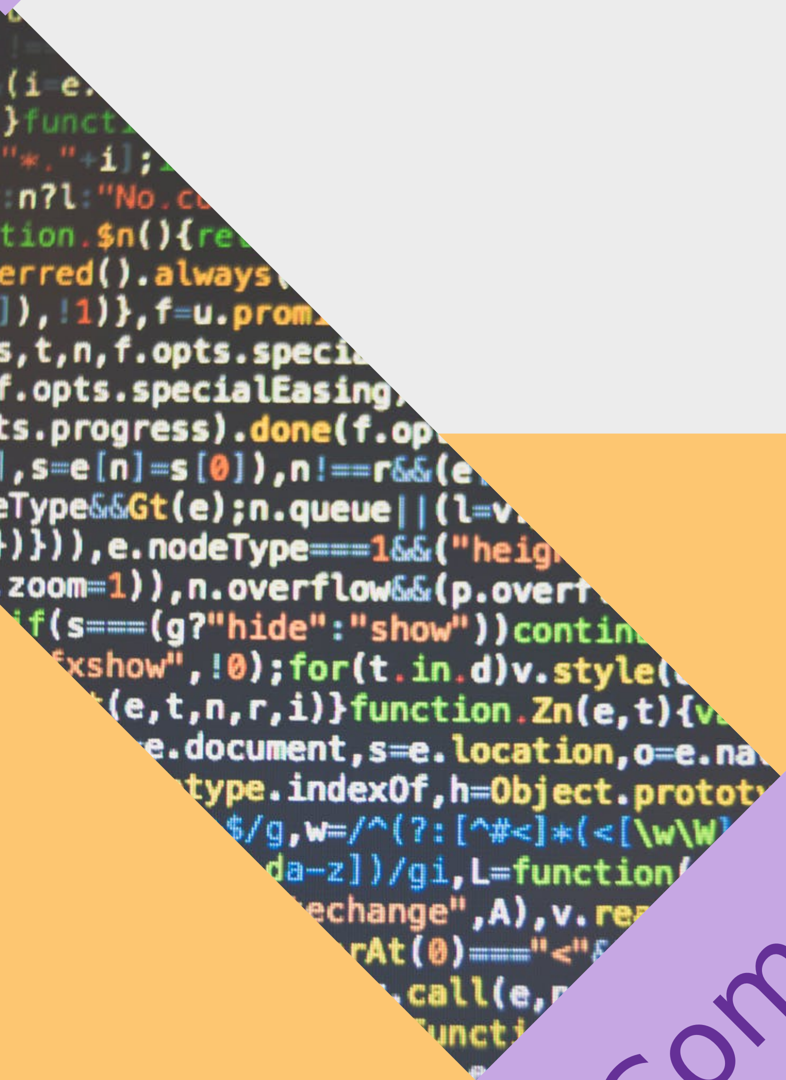
Curs 2020-2021



Computer Systems
Operating Systems

IES Jaume II El Just
Tavernes de la Valldigna

Contents

1 Introduction

This chapter focuses on Operating Systems. It explains the main functions of the operating system.

1.1 What is an Operating System

An Operating System is a program or set of programs with two main targets:

- On the one hand, the operating system must be able to offer the user a friendly environment in which the user will get comfortable using the computer.
- On the other hand, the operating system will communicate with the hardware properly for it to function in the best way possible.

  To sum up, an operating system is a program or set of programs whose main function is to offer a good interface between the hardware and the user. Essentially, whoever manages the computer.

1.2 Operating System Functions

The main functions of an operating system are:

- Process management: operating system manage processes in order to assign resources as CPU, hard disk, memory, etc... And priorities are established.
- Input /Output device management: the operating system coordinates and handles input/output devices.
- Memory management: operating system manage memory and it use.
- File system management: operating system manages how data is organized.

2 Historical changing of the operating system

Many elements of a computer's operating system are known in the computer science generation. However, we must first consider the history of its development. It started to emerge in the forties and continues in our present day. Throughout these years, each generation starts with a technological invention such as: vacuum valves, transistors, integrated circuits, etc. As regards to an operating system, we can find a historical evolution as well.

