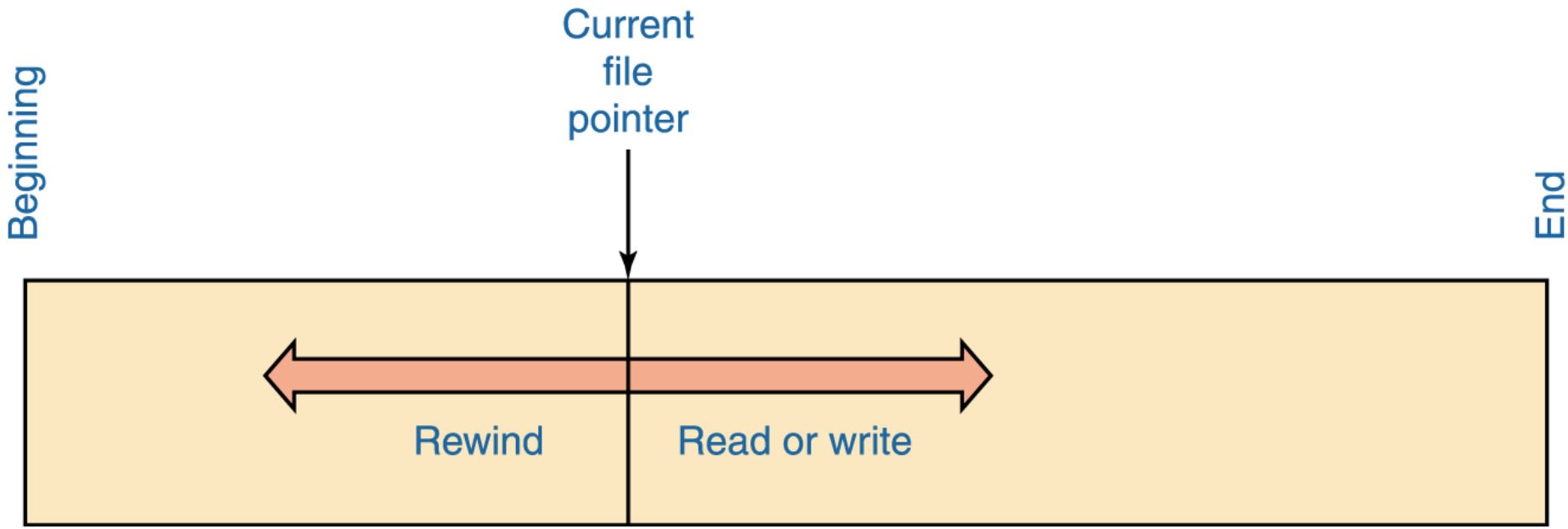


Figure 11.2 **Sequential file access**

File Access

- **Direct access** Files are conceptually divided into numbered logical records and each logical record can be accessed directly by number

