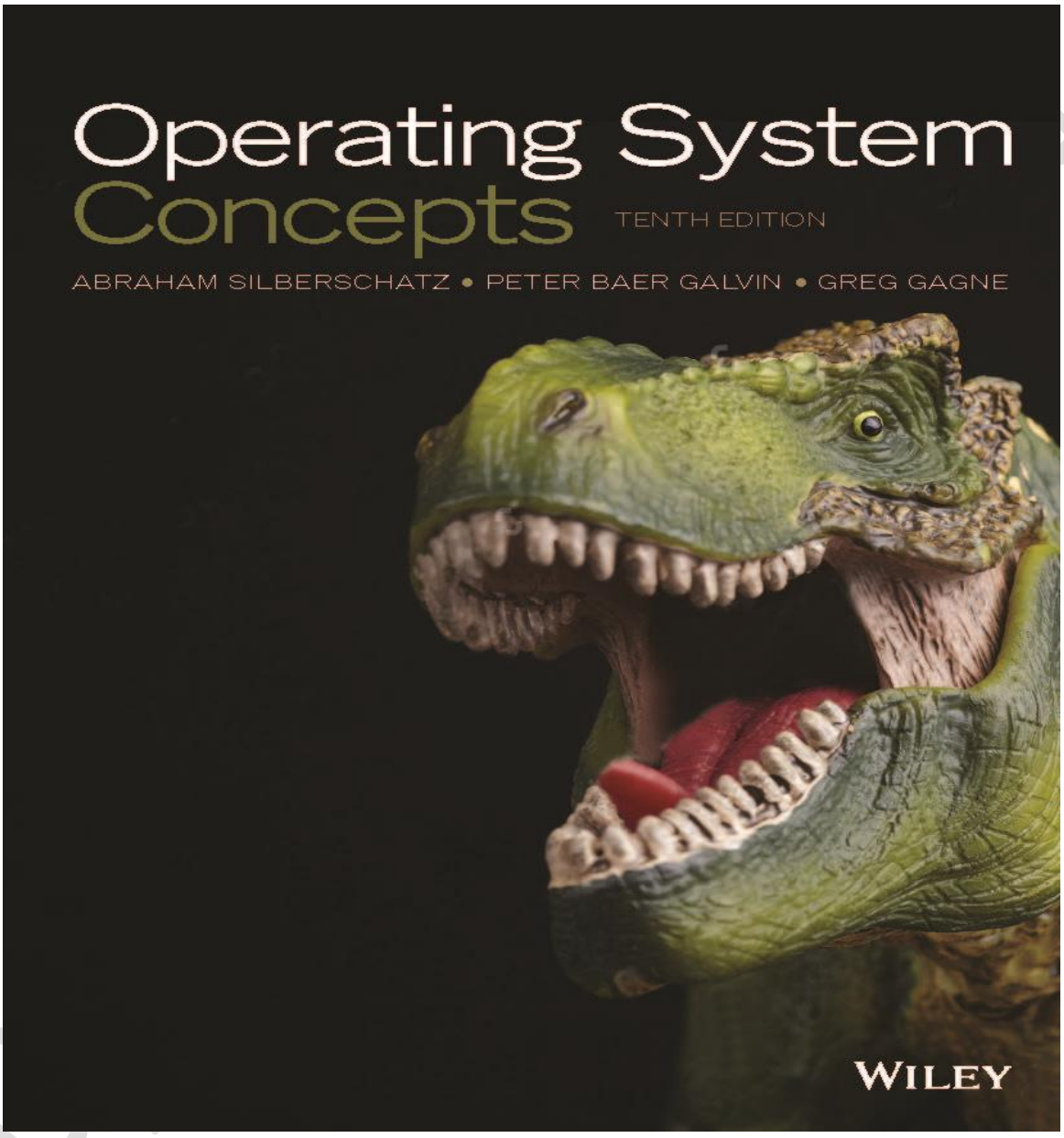
Mustansiriayah University /College of Education
Computer Science Department



Fourth Stage

Prepared By: Dr. Hesham Adnan ALABBASI

2019 – 2020

Chapter One

1. Intorduction

A modem computer system consists of one or more processors, some main memory, disks, printers, a keyboard, a display, network interfaces, and other input/output devices. All in all, a complex system. Writing programs that keep track of all these components and use them correctly, let alone optimally, is an extremely difficult job. For this reason, computers are equipped with a layer of software called the operating system, whose job is to manage all these devices and provide user programs with a simpler interface to the hardware.

What is an Operating System?

