Introduction to Operating System

Chapter · November 2015

DOI: 10.13140/RG.2.1.3438.5365

1 author:

Qasim Mohammed Hussein
Tikrit University
56 PUBLICATIONS **16** CITATIONS
SEE PROFILE

Lecture Notes on Operating System

Assistant Prof. Dr.

Qasim Mohammed Hussein

Introduction to operating

An **operating system** (OS) is a set of programs that control the execution of application programs and act as an intermediary between a user of a computer and the computer hardware. OS is software that manages the computer hardware as well as providing an environment for application programs to run.

Examples of OS are: Windows, Windows/NT, OS/2 and MacOS.

Operating system Objectives

The objectives of OS are:

(1) To make the computer system convenient and easy to use for the user.

(2) To use the computer hardware in an efficient way.

(3) To execute user programs and make solving user problems easier.

Computer System

A computer system can be divided into four components: the hardware, the operating system, the application programs and the users. The abstract view of system components is shown in figure 1.

1. **Hardware**: such as CPU, memory and I/O devices.
2. **Operating system**: provides the means of proper use of the hardware in the operations of the computer system, it is similar to government.
3. **Application programs**: solve the computing problems of the user, such as : compilers, database systems and web browsers.
4. **Users:** peoples, machine, or other computer.

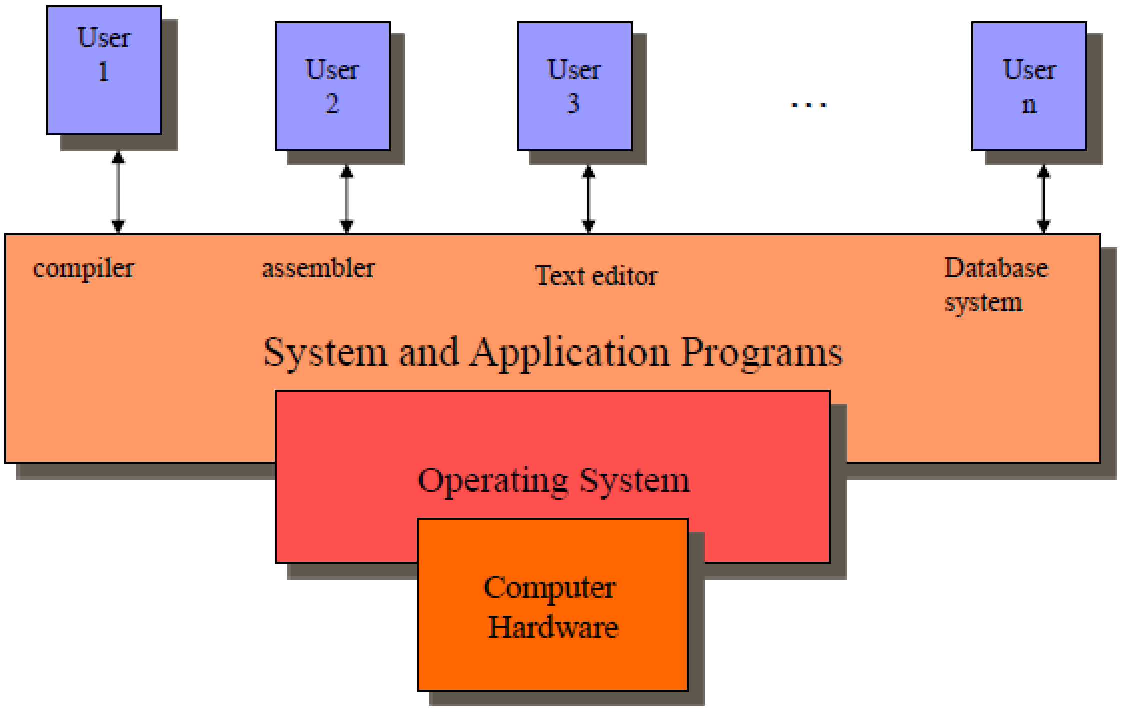


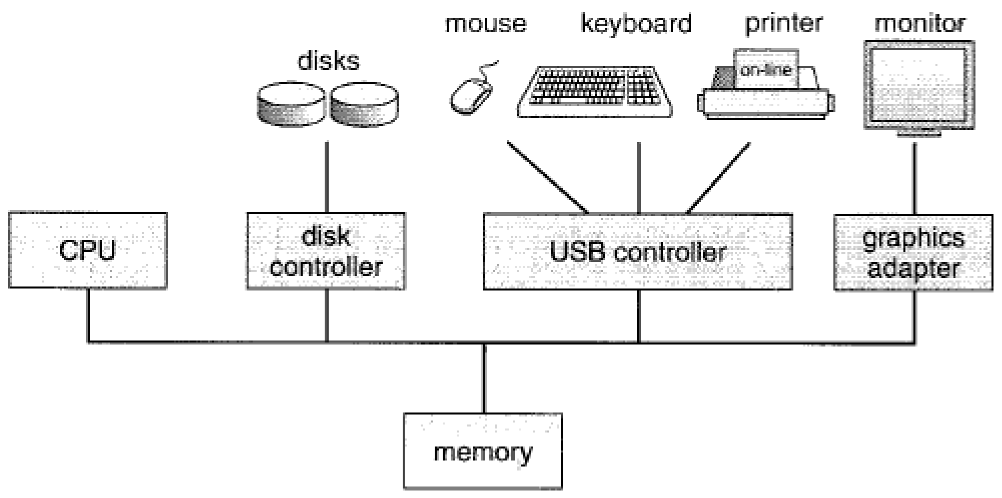
Figure 1: computer system

Computer System Organization

1. Computer-System Operation

A modern general-purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access to shared memory (Figure 1.2). Each device controller is in charge of a specific type of device (for example, disk drives, audio devices, and video displays). The CPU and the device controllers can execute concurrently, competing for memory cycles. To ensure orderly access to the shared memory, a memory controller is synchronizing access to the memory.