File Access

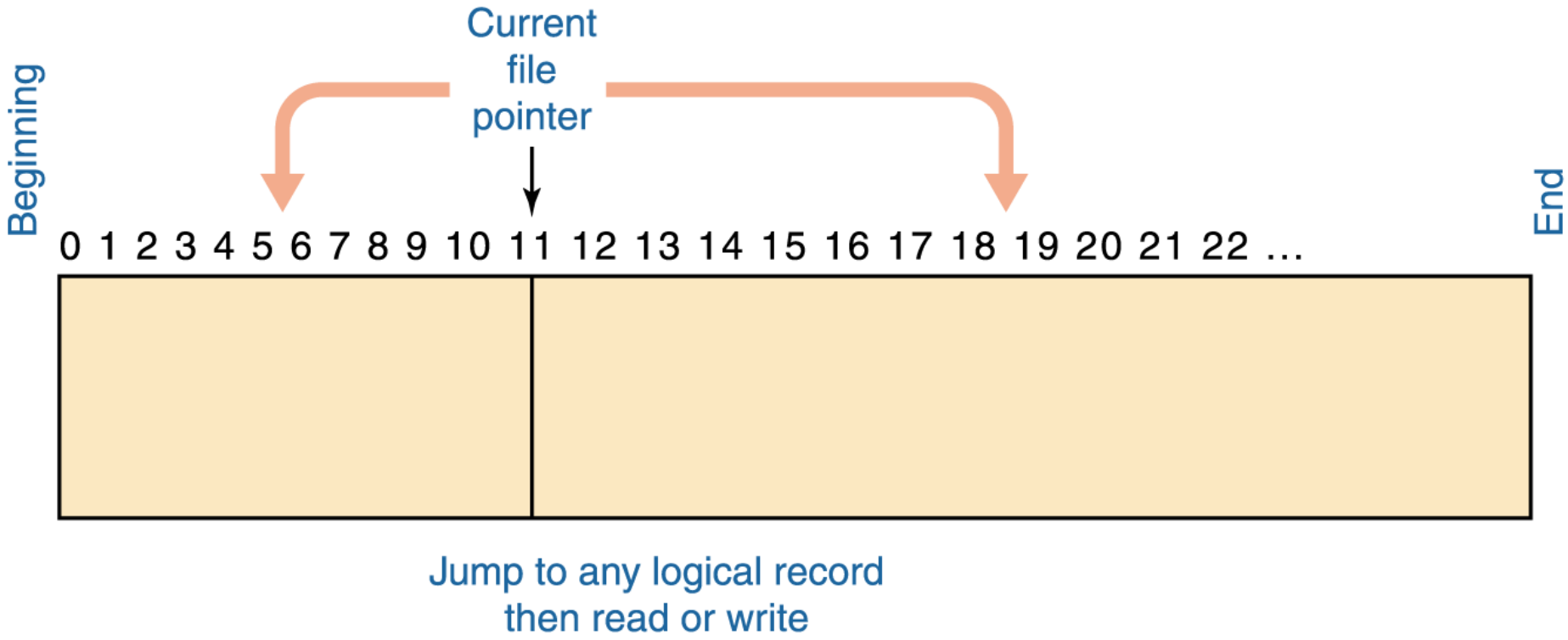


Figure 11.3 **Direct file access**

File Protection

- In multiuser systems, file protection is of primary importance
- We don’t want one user to be able to access another user’s files unless the access is specifically allowed
- A file protection mechanism determines who can use a file and for what general purpose

File Protection

- A file's protection settings in the Unix operating system is divided into three categories
  - **Owner**
  - **Group**
  - **World**

	Read	Write/Delete	Execute
Owner	Yes	Yes	No
Group	Yes	No	No
World	No	No	No

Directory Trees

- A directory of files can be contained within another directory

  The directory containing another is usually called the *parent directory*, and the one inside is called a *subdirectory*

- **Directory tree** A logical view of a file system; a structure showing the nested directory organization of a file system
- **Root directory** The directory at the highest level