An operating system is a program that acts as an intermediary (وسيط) between a user of a computer and the computer hardware. It is an important part of almost every computer system. In simple terms, an operating system is the interface between the user and the machine. The purpose of an operating system is to provide the environment in which the user can execute programs. The O.S. is a resource allocator (managers all resources, decides between conflicting requests for efficient and fair resource use) and is a control program (controls execution of programs) to prevent errors and improper use of the computer.

1.2 Computer Systems Components

The computer system can be divided into four components:

1. Hardware: (CPU, memory, I/O devices): provides basic computing resources.
2. Operating system: Controls and coordinates use of hardware among various applications and users.
3. Application programs: define the ways in which the system resources are used to solve the computing problems of the users (Word processors, compilers, web browsers, database systems, video games
4. Users: (People, machines, other computers).

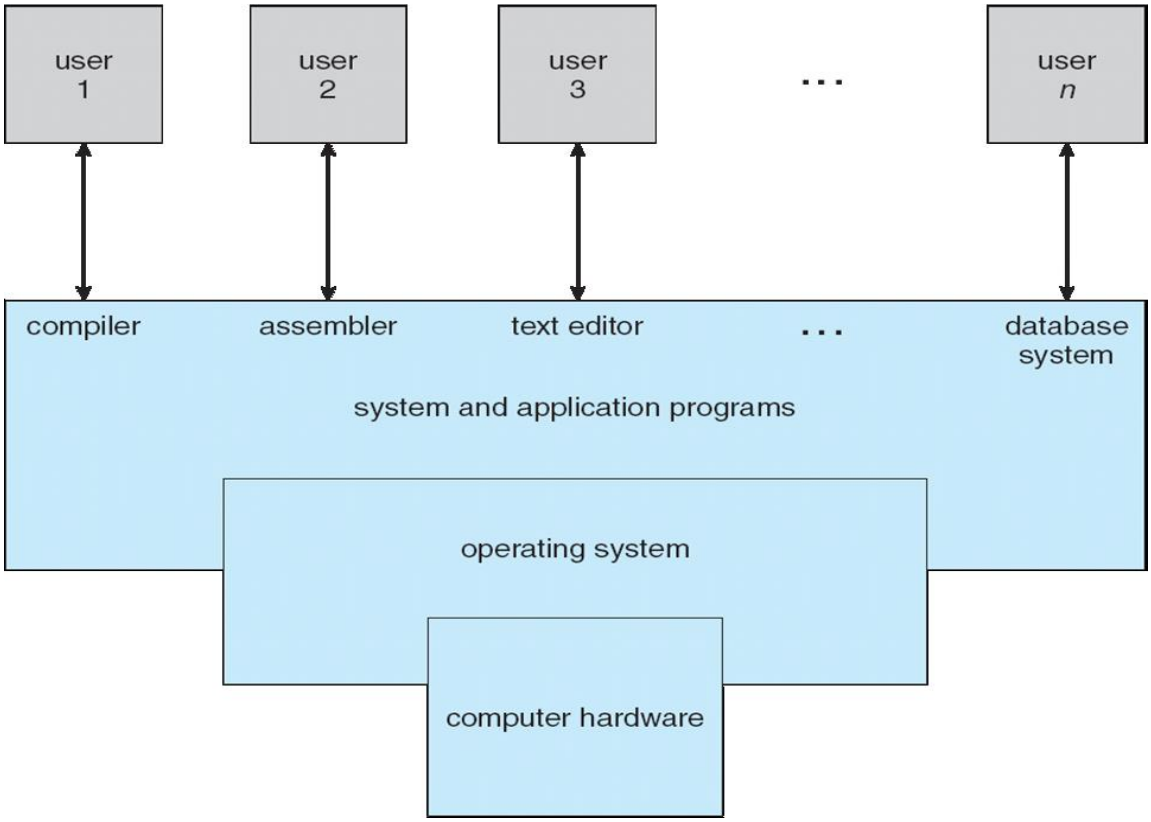


Figure 1.1 Abstract views of the components of a computer system.

1.3 The Operating system goals

1. The primary goal of an O.S. is to make the Computer System convenient to use. Operating systems exist because they are supposed to make it easier to compute with them than without them. This view is particularly clear when you look at operating systems for small personal computers.

2. A secondary goal is to use the computer H/W in an efficient manner. This goal is particularly important for large, shared multiuser systems. These systems are typically expensive, so it is desirable to make them as efficient as possible.

