



---

revised: 01/18/99, 01/16/00, 01/09/06, 08/22/08

---

## CHAPTER 3 — Computer Memory

Recall from chapter one that the major hardware components of a computer system are:

- Processor
- Main memory
- Secondary memory devices
- Input/output devices

This chapter looks at main and secondary memory.

### Chapter Topics:

- Bits and Bytes
- Kilobytes, megabytes, gigabytes
- Main memory and addresses
- Secondary memory
- Files

### QUESTION 1:

(Educated Guess: ) If you bought a new computer, about how much main memory would you expect it to have: 256 bytes, 256 kilobytes, or 256 megabytes?

---



---

## Answer:

A new PC should have at least 256 megabytes, and more if possible. Top-end PCs have several gigabytes of main memory.

# Characteristics of Computer Memory

Main memory is as vital as the processor chip to a computer system. Fast computer systems have both a fast processor and a large, fast memory. Here is a list of some characteristics of computer memory. Some characteristics are true for both kinds of memory; others are true for just one.

Characteristic	True for Main Memory	True for Secondary Memory
Very closely connected to the processor.		
Holds programs and data that the processor is actively working with.		
Used for long term storage.		
The processor interacts with it millions of times per second.		
The contents is easily changed.		
Relatively low capacity.		
Relatively huge capacity.		

Fast access.		
Slow access.		
Connected to main memory.		
Holds programs and data.		
Usually its contents are organized into <i>files</i> .		

## QUESTION 2:

What do you remember about computer memory? In the table, decide what is true for main memory and for secondary memory. (Click on the buttons to see if you are correct.)

---





---

**Answer:**

Click!

## Bit

In both main and secondary memory, information is stored as patterns of bits. Recall from chapter two what a bit is:

A **bit** is a single *on/off* value. Only these two values are possible.

The two values may go by different names, such as *true/false*, or *1/0*. There are many ways in which a bit can be *implemented*. For example a bit could be implemented as:

- A mechanical electrical switch (like a light switch.)
- Voltage on a wire.
- A single transistor (used in main memory).
- A tiny part of the surface of a magnetic disk.
- A tiny part of the surface of a magnetic tape.
- A hole punched in a card.
- A tiny part of the light-reflecting surface of a CD.
- Part of a radio signal.
- Many, many more ways

So the particular implementation of bits is different in main memory and secondary memory, but logically, both types of memory store bits.

**QUESTION 3:**



---

## Answer:

No. The different ways of storing a bit do not change the information that is being stored.

## Copied Information

Information stored in binary form does not change when it is copied from one medium (storage method) to another. And an unlimited number of such copies can be made (remember the advantages of binary.) This is a very powerful combination. You may be so accustomed to this that it seems commonplace. But when you (say) download an image from the Internet, the data has been copied many dozens of times, using a variety of storage and transmission methods.

It is likely, for example, that the data starts out on magnetic disk and is then copied to main storage of the web site's computer (involving a voltage signal in between.) From main storage it is copied (again with a voltage signal in between) to a network interface card, which temporarily holds it in many transistors. From there it is sent as an electrical signal down a cable. Along the route to your computer, there may be dozens of computers that transform data from an electrical signal, into main memory transistor form, then back to an electrical signal on another cable. Your data may even be transformed into a radio signal, sent to a satellite (with its own computers), and sent back to earth as another radio signal. Eventually the data ends up as data in your video card (transistors), which transforms it into a TV signal for your monitor.

The point of all of this is that the actual information (in this example the picture) does not change from one medium to the next.

## QUESTION 4:



## Answer:

Usually we don't want it to (and for audio data use "high fidelity" electronics to minimize the change.) But always some information is lost.

## Byte

One bit of information is so little that usually computer memory is organized into groups of eight bits. Each eight bit group is called a **byte**. When more than eight bits are required for some data, a whole number of bytes are used. One byte is about enough memory to hold a single character.

Name	Number of Bytes	power of 2
byte	1	$2^0$
kilobyte	1024	$2^{10}$
megabyte	1,048,576	$2^{20}$
gigabyte	1,073,741,824	$2^{30}$
terabyte	1,099,511,627,776	$2^{40}$

Often very much more than eight bits are required for data, and thousands, millions, or even billions of bytes are needed. These amounts have names, as seen in the table.