Directory Trees

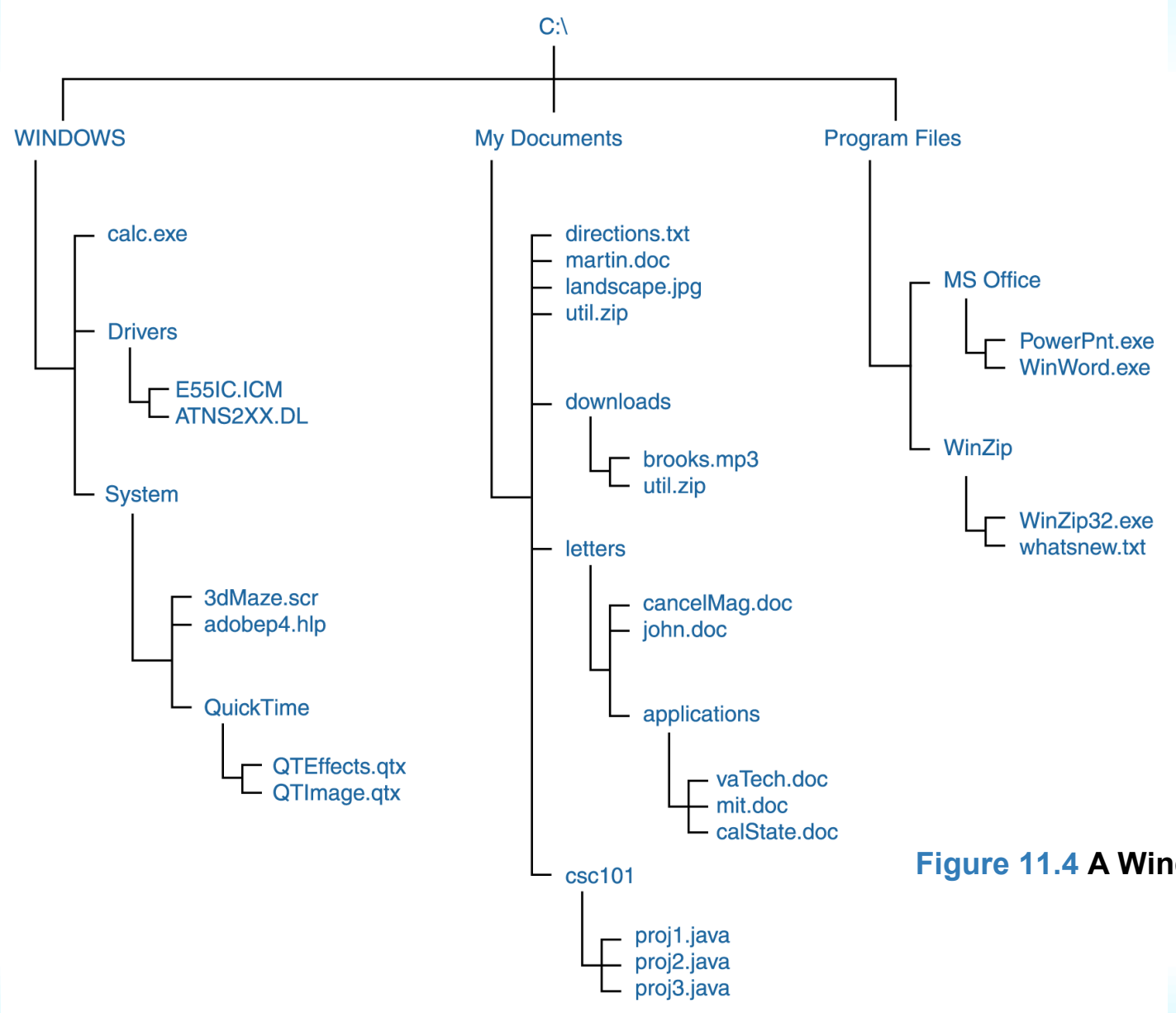