For a computer to start running-for instance, when it is powered up or rebooted-it needs to have an initial program to run. This initial program, or **bootstrap program**, tends to be simple. Typically, it is stored in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM),known by the general term **firmware**, within the computer hardware. It initializes all aspects of the system, from CPU registers to device controllers to memory contents. The bootstrap program must know how to load the operating system and to start executing that system. To accomplish this goal, the bootstrap program must locate and load into memory the operating system kernel. The operating system then starts executing the first process, such as "init," and waits for some event to occur.



A modern computer system.

Figure 2

2. Storage Structure

Computer programs must be in main memory (also called RAM) to be executed. Main memory is the only large storage area that the processor can access directly. It forms an array of memory words. Each word has its own address. Interaction is achieved through a sequence of load or store instructions to specific memory addresses. The load instruction moves a word from main memory to an internal register within the CPU, whereas the store instruction moves the content of a register to main memory.

The **instruction-execution cycle** includes:

1) Fetches an instruction from memory and stores that instruction in the instruction register. And increment the PC register.
2) Decode the instruction and may cause operands to be fetched from memory and stored in some internal register.
3) Execute the instruction and store the result in memory.

The programs and data are not resided in main memory permanently for the following two reasons:

1) Main memory is usually too small to store all needed programs and. Data permanently.
2) Main memory is a *volatile* storage device that loses its contents when power is turned off or otherwise lost.

Thus, most computer systems provide secondary storage as an extension of main memory to hold large quantities of data permanently.

The wide variety of storage systems in a computer system can be organized in a hierarchy (figure 2). **The main differences among the various storage systems lie in speed, cost, size, and volatility**. The higher levels are expensive, but they are fast.