If you expect computers to be your career, it would be a good idea to become familiar with this table. The only number you should remember from the middle column is that a kilobyte is 1024 bytes. Often a kilobyte is called a "K", a megabyte is called a "Meg", and a gigabyte is called a "Gig".

## QUESTION 5:

How many 10 Meg files would it take to fill a 500 Gig hard drive?



---

**Answer:**

$$(500 \times 2^{30}) / (10 \times 2^{20}) == 50 \times 2^{10} == 50K \text{ files}$$

## Bytes, not Bits

The previous table listed the number of **bytes**, not bits. So one K of memory is 1024 bytes, or  $1024 * 8 == 8,192$  bits. Usually one is not particularly interested in the exact number of bits.

It will be very useful in your future career to be sure you know how to multiply powers of two.

$$2^M * 2^N = 2^{(M+N)}$$

In the above, \* means *multiplication*. For example:

$$2^6 * 2^{10} = 2^{16}$$

### QUESTION 6:

Locations in a digital image are specified by a row number and a column number (both of these are integers). Say that a particular digital image is 1024 rows by 1024 columns, and that each location holds one byte. How many megabytes are in that image?



---

Locations in a digital image are specified by a row number and a column number (both of them integers). A particular digital image is 1024 rows by 1024 columns, and each location holds one byte. How many megabytes are in that image?

---

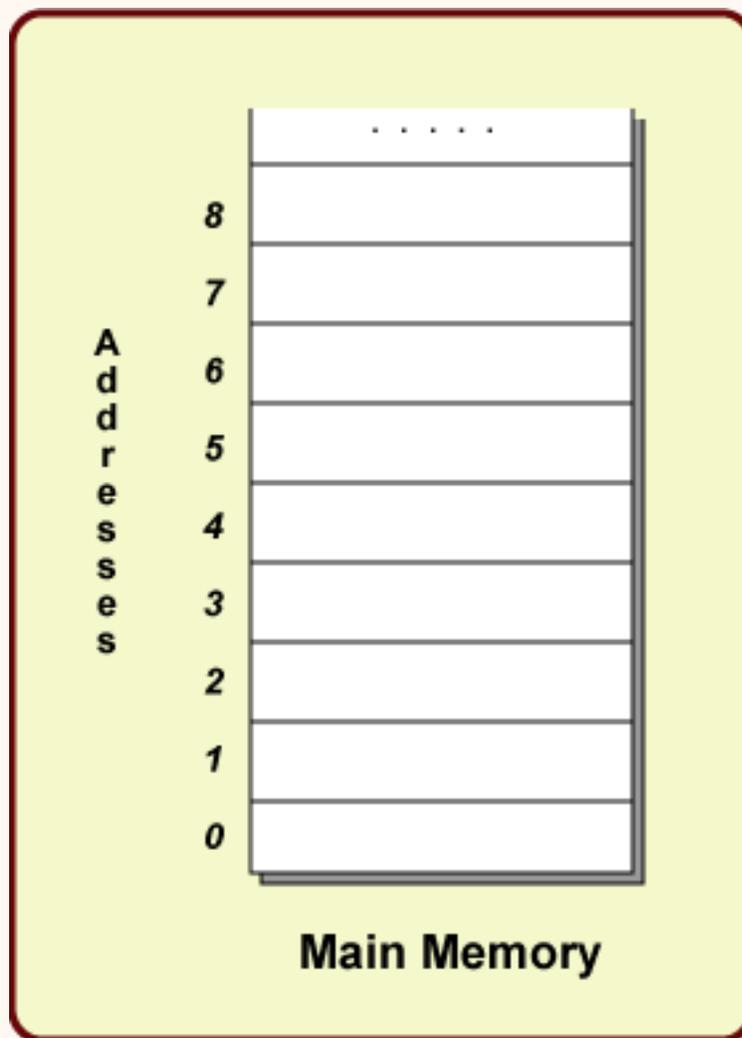
**Answer:**

$$1024 * 1024 = 2^{10} * 2^{10} = 2^{(10+10)} = 2^{20} = \text{one megabyte}$$

## Picture of Main Memory

Main memory consists of a very long list of bytes. In most modern computers, each byte has an **address** that is used to locate it. The picture shows a small part of main memory:

Each box in this picture represents a single byte. Each byte has an address. In this picture the addresses are the integers to the left of the boxes: 0, 1, 2, 3, 4, ... and so on. The addresses for most computer memory start at 0 and go up in sequence until each byte has an address.