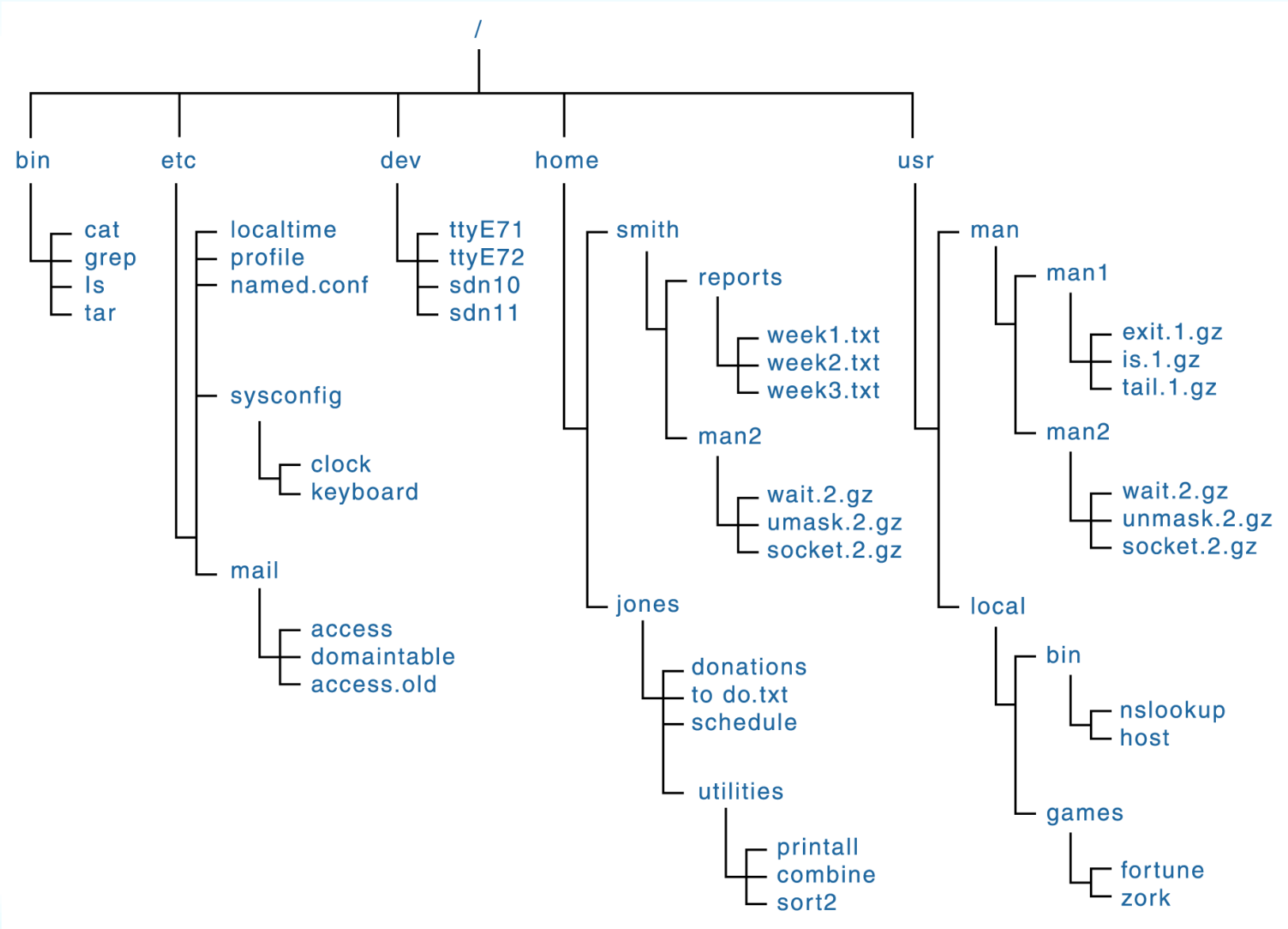