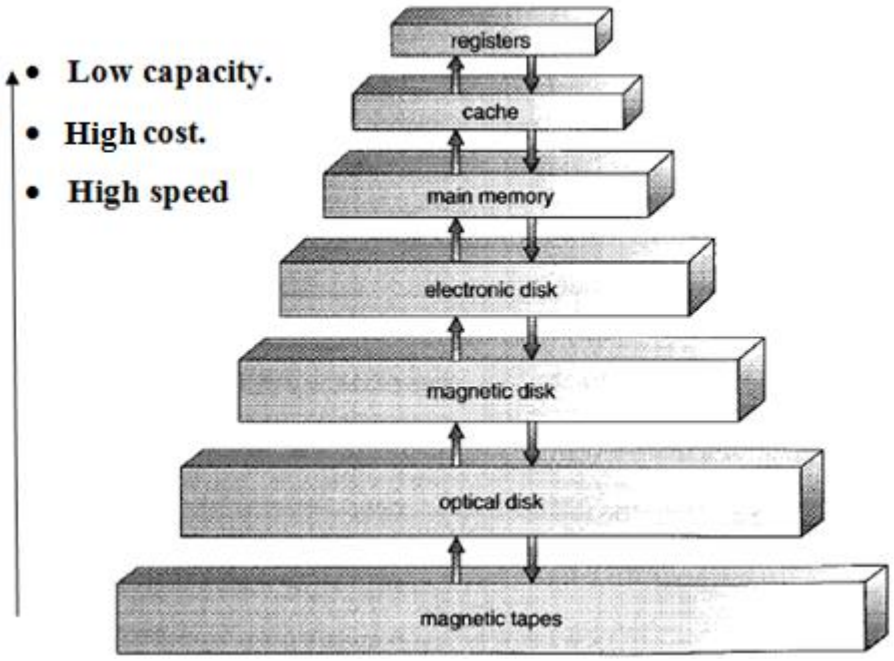


Figure 3: Storage device hierarchy

3. I/O Structure

A computer system consists of **CPUs and multiple device controllers** that are connected through a common bus. **The device controller** is responsible for moving the data between the peripheral devices that it controls and its local buffer storage. Typically, operating systems have a device driver for each device controller.

To start an I/O operation, the device driver loads the appropriate registers within the device controller. The device controller examines the contents of these registers to determine what action to take. The controller starts the transfer of data from the device to its local buffer. Once the transfer of data is complete, the device controller informs the device driver via an **interrupt** that it has finished its operation. The device driver then returns control to the operating system . For other operations, the device driver returns status information.

For moving bulk data, direct memory access (DMA) is used. After setting up buffers, pointers, and counters for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU. Only one interrupt is generated per block, to tell the device driver that the operation has completed, rather than the one interrupt per byte generated for low-speed devices.

Computer system structure

There are different categories for designing a computer system according to the number processors used.

1. Single-processor system: there is one CPU for executing instructions.
2. Multiprocessor system: It contains two or more processors that share bus, clock, physical memory and peripheral devices. The advantages of multiprocessors are:
   a) Increase throughput.
   b) Economy scale (less cost).
   c) Increase reliability.
3. Clustered system: it consists of multiple computer systems connected by a local area network.

Operating system History

Operating systems have been evolving through the years. Following table shows the history of OS.

Generation	Year	Electronic devices used	Types of OS and devices
First	1945 – 55	Vacuum tubes	Plug boards
Second	1955 – 1965	Transistors	Batch system
Third	1965 – 1980	Integrated Circuit (IC)	Multiprogramming
Fourth	Since 1980	Large scale integration	PC

Operating system Functions

OS performs many functions such as:

1. Implementing user interface.
2. Sharing HW among users.
3. Allowing users to share data among themselves.
4. Preventing users from interfering with one another.
5. Scheduling resource among users.
6. Facilitating I/O operations.
7. Recovering from errors.
8. Accounting for resource storage.
9. Facilitating parallel operations.
10. Organizing data for secure and rapid access.
11. Handling network communications.

Operating system Categories

The main categories of modern OS may be classified into three groups which are distinguished by the nature of interaction that takes place between the computer and the user:

1. Batch system

In this type of OS, users submit jobs on regular schedule (e.g. daily, weekly, monthly) to a central place where the user of such system did not interact directly with computer system. To speed up the processing, jobs with similar needs were batched together and were run through the computer as a group. Thus, the programmer would leave the programs with the operator. The output from each job would send to the appropriate programmer. The major task of this type was to transfer control automatically from one job to the next.

Disadvantages of Batch System

1. Turnaround time can be large from user standpoint.
2. Difficult to debug program.

2. Time-Sharing System

This type of OS provides on-line communication between the user and the system, the user gives his instructions directly and receives intermediate response, and therefore it called interactive system.

The time sharing system allows many user simultaneously share the computer system. The CPU is multiplexed rapidly among several programs, which are kept in memory and on disk. A program swapped in and out of memory to the disk.