2.1 Zero generation

In this generation (in the forties), operating systems didn't exist, and every instruction had to be coded by hand.

2.2 First generation

In the fifties, the first operating systems appeared. Whenever some task needed to be done, this task would take control of the computer until it finished. Then the control of the computer was taken by the operating system again. Later punched cards were invented in order to give some different programs to the computers. At the end of this generation, the transistor was invented.

2.3 Second generation

This second generation was born in the sixties. Multitasking and multiprogramming appeared. Consisting of:

- Multiprogramming: several processes are remaining in main memory. The central processing unit changes each program in order to provide a feeling of executing several things at the same time, and to get closer to multitasking.
- Multitasking: several central processing units working together in order to execute several programs at the same time.

2.4 Third generation

IBM introduces computers for general use. Not only they were huge, but also expensive. The third generations were from the mid-sixties to mid-seventies.

2.5 Fourth generation

From the mid-seventies until present day. A new approach was taken because of the new concept of computing. Networks and the new way of working with computers that were geographically remote facilitated operating systems that show more complexity with security systems, user policy, codification of data, etc.

3 Operating System classification

According to the perspective taken, we can carry out different classifications of operating systems:

- According to the number of users:
  - **Mono user**: it is only possible for one user to work at the same time
  - **Multiusers**: it is possible for more than one user to work at the same time
- According to intern structure: in this classification it is taking into account the means of designing the operating systems:
  - **Monolithic**: this was used for the first operating system. Kernel carries out every function.
  - **By Layers**: It was necessary to divide operating system by layers because technology grew and complexity with it.
  - **Virtual machine**: each virtual machine executes an operating system on a hardware replica of the real computer.
- According to the processors:
  - **Mono process**: this kind of operating system, changes each program in order to provide a feeling of executing several things at the same time, and to get closer to achieve multitasking. However, the CPU is only able to execute just one process.
  - **Multiprocessor**: it is possible to carry out several processes simultaneously, that is, at the same time. We can find two different kinds of multiprocessors:

  * **Symmetric**: processes are distributed through all processors.
  * **Asymmetric**: one or several processors are assigned with high priority task and one or even several processors are assigned with low level priority tasks.

- According to the task: we have both kinds in this classification:
  - **Monotask**: one task per user at the same time. It is possible for a multiuser to exist in a monotask operating system. This operating system would be able to accept several users, but one task per user at the same time.
  - **Multitask**: The operating system which allows several tasks to be carried out at the same time.

Try it 1
Investigate about the following operating system. Point out what kind of operating system it is according to the number of users, number of tasks and number of processors:

OS	According to the number of users	According to the number of task	According to the number of processors
MS-DOS			
Windows 98, ME			
Windows XP/ vista /7 /8 /10			
Unix, Linux, Windows server			

Try it 2
Could you say what the most popular operating systems are nowadays?

4 Components of an operating system

The main components of an operating system are:

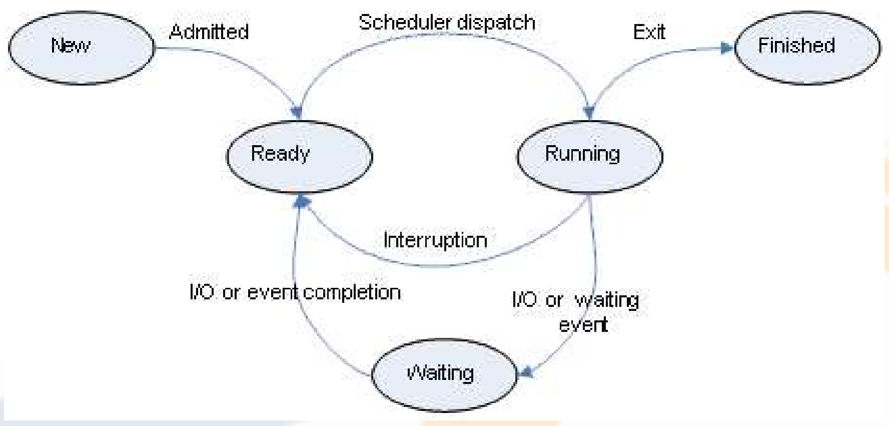
- Kernel: this is the core of the operating system and is loaded in the main memory when the computer turns on. The kernel remains in the memory executing the following tasks:
  - Memory managing
  - Communications between processes
  - Interruptions managing
  - Error managing
  - Input /output control
- Processes: a process is a program in execution. A program is converted to a process when it is loaded in memory. When a program is using the CPU, it is a process as well.