Figure 11.4 A Windows directory tree

Directory Trees

- At any point in time, you can be thought of as working in a particular location (that is, a particular subdirectory)
- **Working directory** The subdirectory in which you are working

A Unix Directory Tree



Path Names

- **Path** A text designation of the location of a file or subdirectory in a file system, consisting of the series of directories through which you must go to find the file
- **Absolute path** A path that begins at the root and specifies each step down the tree until it reaches the desired file or directory
- **Relative path** A path name that begins at the current working directory

Path Names

- Examples of absolute path

  C:\Program Files\MS Office\WinWord.exe

  C:\My Documents\letters\applications\vaTech.doc

  C:\Windows\System\QuickTime
- Suppose the current working directory is

  C:\My Documents\letters
- Then the following relative path names could be used

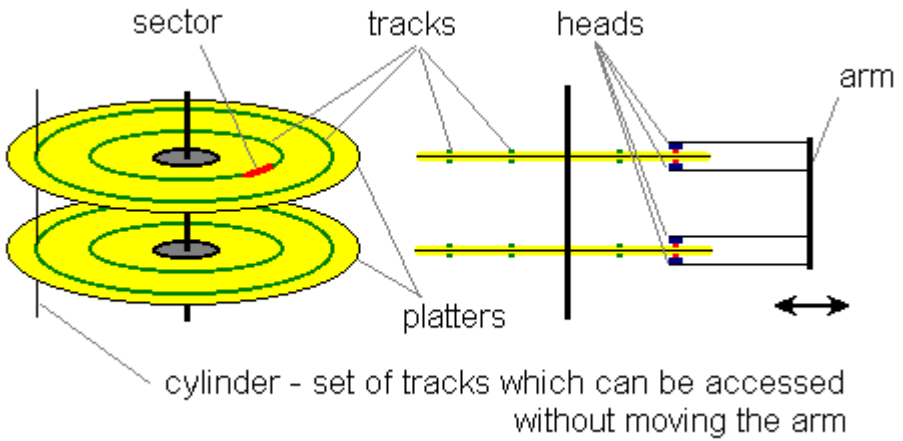
  cancelMag.doc

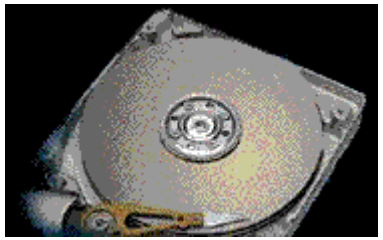
  applications\calState.doc

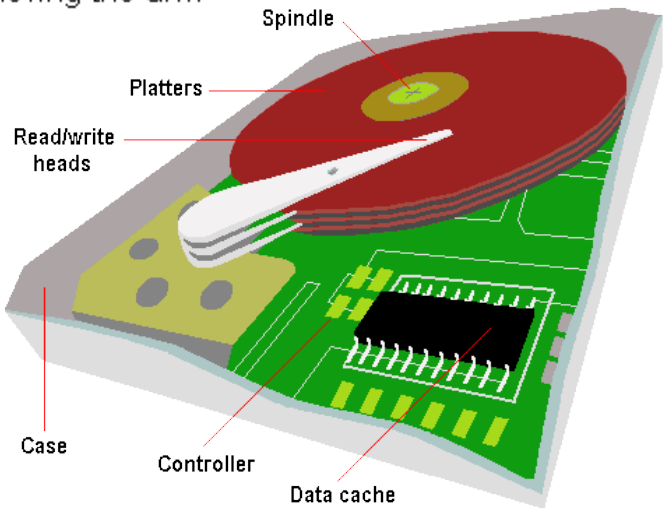
Disk Scheduling

- File systems must be accessed in an efficient manner
- As a computer deals with multiple processes over a period of time, a list of requests to access the disk builds up
- **Disk scheduling** The technique that the operating system uses to determine which requests to satisfy first

