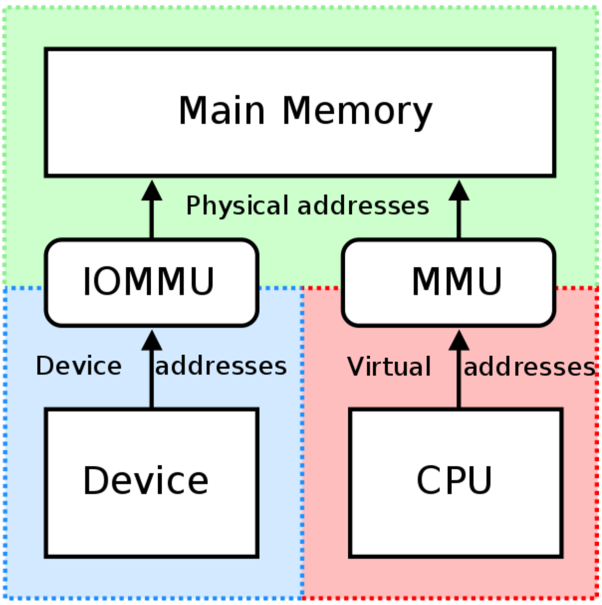
Contents

Background

There is such a wide variety of input and output devices and many different applications of those devices that leads to the reason why the I/O management is a difficult responsibility of an operating system to develop a general, consistent solution.



No matter how powerful the CPU is, a computer system's usefulness ultimately depends on its input and output facilities. Without I/O there is no possibility of keyboard input, of screen output, of printout, or even of disk storage and retrieval. Although you might be inclined to think of I/O in terms of user input and output, there would be no computer network or Internet access either. To the CPU and its programs, all these devices require specialised input and output processing facilities and routines

ASIDE:

A hard drive is a secondary storage device that consists of one or more platters to which data is written using a magnetic head, all inside of an air-sealed casing. Internal hard disks reside in a drive bay, connect to the motherboard using an ATA, SCSI, or SATA cable, and are powered by a connection to the PSU (power supply unit).

Data sent to and read from the hard drive is interpreted by the disk controller, which tells the hard drive what to do and how to move the components in the drive. Once the OS can determine file location and available write areas, the disk controller instructs the actuator to move the read/write arm and align the read/write head. Because files are often scattered throughout the platter, the head needs to move to different locations to access all information.

SSDs (solid-state drives) have started to replace HDDs (hard disk drives) because of the distinct performance advantages they have over HDD, including faster access times and lower latency. While SSDs is becoming more and more popular, HDDs continue to be used in many desktop computers largely due to the cost of HDDs versus SSDs. However, more and more laptops are beginning to utilise SSD over HDD, helping to improve the reliability and stability of laptops.

Much of the notes below relate to HDD.

Design objectives of the I/O facility in an OS

Two objectives are paramount in designing the I/O facility: efficiency and generality

1. **Efficiency:**

Efficiency is important because I/O operations often form a bottleneck in a computing system. We can see above that most I/O devices are extremely slow when compared with main memory and the processor. One way to tackle this is multiprogramming, which allows some processes to be waiting on I/O operations while another process is executing. Swapping is used to bring in additional ready processes to keep the processor busy (but this in itself is an I/O operation). Therefore, disk I/O needs to be efficient.

2. **Generality:**

In the interest of simplicity and minimum errors, devices should be all handled in the same way – the way in which processes view the I/O devices and the way in which the OS manages I/O devices and operation. This is difficulty because of the variety of device characteristics.

So a hierarchical, modular approach is taken and this approach hides most of the details of the device I/O in lower-level routines so that the user processes and upper levels of the OS see devices in terms of functions i.e. read, write, open, close, lock, unlock

Device Types

We differentiate roughly between two types of devices: communication and storage

1. ***communication devices***
    a. input devices

    Those that accept data generated by an external agent and transform data into a binary format to be stored and processed in the computer memory. The most common ones include keyboards, scanner and pointing device.