Each byte contains a pattern of eight bits. When the computer's power is on, every byte contains some pattern or other, even those bytes not being used for anything. (Remember the nature of binary: when a binary device is working it is either "on" or "off", never inbetween.)

The address of a byte is not explicitly contained in memory. When the processor needs to access the byte at a particular address, the electronics of the computer "knows how" to find that byte in memory.

## QUESTION 7:



## Answer:

- Each byte of main memory has a unique address. **True**
- If main memory is 4 megabytes large, there will be  $4 * 2^{20}$  addresses. **True**

## Contents of Main Memory

Main memory (as all computer memory) stores bit patterns. That is, each memory location consists of eight bits, and each bit is either "0" or "1". For example, the picture shows the first few bytes of memory.

The *only* thing that can be stored at one memory location is eight bits, each with a value of "0" or "1". The bits at a memory location are called the *contents* of that location.

		.....
8		0100 1001
7		1100 1100
6		0110 1110
5		0110 1110
4		0000 0000
3		0110 1011
2		0101 0001
1		1100 1001
0		0100 1111

**Main Memory**

Sometimes people will say that each memory location holds an eight bit binary

number. This is OK, as long as you remember that the "number" might be used to represent a character, or anything else.

Remember that what a particular pattern represents depends on its context (ie., how a program is using it.) You cannot look at an arbitrary bit pattern (such as those in the picture) and say what it represents.

## QUESTION 8:

In the picture, what is the contents of:

- Memory location 0?
- Memory location 4?





---

## Answer:

In the picture, what is the contents of:

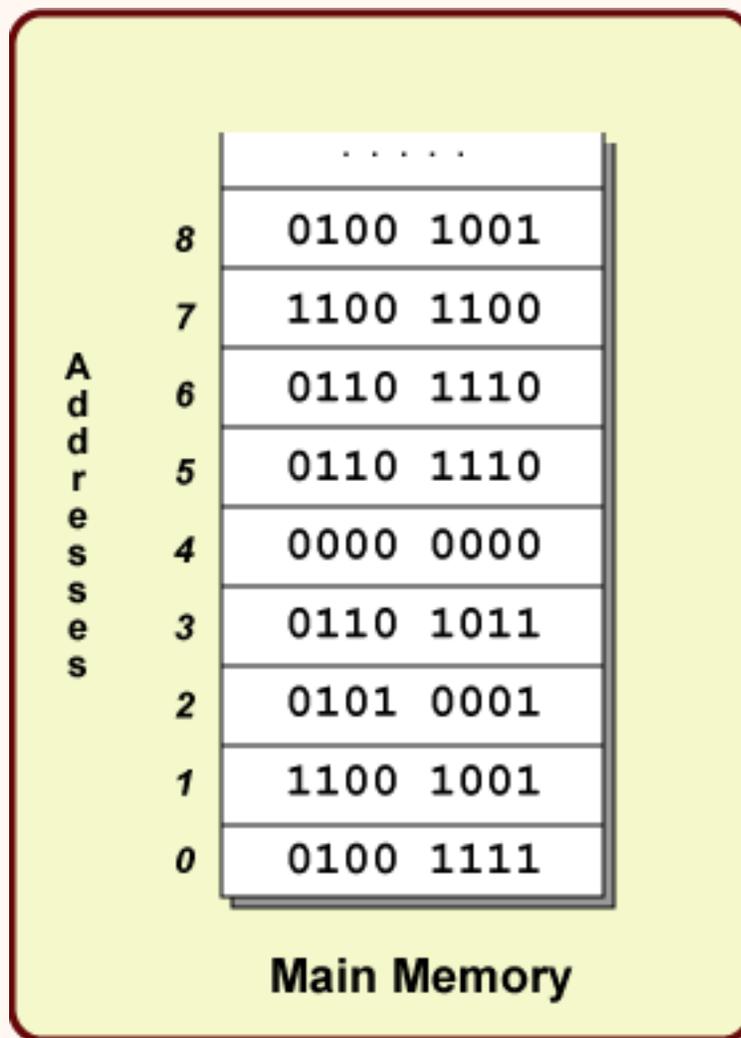
- Memory location 0? 0100 1111
- Memory location 4? 0000 0000

(It is customary to show bit patterns with a space every 4 bits. The space is not actually part of the pattern.)