Disk Scheduling







Disk Scheduling

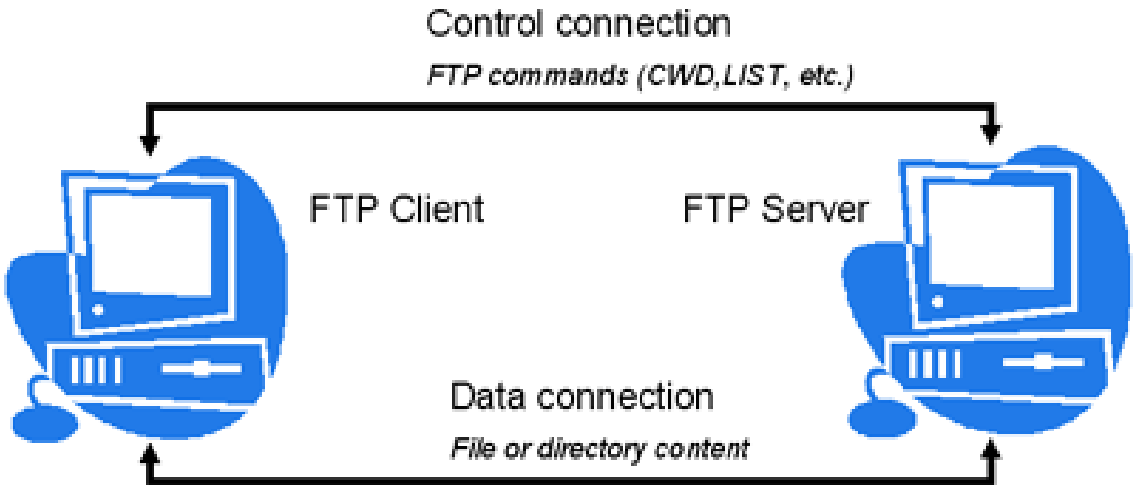
- **First-Come, First-Served** Requests are serviced in the order they arrive, without regard to the current position of the heads
- **Shortest-seek-time-first** (SSTF) Disk heads are moved the minimum amount possible to satisfy a pending request
- **Scan** Disk heads continuously move in and out servicing requests as they are encountered

Disk Scheduling

- SCAN Disk Scheduling works like an elevator
  - An elevator is designed to visit floors that have people waiting. In general, an elevator moves from one extreme to the other (say, the top of the building to the bottom), servicing requests as appropriate.
  - The SCAN disk-scheduling algorithm works in a similar way, except instead of moving up and down, the read/write heads move in toward the spindle, then out toward the platter edge, then back toward the spindle, and so forth.

How To Transfer Files

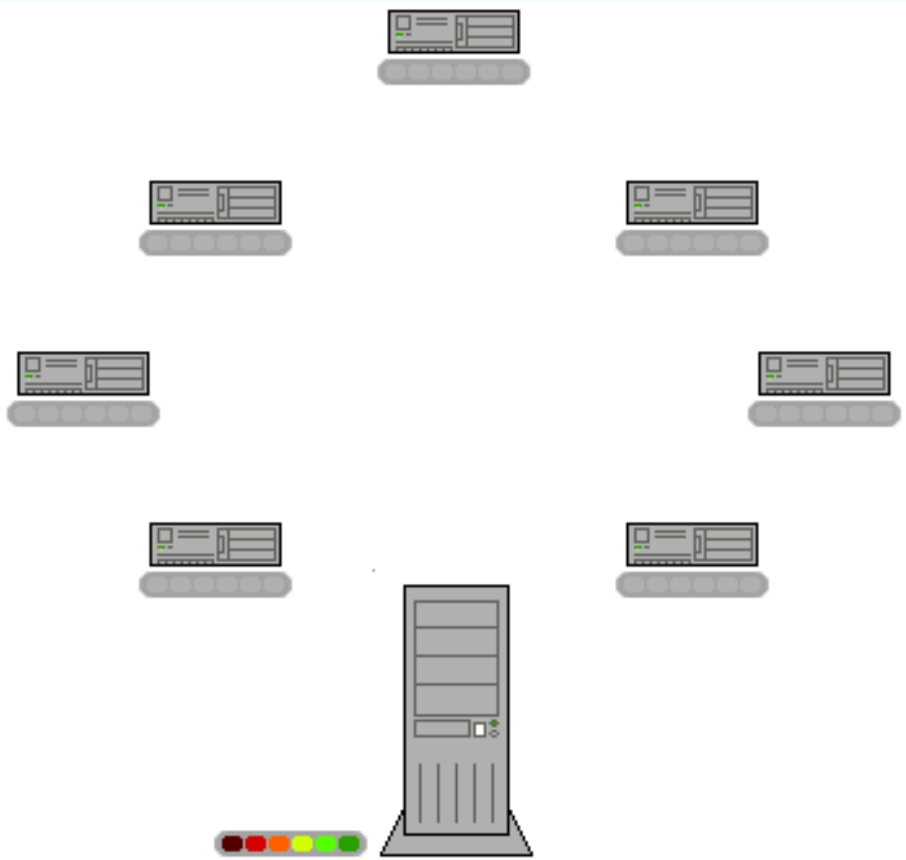
- FTP Clients
- FileZilla is one of the best