These two goals—convenience and efficiency-are sometimes contradictory. In the past, efficiency considerations were often more important than convenience. Thus, much of operating-system theory concentrates on optimal use of computing resources.

1.4 Th Operating System Functions

A more common definition, and the one that we usually follow is that the **operating system is the one program running at all times on the computer**—usually called the **kernel (Internal Commands)**.

O.S. performs many functions such as:

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

1.5 Operating System Categories

The main categories of modem O.S. may be classified into three groups, which are distinguished by the nature of interaction that takes place between the computer and the user:

1.5.1. Batch -System

In this type of O.S. Users submit jobs on a regular schedule (e.g. daily, weekly, monthly) to a central place where the user of such system did not interact directly with C/S. To speed up processing, jobs with similar needs were batched together and were run through the computer as a group. Thus, the programmers would leave their programs with the operator. The major

task of this type was to transfer control automatically from one job to the next. The O.S always resident in memory as in the figure 1-2. The output from each job would be send back to the appropriate programmer.

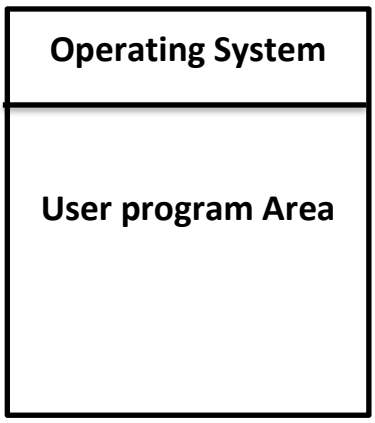


Figure 1-2 Memory layout for a simple batch system

Advantages:

batch system is very simple.

Disadvantages:

There is no direct interaction between the user and the job while that job is executing.

The delay between job submission and job completion (called turnaround time) may result from amount of computing time needed.

1.5.2. Time-sharing system:

This type of O.S. provides online communication between the user and the system, where the user will give instruction to the O.S. or to the program directly (usually from terminal) and receives an immediate response, therefore some time called an interactive system. The Time-Sharing system allows many users simultaneously share the computer system where little CPU time is needed for each user. As the system switches rapidly from one user to the next user is given the impression that they each have their own computer, while actually one C/S shared among the many users.

Advantage: Reduce the cup idle time.

Disadvantage: More Complex.

1.5.3. Real-Time system

A Real-Time system is used when there are rigid time requirements on the operation of a processor or the flow of data. A Real-time system guarantees that critical tasks complete on time. The secondary storage of any sort is usually limited; data instead being stored in short-term memory (Rom), (The Radar system is a good example for the real time system).

1.6. Performance Development:

O.S. attempted to schedule computational activities to ensure good performance, where many facilities had been added to O.S. some of these are:

1.6.1. On-Line and off-Line operations:

A special subroutine was written for each I/O device called a device-driver, and some peripherals (I/O devices) has been equipped for either On-Line operation, in which they are connected to the processor, or off-line operations in which they are run by control units not connected to the central computer system.

1.6.2. Buffering

A buffer is an area or primary storage for holding data during I/O transfers, where the I/O transfer speed depends on many factors related to I/O Hardware but normally unrelated to processor operation. On input the data placed in the buffer by an I/O channel when the transfer is complete the data may be accessed by the processor. There are two types of buffering:

1.6.2.1. The single-buffered

The channel deposits data in a buffer the processor will accessed that data the channel deposits the next data, etc. while the channel is depositing data, processing on that data may occur.

1.6.2.2. The Double-buffering

This system allows overlap of I/O operation with processing; while the channel is depositing data in one buffer the processor may be processing data in the other buffer. When the processor is finished processing data in one buffer it may process data in the second buffer. In buffering the CPU and I/O are both busy.

1.6.3. Spooling: (Simultaneous Peripheral Operation On-line)

Spooling uses the disk as a very large buffer for reading as far ahead as possible on input devices and for storing output files until the output devices are able to accept them. Spooling is now a standard feature of most O.S. Spooling allows the computation of one job can overlap with the I/O of another jobs, therefore spooling can keep both CPU and the I/O devices working as much higher rates. The figure below show the spooling layout.

