Chapter 10

- Modern computer systems use ***disks*** as the primary on-line storage medium for information (both programs and data).
- The ***file system*** provides the mechanism for on-line storage of and access to both data and programs residing on the disks,
- A ***file*** is a collection of related information defined by its ***creator***.
- Files are normally organized into ***directories*** for ease of use.
- Because of all this device variation，one key goal of an operating system's I/O subsystem is to provide the simplest ***interface*** possible to the rest of the system.
- The file system is the most visible aspect of an operating system. It provides the mechanism for on-line ***storage*** of and ***access*** to both data and programs of the operating system.
- The file system consists of two distinct parts: a collection of ***files***, each storing related data, and a ***directory structure***, which organizes and provides information about all the files in the system.
- File ***name*** is the only information kept in ***human-readable*** form, and file ***identifier*** is the ***unique*** tag, usually a number, identifies the file within the file system, it is the ***non-human-readable*** name for the file.
- The minimal set of required file operations includs ***create, write, read, reposition, delete,*** and ***truncate***.
- A common technique for implementing file types is to include the type as part of the file name. The name is split into two parts-a ***name*** and an ***extension***, usually separated by a period character.
- Application programs also use ***extensions*** to indicate file ***types*** in which they are interested.
- Disk systems typically have a well-defined block size determined by the size of a ***sector***.
- All disk I/O is performed in units of one ***block*** (physical record), and all blocks are the ***same*** size.
- The UNIX operating system defines all files to be simply streams of ***bytes***. Each byte is individually addressable by its ***offset*** from the beginning (or end) of the file. In this case, the ***logical*** record size is 1 byte.
- The ***logical*** record size, ***physical*** block size, and ***packing*** technique determine how many logical records are in each physical block.
- The basic I/O functions operate in terms of ***blocks***.
- Because disk space is always allocated in ***blocks***, some portion of the last block of each file is generally wasted. The waste incurred to keep everything in units of blocks (instead of bytes) is ***internal*** fragmentation.
- All file systems suffer from ***internal*** fragmentation; the ***larger*** the block size, the ***greater*** the ***internal*** fragmentation.
- ***Sequential*** access is based on a tape model of a file and works as well on sequential-access ***devices*** as it does on random-access ones.
- The ***direct-access*** method is based on a ***disk*** model of a file, since disks allow random access to any file block.
- There are no restrictions on the ***order*** of reading or writing for a direct-access file.
- The block number provided by the user to the operating system is normally a ***relative*** block

number. A relative block number is an index relative to the ***beginning*** of the file. The first relative block of the file is ***0***.

Chapter 11

- The file system provides the mechanism for on-line ***storage*** and access to file contents, including ***data*** and programs.
- The file system resides permanently on ***secondary storage***, which is designed to hold a large amount of data permanently.
- To improve l/O efficiency, I/O transfers between memory and disk are performed in units of ***blocks***.
- A file system poses two quite different design problems. The first problem is defining the file system interface. This task involves defining a file and its attributes, the operations allowed on a file, and the directory structure for organizing files. The second problem is creating algorithms and data structures to ***map*** the logical file system onto the physical secondary-storage devices.
- The lowest level, the ***I/O control,*** consists of ***device drivers*** and ***interrupt handlers*** to transfer information between the main memory and the disk system.
- The device ***driver*** usually writes specific bit patterns to special locations in the I/O ***controller's*** memory to tell the ***controller*** which device location to act on and what actions to take.
- The ***basic file system*** needs only to issue generic commands to the appropriate device ***driver*** to read and write physical blocks on the disk.
- The ***file-organization*** module can translate logical block addresses to physical block addresses for the basic file system to transfer.
- Each file's logical blocks are numbered from 0 (or 1) through N.
- The ***file-organization*** module also includes the ***free-space*** manager, which tracks unallocated blocks and provides these blocks to the file-organization module when requested.
- The ***logical*** file system manages ***metadata*** information. Metadata includes all of the file-system structure except the actual *data* (or contents of the files).
- The ***logical*** file system manages the ***directory*** structure to provide the file organization module with the information the latter needs, given a symbolic file name. It maintains file structure via file-control blocks.
- A ***file-control*** block (FCB) contains information about the file, including ownership, permissions, and location of the file contents.
- The ***logical*** file system is also responsible for protection and security,
- On-disk, the file system may contain information about how to boot an ***operating*** system stored there, the total number of blocks, the ***number*** and ***location*** of free blocks, the ***directory*** structure, and individual ***files*****.**
- A ***boot control*** block can contain information needed by the system to boot an ***operating system*** from that volume.
- A ***volume control*** block contains volume (or partition) details, such as the number of blocks in the partition, size of the blocks, free block count and free-block pointers, and free FCB count and FCB pointers.
- A per-file ***FCB*** contains many details about the file, including file permissions, ownership, size, and location of the data blocks.