Other Types Of File Transfers & Systems

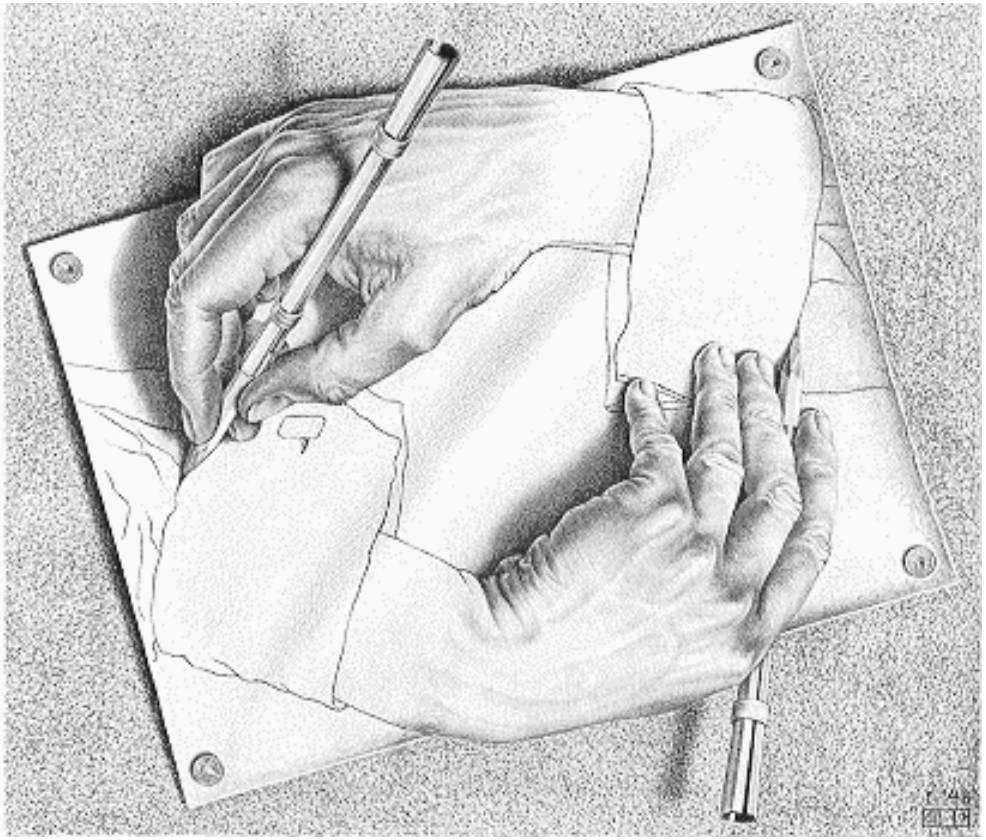
- Secure FTP (SSH, SFTP, FTPS)
- Network File System (NFS)
- Storage Area Networking (SANs)
- RAID Devices
- Peer-to-Peer (P2P)
- BitTorrent

BitTorrent



BitTorrent greatly reduces the load on seeders, because clients generally download the file from each other. In this animation, the colored bars beneath all of the clients represent individual pieces of the file. After the initial pieces transfer from the seed, the pieces are individually transferred from client to client. This demonstrates how the original seeder only needs to send out one copy of the file for all the clients to receive a copy. -from WikiPedia

A Little Hands On



Homework

- **Read Chapter Eleven**
- **Program Assignment #2 – Let Me Know If You Are Having Trouble**
- **Workshop Class On 11/20**

Have A Nice Night