# Reading and Writing Memory

The processor can do two fundamental things with in main memory:

1. It can **write** to a byte at a given



memory location.

- The previous bit pattern in that location will be destroyed.
  - The new bit pattern is now saved for future use.
2. It can **read** a byte from a given location.
    - The processor gets the bit pattern stored at that location.
    - The contents of that location are NOT changed.

For example, if the processor needs to get the byte stored at location 5, it can read it. It gets the byte "0110 1110" as the data it needs (but location 5 in memory does not change.)

Most processors can write (and read) more than a single byte at a time. This speeds things up. But the two operations above are fundamental. You may have heard talk about the new 64-bit processors and 128-bit processors. A 64-bit processor can read and write  $64/8 = 8$  bytes at a time.

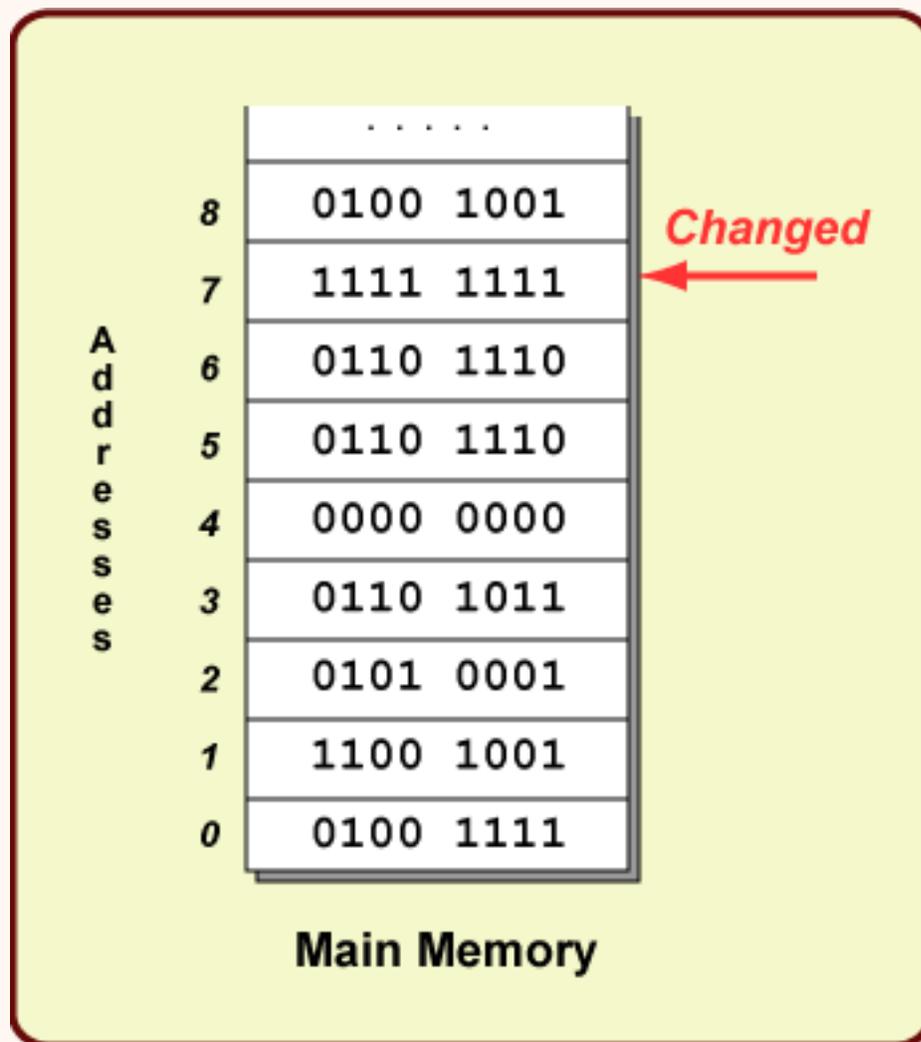


## Answer:

The answer is in the picture below.

# Programs and Memory

The processor has written a byte of data at location 7. The old contents of that location are lost. Main memory now looks like the picture.



When a program is running, it has a section of memory for the data it is using. Locations in that section can be changed as many times as the program needs. For example, if a program is adding up a list of numbers, the sum will be kept in main memory (probably using several bytes.) As new numbers are added to the sum, it will change and main memory will have to be changed, too.

Other sections of main memory might not change at all while a program is running. For example, the *instructions* that make up a program do not (usually) change as a program is running. The instructions of a running program are located in main memory, so those locations will not change.

