

FILES

Introduction

- ▶ A file is a collection of related information defined by its creator.
- ▶ Files are mapped by the operating system onto physical mass-storage devices.
- ▶ A file system describes how files are mapped onto physical devices, as well as how they are accessed and manipulated by both users and programs.
- ▶ A file has a certain defined structure, which depends on its type.
 - ▶ A text file is a sequence of characters organized into lines (and possibly pages).
 - ▶ A source file is a sequence of functions, each of which is further organized as declarations followed by executable statements.
 - ▶ An executable file is a series of code sections that the loader can bring into memory and execute

File Attributes :

- ▶ A file's attributes vary from one operating system to another but typically consist of these:
 - ▶ Name : The symbolic file name is the only information kept in human readable form.
 - ▶ Identifier : This unique tag, usually a number, identifies the file within the file system : it is the non-human-readable name for the file.
 - ▶ Type : This information is needed for systems that support different types of files.
 - ▶ Location :This information is a pointer to a device and to the location of the file on that device.
 - ▶ Size. The current size of the file (in bytes, words, or blocks) and possibly the maximum allowed size are included in this attribute.
 - ▶ Protection :Access-control information determines who can do reading, writing, executing, and so on.
 - ▶ Timestamps and user identification . This information may be kept for creation, last modification, and last use. These data can be useful for protection, security, and usage monitoring.

File Operations

- ▶ A file is an abstract data type. To define a file properly, we need to consider the operations that can be performed on files.
- ▶ Creating a file :
 - ▶ Two steps are necessary to create a file.
 - ▶ First, space in the file system must be found for the file
 - ▶ Second, an entry for the new file must be made in a directory.
- ▶ Opening a file .
 - ▶ Rather than have all file operations specify a file name, causing the operating system to evaluate the name, check access permissions, and so on, all operations except create and delete require a file open() first.
 - ▶ If successful, the open call returns a file handle that is used as an argument in the other calls.
- ▶ Writing a file .
 - ▶ To write a file, we make a system call specifying both the open file handle and the information to be written to the file.
 - ▶ The system must keep a write pointer to the location in the file where the next write is to take place if it is sequential. The write pointer must be updated whenever a write occurs.

- ▶ **Reading a file .**
 - ▶ To read from a file, we use a system call that specifies the file handle and where (in memory) the next block of the file should be put. Again, the system needs to keep a read pointer to the location in the file where the next read is to take place, if sequential.
- ▶ **Repositioning within a file.**
 - ▶ The current-file-position pointer of the open file is repositioned to a given value. Repositioning within a file need not involve any actual I/O.
 - ▶ This file operation is also known as a file seek.
- ▶ **Deleting a file**
 - ▶ To delete a file, we search the directory for the named file. Having found the associated directory entry, we release all file space, so that it can be reused by other files, and erase or mark as free the directory entry.
- ▶ **Truncating a file**
 - ▶ The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged

File Access Methods

- ▶ Files store information.
- ▶ When it is used, this information must be accessed and read into computer memory. The information in the file can be accessed in several ways :
 - ▶ Sequential Access
 - ▶ Direct Access

a) Sequential Access

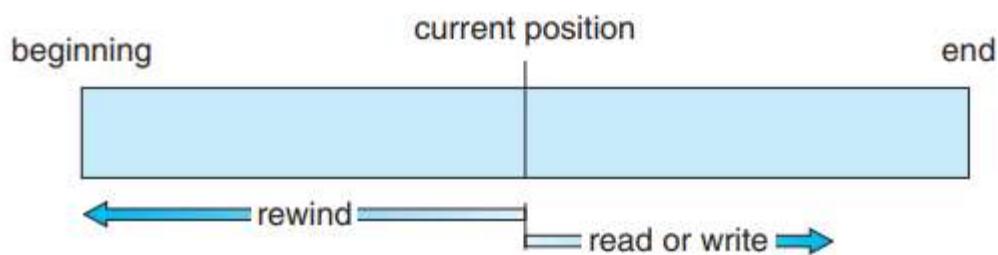


Figure 13.4 Sequential-access file.

- ▶ Information in the file is processed in order, one record after the other.
- ▶ This mode of access is by far the most common; for example, editors and compilers usually access files in this fashion.
- ▶ Reads and writes make up the bulk of the operations on a file.
- ▶ A read operation—`read next()`—reads the next portion of the file and automatically advances a file pointer, which tracks the I/O location.
- ▶ The write operation—`write next()`—appends to the end of the file and advances to the end of the newly written material (the new end of file)

b) Direct Access

- ▶ Here, a file is made up of fixed-length logical records that allow programs to read and write records rapidly in no particular order.
- ▶ The direct-access method is based on a disk model of a file, since disks allow random access to any file block.
- ▶ For direct access, the file is viewed as a numbered sequence of blocks or records. Thus, we may read block 14, then read block 53, and then write block 7.
- ▶ There are no restrictions on the order of reading or writing for a direct-access file.
- ▶ Direct-access files are of great use for immediate access to large amounts of information. Databases are often of this type.

File sharing

- ▶ The ability to share files is very desirable for users who want to collaborate and to reduce the effort required to achieve a computing goal.
- ▶ Once multiple users are allowed to share files, the challenge is to extend sharing to multiple file systems, including remote file systems.
- ▶ To implement sharing and protection, the system must maintain more file and directory attributes than are needed on a single-user system.
- ▶ Although many approaches have been taken to meet this requirement, most systems have evolved to use the concepts of file (or directory) owner(or user) and group.
- ▶ The owner is the user who can change attributes and grant access and who has the most control over the file.
- ▶ The group attribute defines a subset of users who can share access to the file.
- ▶ For example, the owner of a file on a UNIX system can issue all operations on a file, while members of the file's group can execute one subset of those operations, and all other users can execute another subset of operations. Exactly which operations can be executed by group members and other users is definable by the file's owner