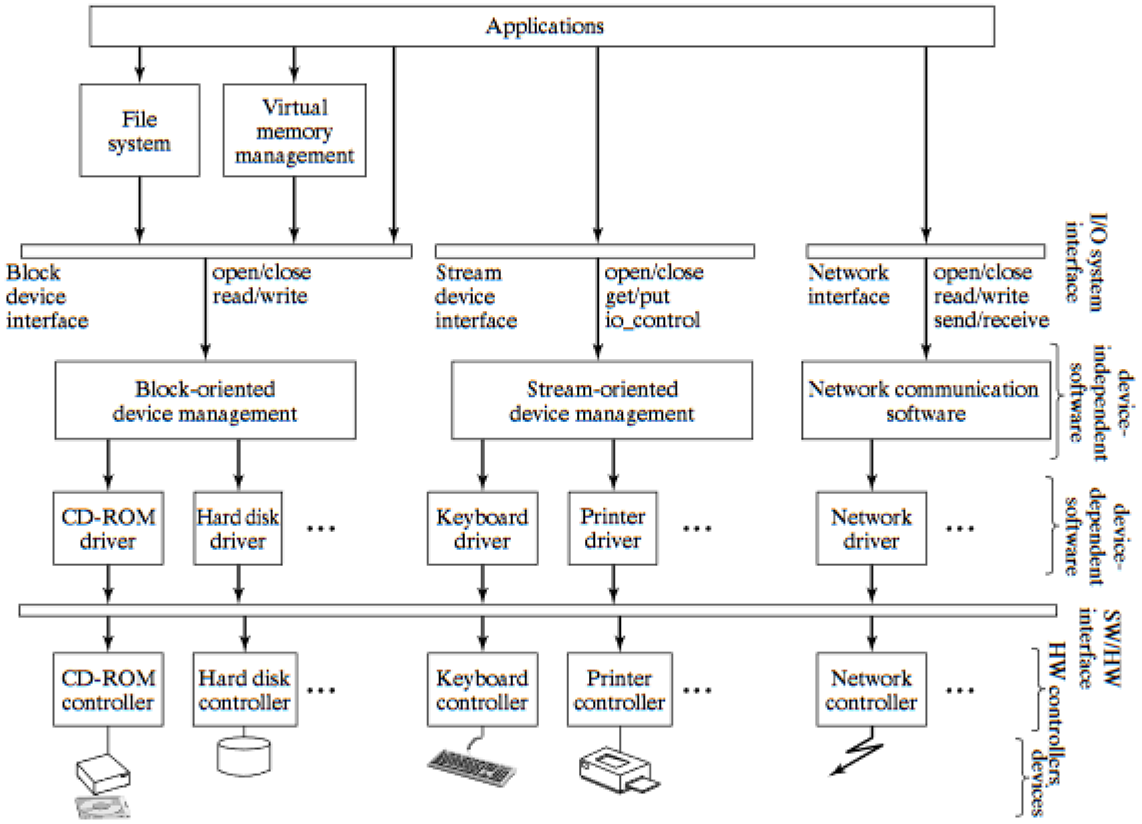
    Many of these devices provide a human-computer interface (HCI). When a computer is connected to other computers by a network, it can view the network as a communication device that can both produce input data and accept output data in various formats.

b. output devices

These accept binary data from a computer and transform these into other formats or media to make them available to an external agent (e.g., a human user, a physical process, or another computer). These include different types of printers, plotters, visual displays etc.

2. ***Storage devices***

Memory and any device that stores data and/or instruction fall under this category



For most business programs and for nearly every multimedia application, I/O is the predominant factor. E-commerce applications offer an even bigger challenge:

Web services generally require massive amounts of fast I/O to handle and process I/O requests as they occur. The speed at which most of these programs operate is determined by the ability of their I/O operations to stay ahead of their processing

LMC

We handled input and output in the Little Man Computer by providing input and output baskets for that purpose. Each input instruction transferred one three-digit data number from the input basket to the calculator; similarly, each output instruction transferred one data number from the calculator to the output basket. If we wanted to input three numbers, for example, an input instruction had to be executed three times. This could be done with three separate input instructions or in a loop, but either way, each individual piece of data required the execution of a separate input instruction

"Here comes the 64-bit local bus."

It is possible to transfer data between input and output devices and the CPU of a real computer in a similar manner. In the real computer, the in basket and out basket are commonly replaced by a bus interface that allows a direct transfer between a register within the CPU and a register within an I/O module that controls the particular device. Both input and output are handled similarly. The technique is known as ***programmed I/O***

Although the method of transferring data one word at a time does really exist the volume of data commonly transferred in I/O devices, such as disks and tapes, makes this method too slow and cumbersome to be practical as the only I/O transfer method in a modern high-speed machine.