When you write a program in Java (or most other languages) you do not need to keep track of memory locations and their contents. Part of the purpose of a programming language is to do these things automatically.

## QUESTION 10:

(Review Question:) Where are programs and data kept for long-term storage?

---



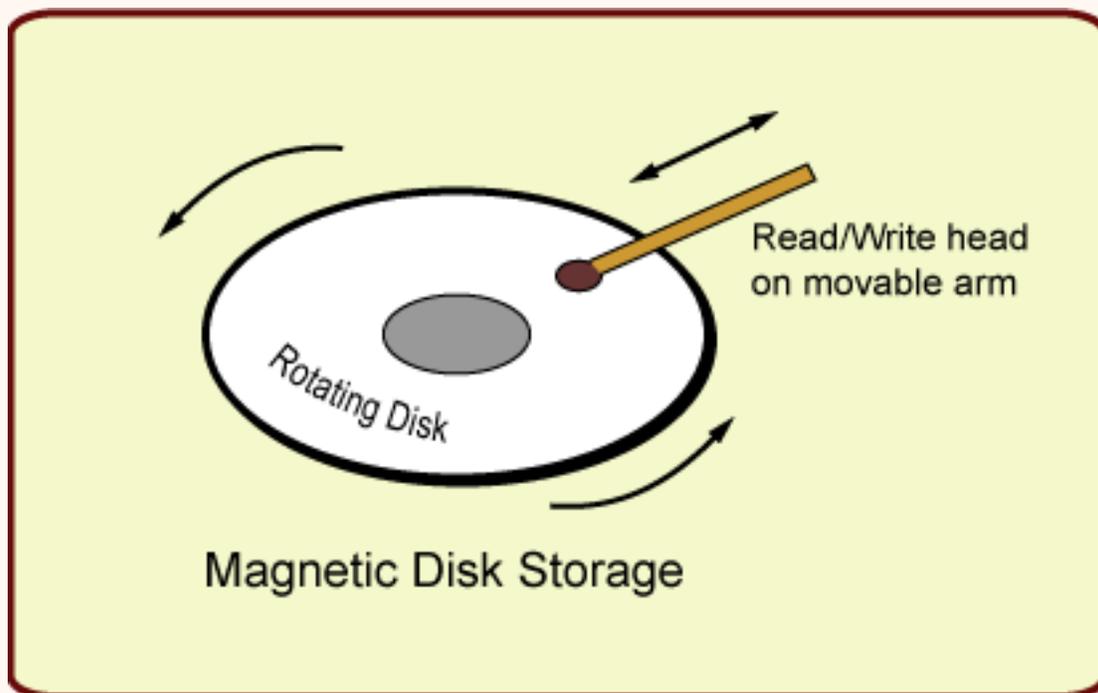


---

## Answer:

In secondary storage (usually the computer system's hard disk.)

# Hard Disks



The hard disk of a computer system records bytes on a magnetic surface much like the surface of audio tape. The recording (writing) and reading of the data is done with a *read/write head* similar to that used with audio tape.

The picture shows one disk and one read/write head at the end of a movable arm. The arm moves in and out along a radius of the disk. Since the disk is rotating it will record data in a circular track on the disk. Later on, to read the data, it must be moved to the right position, then it must wait until the rotating disk brings the data into position. Just as with audio tape, data can be read without changing it. When new data is recorded, it replaces any data that was previously recorded at that location. Unlike audio tape, the read/write head does not actually touch the disk but skims just a little bit above it.

Usually the component called the "hard disk" of a computer system contains many individual disks and read/write heads like the above. The disks are coated with magnetic material on both sides (so each disk gets two read/write heads) and the disks are all attached to one spindle. All the disks and heads are sealed into a dust-free metal can. Since the operation of a hard disk involves mechanical motion (which is much slower than electronic processes), reading and writing data is much slower than with main memory.

## QUESTION 11:

(Thought Question:) Why does not the read/write head of a hard disk touch its surface?

---





---

## Answer:

Since the disk is constantly rotating when the computer is on it would soon wear out if the read/write head were in contact with it.

# Files

Hard disks (and other secondary memory devices) are used for long-term storage of large blocks of information, such as programs and data sets. Usually disk memory is organized into **files**.

A **file** is a collection of information that has been given a name and is stored in secondary memory. The information can be a program or can be data.