Time sharing system reduces the CPU ideal time. The disadvantage is more complex.

3. **Real time operating system**

Real Time System is characterized by supplying immediate response. It guarantees that critical tasks complete on time. This type must have a pre-known maximum time limit for each of the functions to be performed on the computer. Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application.

The airline reservation system is an example of this type.

Performance development of OS

1. **On-line and off-line operation**

A special subroutine was written for each I/O device called a device controller. Some I/O devices has been equipped for either on-line operation (they are connected to the processor), or off-line operations (they are run by control unit).

2. **Buffering**

A buffer is an area of primary storage for holding data during I/O transfer. On input, the data are placed in the buffer by an I/O channel, when the transfer is complete the data may be accessed the processor. The buffing may be single or double.

3. **Spooling (Simultaneously Peripheral Operation On-Line)**

Spooling uses the disk as a very large buffer. Spooling is useful because device access data that different rates. The buffer provides a waiting station where data can rest while the slower device catches up.

Spooling allows overlapping between the computation of one job and I/O of another job.

4. Multiprogramming

In multiprogramming several programs are kept in main memory at the same time, and the CPU is switching between them , thus the CPU always has a program to be execute. The OS begins to execute one program from memory, if this program need wait such as an I/O operation, the OS switches to another program. Multiprogramming increases CPU utilization. Multiprogramming system provide an environment in which the various system resources are utilized effectively, but they do not provide for user interaction with the computer system.

Advantages

a) High CPU utilization.

b) It appears that many programs are allotted CPU almost simultaneously.

Disadvantages

a) CPU scheduling is requires.

b) To accommodate many jobs in memory, memory management is required.

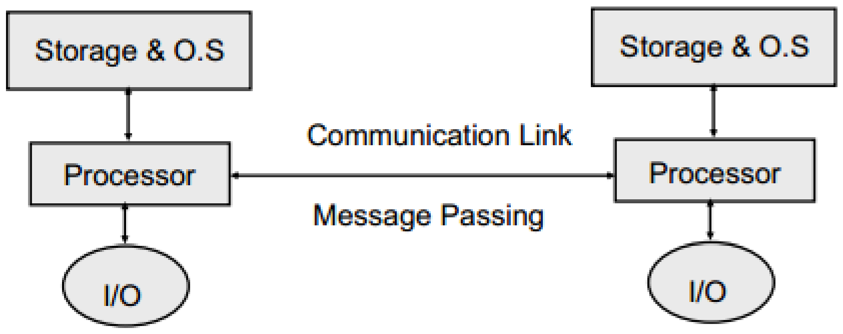
5. Parallel system

There are more than on processor in the system. These processors share the computer bus, clock, memory and I/O devices.

The advantage is to increase throughput (the number of programs completed in time unit).

6. Distributed system

Distribute the computation among several physical processors. It involves connecting 2 or more independent computer systems via communication link. So, each processor has its own O.S. and local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.



Advantages of distributed systems:

a) Resources Sharing – You can share files and printers.

b) Computation speed up – A job can be partitioned so that each processor can do a portion concurrently (load sharing).

c) Reliability – If one processor failed the rest still can function with no problem.

d) Communications – Such as electronic mail, ftp

7. Personal computer

Personal computers – computer system dedicated to a single user. PC operating systems were neither multi-user nor multi-tasking. The goal of PC operating systems were to maximize user convenience and responsiveness instead of maximizing CPU and I/O utilization.

•Examples: Microsoft Windows and Apple Macintosh

Operating system Service

An operating system provides services to programs and to the users of those programs. The common services provided by the operating system are:

1. Program execution: Operating system loads a program into memory and executes the program. The program must be able to end its execution, either normally or abnormally.
2. I/O Operation: I/O means any file or any specific I/O device. Program may require any I/O device while running. So operating system must provide the required I/O.
3. File system manipulation: Program needs to read a file or write a file. The operating system gives the permission to the program for operation on file.
4. Communication: Data transfer between two processes is required for some time. The both processes are on the one computer or on different computer but connected through computer network. Communication may be implemented by two methods:

   a. Shared memory

   b. Message passing.
5. Error detection: error may occur in CPU, in I/O devices or in the memory hardware. The operating system constantly needs to be aware of possible errors. It should take the appropriate action to ensure correct and consistent computing.