We need to consider some method of transferring data in blocks rather than executing an instruction for each individual piece of data.

Also, in a real computer, there may be many input and output devices all trying to do I/O, sometimes at the same time. There needs to be a way of distinguishing and separating the I/O from these different devices.

Additionally, devices operate at different speeds from each other and from the CPU and synchronisation of these different operations must be achieved to prevent data loss.

Finally, it should be noted that I/O operations take up a lot of computer time. Even if a block of data can be transferred between the CPU and a disk with a single instruction, much time is potentially wasted waiting for the completion of the task. A CPU could execute millions of instructions in the time it takes a printer to print a single character. In a large modern computer, the number of I/O operations may be very large. It would be convenient and useful to be able to use the CPU for other tasks while these I/O transfers are taking place

In the computer, several different techniques are combined to resolve the problem of synchronising and handling I/O between a variety of different I/O devices operating with different quantities of data at different speeds

Input Output (I/O) Devices

So far we have studied how resources like processor and main memory are managed.

We shall now examine the I/O management.

Humans interact with machines by providing information through I/O devices. Also, much of whatever a computer system provides as on-line services is essentially made available through specialised devices such as screen displays, printers, keyboards, mouse, etc.

Input from the keyboard is very slow because it is dependent on the speed of typing, as well as on the thought process of the user. There are usually long thinking pauses between bursts of input, but even during those bursts, the actual input requirements to the computer are very slow compared to the capability of the computer to execute input instructions. Thus, we must assume that if the computer is simply performing a single task, it will spend most of its time waiting for input from the keyboard

Contrast the I/O requirements of keyboards, screens, and printers with those of disks and DVDs.

Since the disk is used to store programs and data, it would be very rare that a program would require a single word of data or program from the disk. Disks are used to load entire programs or store files of data. Thus, disk data is always transferred in blocks, never as individual bytes or words and disks operate as very fast transfer rates

With the rapid increase of networking power in recent years, network interfaces have also become an important source of I/O. From the perspective of a computer, the network is just another I/O device. In many cases, the network is used as a substitute for a disk, with the data and programs stored at a remote computer and served to the local station. For the computer that is acting as a server, there may be a massive demand for I/O services. User interfaces such as Windows, which allow the transfer of graphical information from a computer to a display screen located elsewhere on the network, place heavy demands on I/O capability.

Clearly, management of all these devices can affect the throughput of a system. For this reason, I/O management also becomes one of the primary responsibilities of an operating system.

Categories of Communication of External Devices

When we analyse external device that engage in I/O with computer systems, it can be noticed that this communication is required at the following three levels:

1. **The need for a *human* to input information and receive output from a computer.**

These may be character-oriented devices like a keyboard or an event-generating device like a mouse.

Usually, human input using a keyboard will be a few key depressions at a time. This means that the communication is rarely more than a few bytes. Also, the mouse events can be encoded by a small amount of information (just a few bytes). Even though a human input is very small, it is stipulated that it is very important, and therefore requires an immediate response from the system.

A communication which attempts to draw attention often requires the use of an interrupt mechanism or a programmed I/O data mode of operation.

2. **The need for a device to input information and receive output from a computer.**

The second kind of I/O requirement arises from devices which have a very high character density such as tapes, disks, sensors, controllers. With these characteristics, it is not possible to regulate communication with devices on a character by character basis.

The information transfer, therefore, is regulated in blocks of information. Additionally, sometimes this may require some kind of format control to structure the information to suit the device and/or data characteristics.

For instance, a disk drive differs from a line printer or an image scanner. For each of these devices, the format and structure of information is different.

It should be observed that the rate at which a device may provide data and the rates at which an end application may consume it may be considerably different. In spite of these differences, the OS should provide uniform and easy to use I/O mechanisms. Usually, this is done by providing a ***buffer***. The OS manages this buffer so as to be able to comply with the requirements of both the producer and consumer of data.

3. **The need for computers to communicate (receive/send information) over networks.**

The third kind of I/O requirements originate from the need to negotiate system I/O with the communications infrastructure.

