whole file may take up far greater space than that available in primary storage. It is a real situation in many cases (practically always so in the IR system). So let us note one more significant difference between primary and secondary storage: primary storage is significantly smaller than secondary storage.

Hence, some part of the file is read (copied) into primary storage. But which part of the file should be transferred? Should one ensure that the whole file, or a large part of it, is read from the disk into primary storage? Or is it more expedient to read just one record and then, upon its processing in CPU (in our case, upon comparison of the document profile held in the record with the available query formulation), read the next one and so on? To answer these questions we need to have some idea about the design of the disk.

Hard disks are made of sheets of metal cut into disks, usually between 3 and 18 inches in diameter. The disks are coated with a thin layer of magnetic material similar to that used on tape. Several disks are mounted on a common axis or hub to form a disk pack. Typically, a half-dozen to a dozen disks are mounted in one pack. All the surfaces of the pack except the top surface of the top disk and the bottom surface of the bottom disk are used for recording. The recording is done on each disk surface on a series of concentric tracks. The tracks are separate and do not form a spiral as do the grooves of a phonograph record. Each track is divided into a series of blocks. The blocks are separated by a gap. *A block is the unit of data that actually gets transferred (i.e., input or output [I/O] to or from the user's file that is located on a particular device.* A block is the smallest amount of data that can be transferred with a single I/O operation. A block may contain one or more records. In fact, a block may contain only a partial record, so that to access the entire record more than one block will need to be read.

The reading and writing of the track is performed by a *head,* like that in a record player. Because there are usually a few hundred tracks on each surface, but only one head, it is necessary to move the head from track to track to reach all the tracks. The piece of metal that moves the head in and out and keeps it at a constant height above the surface of the disk is called the *arm.* The total time required to transfer data to or from a disk has three components. First is the time required to reposition the arm, which is called *seek time.* This usually varies from a few milliseconds for a one-track move to scores of milliseconds or more for a move from one end of the disk to another. Most drives give a specification for an average, or random, move, which should be used in calculations.

The second component is the time from when the arm is positioned to when the first block of the transfer moves under a head. This time is called *latency* and depends on the speed of the disk's rotation. In the best case there would be no wait at all; in the worst case, such as when the beginning of the block had just been missed, the wait would be the time required for a full revolution. On the average, the latency is equal to the time required for the disk to turn one half of a revolution. The average latency is the time required for one half of a revolution, which is, in many real cases (disks), approximately 8.3 milliseconds.

The time needed for the read/write head to pass over a block is called the *block transfer time.* This is the third component. During the block transfer time, the data can be transferred from the disk to primary storage or from primary storage to the disk. Very often the manufacturer of the disk drive will provide figures for average seek time (in milliseconds), average rotational latency time (in milliseconds), and data transfer time (in bytes per milliseconds). The byte is a standard, fixed-length code representing some symbol (such as a letter or a digit) defined for the given computer. Usually the average seek time is longer than the average rotational latency time, whereas the average rotational latency time is longer, as a rule, than the average necessary to transfer one block of information, which usually consists of a few thousand bytes.

Now we can state the main point in comparing primary and secondary storage: the time needed for reading (and writing) data on disks is much larger than the time needed for moving information around once it has been transferred to primary storage. This is because mechanical movement (a disk is a mechanical device), rather than electronic movement (primary storage is an electronic circuit), is involved. In other words, relative to the other parts of a computer, disks are slow. But how slow are they? The time it takes to get information back from even relatively slow electronic primary storage is about 120 nanoseconds, or 120 billionths of a second. Getting the same information from a typical disk might take 30 milliseconds, or 30 thousandths of a second. To understand the size of this difference, we need an analogy. Assume that primary storage access is like finding something in the index of this book. Let us say that this local, book-in-hand access takes 20 seconds. Assume that disk access is like going to a library for the information you cannot find here in this book. Given that our "primary storage access" takes 20 seconds, how long does the "disk access" to the library take, keeping the ratio the same as that of a real primary storage access and disk access? The disk access is a quarter of a million times longer than the primary storage access. This means that getting information back from the library takes 5,000,000 seconds, or almost 58 days. Disks are very slow compared to primary storage, and this fact has a crucial effect on the performance of any programs (applications) using large quantities of information. How should this situation be handled? There is no way to avoid using a disk, because secondary storage is simply indispensable and the disk today is one of the best devices for secondary storage. So, what is to be done?

In computer science today the problem is successfully solved by different methods of enhancing performance. The use of blocks is one such method. Indeed, as was mentioned earlier, information on the disk is stored (written) in blocks and exchange of information between the disk and primary storage can be done only through a block, despite the fact that any program works only with records. So, why is "communication" with secondary storage maintained by blocks? Consider the following example. Let us assume that we have two files with one of the files containing only 1 record per block and another containing 10 records per block. If one file contains 1 million records, the second file would