The form of the information in a file is the same as with any digital information--it consists of bits, usually grouped into eight bit bytes. Files are frequently quite large; their size is measured in kilobytes or megabytes.

If you have never worked with files on a computer before you should study the documentation that came with your operating system, or look at a book such as *Windows for Dummies* (or whatever is appropriate for your computer.)

One of the jobs of a computer's operating system is to keep track of file names and where they are on its hard disk. For example, in DOS the user can ask to run the program DOOM like this:

```
C:\> DOOM.EXE
```

The "C:\>" is a prompt; the user typed in "DOOM.EXE". The operating system now has to find the file called DOOM.EXE somewhere on its hard disk. The program will be copied into main storage and will start running. As the program runs it asks for information stored as additional files on the hard disk, which the operating system has to find and copy into main memory.



---

## Answer:

Usually in a file in secondary storage. If the file does not already exist, the program will ask the operating system to create it.

# Files and the Operating System

Most collections of data outside of main storage are organized into files. Keeping track of all this information is one of the jobs of the operating system. If the computer is part of a network, keeping track of all the files on all the computers is a big job, and involves all the operating systems on the network.

Application programs (including programs that you might write) do not directly read, write, create, or delete files. Since the operating system has to keep track of everything, all other programs ask it to do file manipulation tasks. For example, say that a program has just calculated a set of numbers and needs to save them. The following might be how it does this:

1. **Program:** asks the operating system to create a file with a name *RESULTS.DAT*
2. **Operating System:** gets the request; finds an unused section of the disk and creates an empty file. The program is told when this has been completed.
3. **Program:** asks the operating system to save the numbers in the file.
4. **Operating System:** gets the numbers from the program's main memory, writes them to the file. The program is told when this has been completed.
5. **Program:** continues on with whatever it is doing.

So when an application program is running, it is constantly asking the operating system to perform file manipulation tasks (and other tasks) and

waiting for them to be completed. If a program asks the operating system to do something that will damage the file system, the operating system will refuse to do it. Modern programs are written so that they have alternatives when a requests is refused. Older programs were not written this way, and do not run well on modern computers.

### **QUESTION 13:**

In modern computer systems, only the operating system can directly do anything with disk files. How does this:

1. affect the security of the system?
2. affect computer games?
3. affect the ease in creating programs?

(You might not be certain about your answers, but please think about the questions.)

---





---

## Answer:

In modern computer systems, only the operating system can directly do anything with disk files. How does this:

1. affect the security of the system?
  - The security is increased because programs that try to do dangerous or stupid things to files can't. They have to ask the operating system, which will only do safe and sensible things.
2. affect computer games?
  - Older computer games did their file manipulation themselves without asking the operating system (old DOS was stupid and slow and many application programs ignored it.) Those games won't run on modern computers.
3. affect the ease in creating programs?
  - Program creation is easier because the work of dealing with files is done by the operating system.

## Types of Files

As far as the hard disk is concerned, all files are the same. At the electronic level, there is no difference between a file containing a program and a file containing data. All files are named collections of bytes. Of course, what the files are *used for* is different. The operating system can take a program file, copy it into main memory, and start it running. The operating system can take a data file, and supply its information to a running program when it asks.

Often then last part of a file's name (the *extension*) shows what the file is expected to be used for. For example, in `mydata.txt` the `.txt` means that the file is expected to be used as a collection of text, that is, characters. With `doom.exe` the `.exe` means that the file is an "executable," that is, a program that is ready to run. With `program1.java` the `.java` means that the file is a source program in the language Java (there will be more about source programs later on in these notes.) To the hard disk, each of these files is the same sort of thing: a collection of bytes.



---

## Answer:

Something will likely go wrong. The information contained in the file is not appropriate for a word processor.

# End of the Chapter

You have reached the end of the chapter. If you have trouble with your own memory, and need to review a topic, click on it in the list below to go to where it was discussed.

- [Types of computer memory](#) and their characteristics.
- [Bit.](#)
- [Byte, kilobyte, megabyte, gigabyte, terabyte.](#)
- [Multiplication rule](#) for exponents.
- [Picture of main memory.](#)
- [Contents of main memory,](#)
- [Two things](#) that the processor can do with main memory.
- [Picture of a hard disk.](#)
- [Files.](#)
- [Operating system and file I/O.](#)

You have reached the end of the chapter.