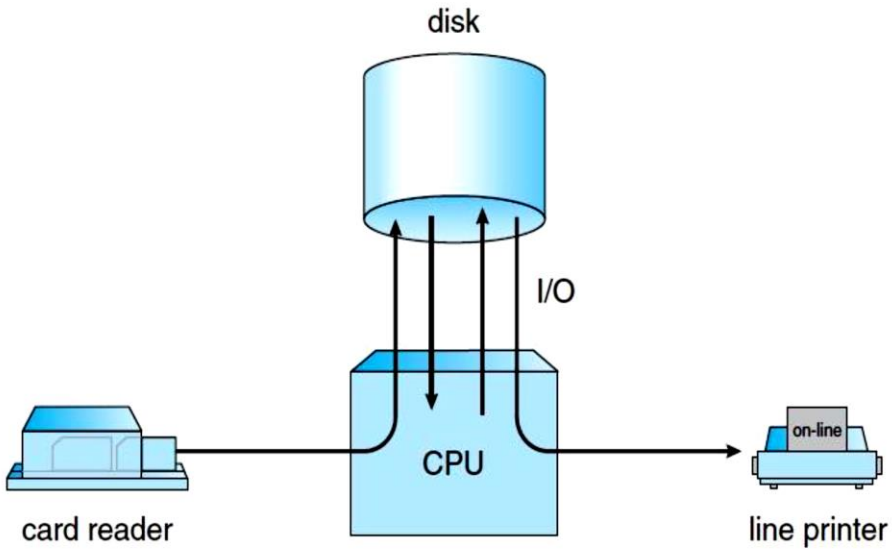


Figure 1.3 Spooling

1.7. Multiprogramming

Spooling provides an important data structure called a job pool kept on disk. The O.S. picks one job from the pool and begin to execute it. In multiprogramming system, when the job may have to wait for any reason such as an I/O request, the O.S. simply switches to and executes another job.

When the second job needs to wait the CPU is switches to another job and So on. Then the CPU will never be idle. The figure shows the multiprogramming layout where the O.S. Keeps Several jobs in memory at a time. This set of jobs is a subset of the jobs kept in the job pool.

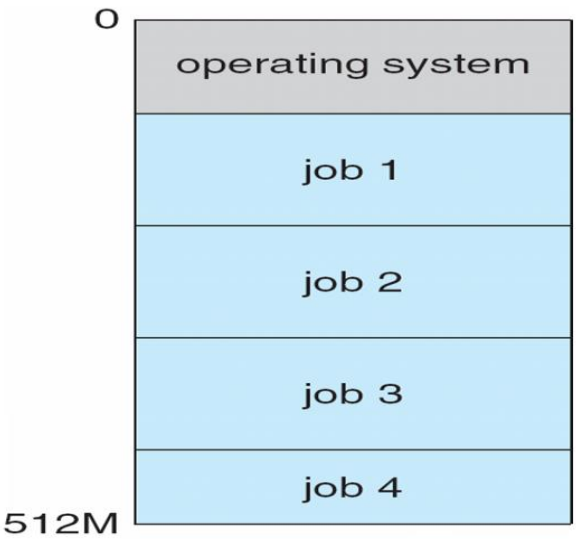


Figure 1-4 Memory layout for a multiprogramming system

1.8. Parallel Systems

Most systems today are a single-processor system that is they have one main CPU. There is a trend to have multiprocessor system, where such communication sharing the computer Bus, the clock, and sometimes memory and peripheral devices, as in the figure behind. The advantage of this type of systems to increase the throughput (the number of jobs completed in unit of time). Multiprocessors can also save money compared to multiple single systems because the processors can share peripherals, cabinets, and power supplies. Another reason for multiprocessor systems is that they increase reliability. The most common

multiple-processor systems now use the ***symmetric multiprocessing*** model, in which each processor runs an identical copy of the operating system, and these copies communicate with one another as needed. Some systems use ***asymmetric multiprocessing,*** in which each processor is assigned a specific task. A master processor controls the system; the other processors either look to the master for instruction or have predefined tasks. This scheme defines a master-slave relationship. The master processor schedules and allocates work to the slave processors.

1.9. Distributed systems

A resent tread in C/S is to distribute computation among several processors. In Contrast to the parallel system, the processors do not share memory and clock. The processors communicate with one another through various communication lines, such as high speed buses or telephone lines. This type of systems called a distributed system. There is a variety of reasons for building distributed systems, the major ones being these:

1. Resource sharing. If a number of different sites (with different capabilities) are connected to one another, then a user at one site may be able to use the resources available at another.

2. Computation speedup. If a particular computation can be partitioned into a number of sub computations that can run concurrently, then a distributed system may allow us to distribute the computation among the various sites to run that computation.

3. concurrently. In addition, if a particular site is currently overloaded with jobs, some of them may be moved to other, lightly loaded, sites. This movement of jobs is called load sharing.

4. Reliability. If one site fails in a distributed system, the remaining sites can potentially continue operating.