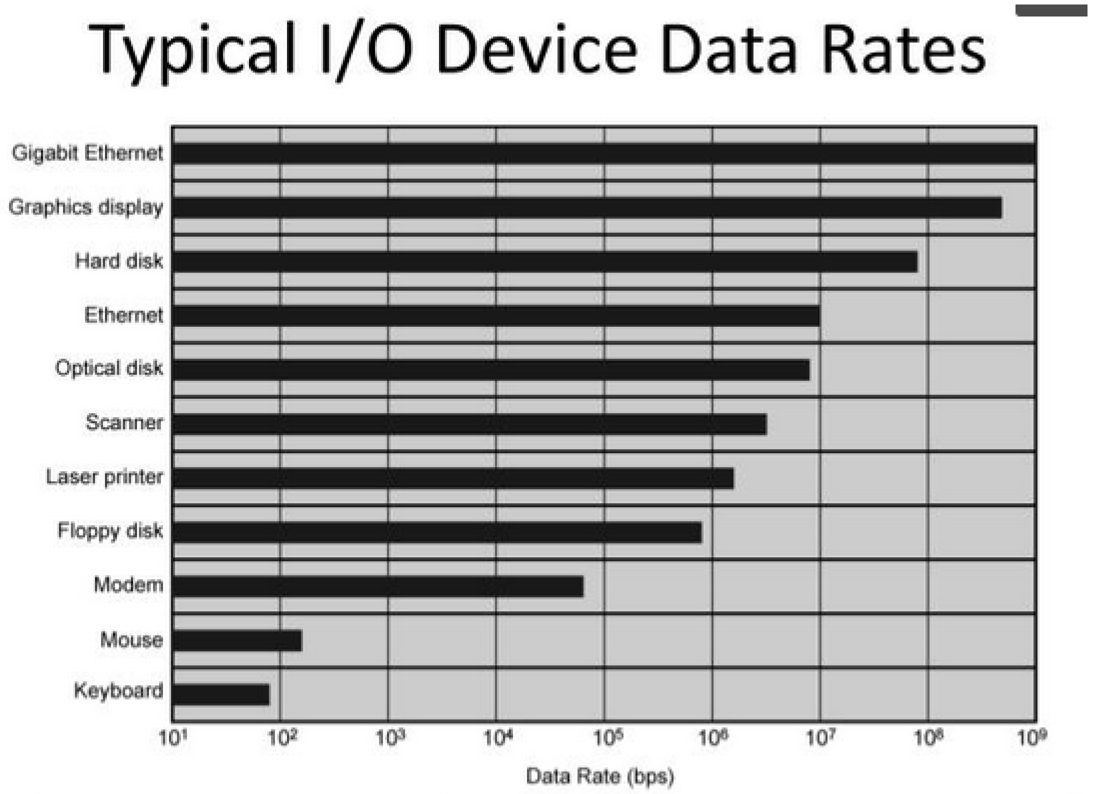
The system should be able to manage communications traffic across the network. This form of I/O facilitates access to internet resources to support e-mail, file-transfer amongst machines or Web applications. Additionally now we have a large variety of options available as access devices. These access devices may be in the form of mobile phones which have infrared or wireless enabled communications. This rapidly evolving technology makes these forms of communications very challenging.

Typically the character-oriented devices operate with speeds of tens of bytes per second (for keyboards, voice-based input, mouse, etc.).

Different devices operate over a very wide range. Printers operate at a slow speed for example ______ per second, disks transfer at much faster rates of _____ per second or more. The graphics devices fall between these two ranges while the graphics cards may in fact be even faster.

There are great differences across and within these categories such as:

- **Data Rate**: there are massive differences between the data rates of I/O devices. as can be seen here:



- **Application**: The use to which a device is put has an influence on the software and policies in the O.S. and supporting utilities.

For example: A disk used for file requires the support of file- management software; A disk used as a backing store for pages in a virtual memory scheme depends on the use of virtual memory hardware and software; A terminal can be used by the system administrator or regular user. These use imply different levels of privilege and priority in the O.S.

- **Complexity of control**: A printer requires a relatively simple control interface. A disk is much more complex.
- **Unit of transfer**: Data may be transferred as a stream of bytes or characters or in large blocks.
- **Data representation**: Different data-encoding schemes are used by different devices, includes differences in character code and parity conventions.
- **Error conditions**: The nature of errors, the way in which they are reported, their consequences, and the available range of responses differ widely from one device to another.

For all of these reasons, achieving a ***uniform*** and ***consistent*** approach to I/O from the OS's perspective and the user processes perspective is complicated.