- Three major methods of allocating disk space are in wide use: ***contiguous***, ***linked***, and ***indexed***.
- ***Contiguous*** allocation requires that each file occupy a set of contiguous blocks on the disk.
- When head movement is needed (from the last sector of one cylinder to the first sector of the next cylinder), the head need only move from one track to the next. Thus, the number of disk seeks required for accessing ***contiguously*** allocated files is ***minimal***.
- Both ***sequential*** and ***direct*** access can be supported by ***contiguous*** allocation.
- There is no external fragmentation with ***linked*** allocation.
- It can be used effectively only for ***sequential-access*** files. it is inefficient to support a ***direct-access*** capability for ***linked-allocation*** files.
- ***Indexed*** allocation supports ***direct*** access, without suffering from ***external*** fragmentation.

Chapter 12

- Disk speed has two parts. The ***transfer rate*** is the rate at which data flow between the drive and the computer. The ***positioning time,*** sometimes called the random-access time, consists of the time to move the disk arm to the desired cylinder, called the ***seek*** time, and the time for the desired sector to rotate to the disk head, called the ***rotational latency.***
- The ***disk access time*** has two major. The ***seek*** time is the time for the disk arm to move the heads to the cylinder containing the desired sector. The ***rotational latency*** is the additional time for the disk to rotate the desired sector to the disk head.
- The ***disk bandwidth*** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.
- ***Low-level formatting*** fills the disk with a special data structure for each sector. The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.
- The main goal for the design and implementation of swap space is to provide the best ***throughput*** for the virtual memory system.

Chapter 13

- The two main jobs of a computer are I/O and ***processing***.
- The role of the operating system in computer I/O is to manage and control I/O ***operations*** and I/O ***devices***.
- The ***device drivers*** present a uniform ***device access interface*** to the I/O subsystem, much as system calls provide a standard interface between the application and the operating system.
- An I/O port typically consists of four registers, called the (1) ***status***, (2) ***control***, (3) ***data-in,*** and (4) ***data-out*** registers.
- The processor communicates with the controller using special I/O ***instructions*** or ***memory-mapped*** I/O.
- The ***special I/O instructions*** specify the transfer of a byte or word to an I/O port address. The I/O instruction triggers bus lines to select the proper device and to move bits into or out of a device register.
- The ***memory-mapped*** I/O, the device-control registers are mapped into the address space of the processor. The CPU executes I/O requests using the ***standard*** data-transfer instructions to read and write the device-control registers.

- The CPU performs a state save and jumps to the ***interrupt handler*** routine at a fixed address in memory.
- The ***interrupt handler*** determines the ***cause*** of the interrupt, performs the necessary ***processing***, performs a state ***restore***, and executes a ***return*** from interrupt instruction to return the CPU to the execution state prior to the interrupt.
- The device ***controller raises*** an interrupt by asserting a signal on the ***interrupt request line***, the ***CPU catches*** the interrupt and ***dispatches*** it to the interrupt ***handler***, and the ***handler clears*** the interrupt by ***servicing*** the device.
- Most CPUs have two interrupt request lines: the ***nonmaskable*** interrupt line, and the ***maskable*** interrupt line.
- The ***nonmaskable*** interrupt line is reserved for events such as unrecoverable memory errors.
- The ***maskable*** interrupt line can be turned off by the CPU before the execution of critical instruction sequences that must not be interrupted.
- The ***maskable*** interrupt is used by device controllers to request service.
- The purpose of the ***device-driver*** layer is to hide the differences among device ***controllers*** from the I/O subsystem of the kernel, much as the I/O system calls encapsulate the behavior of devices in a few generic classes that hide hardware differences from applications.
- The difference between ***nonblocking*** and ***asynchronous*** system calls is that a nonblocking read () returns ***immediately*** with whatever data are available--the full number of bytes requested, fewer, or none at all. An asynchronous readO call requests a transfer that will be performed in its entirety but that will complete at some ***future*** time.
- The kernel's ***I/O subsystem*** provides several services, including ***scheduling***, ***buffering***, ***caching***, ***spooling***, device ***reservation***, and ***error*** handling.
- A ***buffer*** is a memory area that stores data while they are transferred between two devices or between a device and an application.
- A ***cache*** is a region of fast memory that holds copies of data.
- A ***spool*** is a buffer that holds output for a device, such as a printer, that cannot accept interleaved data streams.