Operating system with multiple users provides efficient system operations:

1. Resource allocation: For simultaneously executing job.

2. Accounting: For account billing and usage statistics.
3. Protection: Ensure access to system resource is controlled.

Operating system operations

Modern operating systems are **interrupt driven**. If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen. Events are almost always signaled by the occurrence of an interrupt or a trap. **A trap** is a software-generated interrupt caused either *by an error* (for example, division by zero or invalid memory access) or by *a specific request from a user program* that an operating-system service be performed. For each type of interrupt, separate segments of code in the operating system determine what action should be taken. An interrupt service routine is provided that is responsible for dealing with the interrupt.

Since the operating system and the users share the hardware and software resources of the computer system, we need to make sure that an error in a user program could cause problems only for the one program that was running. With sharing, many processes could be adversely affected by a bug in one program. A properly designed ***operating system must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.***

A) Dual-Mode Operation

We must be able to distinguish between the execution of operating-system code and user defined code. The approach is to separate the two modes of operation**:** **user mode and kernel mode** (also called supervisor mode, system mode, or privileged mode). **A bit, called the mode bit** is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). The dual mode of operation provides us with the means for protecting the operating system from errant users-and errant users from one another.

System calls provide the means for a user program to ask the operating system to perform tasks reserved for the operating system on the user program's behalf.

B) Protection CPU

To ensure that the operating system maintains must control over the CPU. We must prevent a user program from **getting stuck in an infinite loop** or not calling system services and never returning control to the operating system. To accomplish this goal, we can use a **timer**. A timer can be set to interrupt the computer after a specified fixed or variable period.

Operating System Components

The operating system components are :

1. Process Management

In multiprogramming environment, OS decides which process gets the processor when and how much time. The operating system is responsible for the following activities in regard to process management:

1) Creating and deleting both user and system processes
2) Suspending and resuming processes
3) Providing mechanisms for process synchronization
4) Providing mechanisms for process communication
5) Providing mechanisms for deadlock handling

Memory Management

Main memory is a large array of words or bytes where each word or byte has its own address. The operating system is responsible for the following activities in regard to memory management:

1) Keeping track of which parts of memory are currently being used and by whom
2) Deciding which processes (or parts thereof) and data to move into and out of memory

3) Allocating and deallocating memory space as needed.

File system Management

The operating system is responsible for the following activities in regard to file management:

1) Creating and deleting files
2) Creating and deleting directories to organize files
3) Supporting primitives for manipulating files and directories
4) Mapping files onto secondary storage
5) Backing up files on stable (nonvolatile) storage media

Secondary storage Management

OS provides the following activities in connection with disk management:

1. Free-space management.
2. Storage allocation
3. Disk scheduling.

System Call and System Program

System calls provide an interface between the running program and the operating system. User cannot execute privileged instructions; the user must ask OS to execute them- system calls. System calls are implemented using traps.

OS gains control through trap, switches to kernel mode, performs service, switches back to user mode, and gives control back to user.

Example about how system calls are used from the OS to read data from one files and copy them to another file, shown in figure . the programmer never see this level of details

```
Acquire input filename
Write prompt to screen
Accept input
Acquire output filename
Write prompt to screen
Accept input
Open the input file
If file does not exist, abort
Create output file
If file exist, abort
Loop
    Read from input file
    Write to output file
Until read fail
Close output file
Write completion message on screen
Terminate normally
```

System program provide basic function to users, so that they don't need to write their own environment for program development (editors, compilers) and program execution (shells).

Protection and security

Protection refers to mechanism that control the access of programs or users to the both system and resources. The protection mechanism must:

1. Distinguish between unauthorized and authorized user.
2. Specify the control to be imposed.
3. Provide a means of enforcement.

Security measures are responsible for defending a computer system from external or internal attack.

Module Questions

There are many questionS the files:

1. Operating System: Questions and their answers: Processes and Deadlock (**Part 1**)
2. Operating system questions with their answers (Memory management, Virtual memory, Processes synchronization) **Part two.**

The link:

https://www.researchgate.net/profile/Qasim_Hussein/contributions