5 Process Management

As said before, a process is any program when it is using the CPU, or when it is loaded in main memory. In order to control and manage these processes, it is necessary to design some algorithms. This is a necessary feature of the multitasking operating system. First, we are going to see the states which can adopt the processes.

5.1 Process states

Along the life of the process, it is possible that it goes through several states. The picture shows us the life of a process in an operating system:



- **New**: in the process management, there is a program called long–term scheduler. This program is in charge of choosing whether the process can be admitted or not. If it is admitted, it will go to the next state. We refer to this as the ready.
- **Ready**: process is ready in order to use the CPU when the short-term scheduler chooses it. During that time, it should wait in the line of the ready processes.
- **Running**: CPU is running the process. Furthermore, the process has all the resources available. Here, the process can take different ways:
  - **Ready**: process will go back to this state when the short-term scheduler considers that the process has to leave the CPU and the process hasn't finished its task. In this case, the dispatcher will have to do a context change. The context change is to keep all the information of the process. When the process goes again into the CPU, this information will be loaded in order to carry on with the process.
  - **Finished**: the process does not need the CPU because it has finished its task.
  - **Waiting**: the process in which this state takes depends on other events, some input/output task, or event taking place in with the process. In this case, the change of context should be done.
- **Waiting**: as it has been said before, the process is waiting for some event or input /output event. For instance, data which should be read from a hard disk. When the event is completed, the process will go to the ready state.
- **Finished**: the process finished its task.

5.2 Short term scheduler algorithms

In order to carry out its task, the short-term scheduler follows some algorithms. These algorithms are to decide when and which process has to come in or come out of the CPU. The algorithms have been studied in order to provide equity, high performance, low time of response, efficiency, and reduce waiting time.

5.2.1 First in First Out

In this case, the first process to arrive is the queue; the first process will be served. The process will leave the CPU when it finishes its task. Therefore, this kind of algorithm does not use ejection.

5.2.2 Shortest Job First

The short-term scheduler chooses those tasks which will take less time to finish. In case of two tasks that have the same time the first to arrive will be chosen. The problem here is to predict the time that each task is going to take. Besides, it can produce starvation with the longest tasks. This algorithm uses ejection.

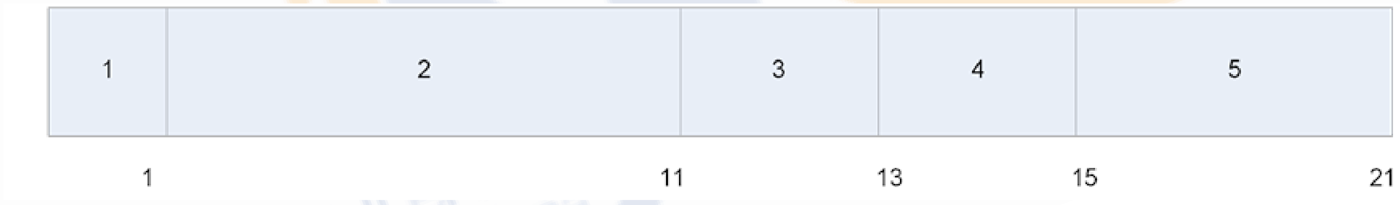
5.2.3 Round Robin

Processes use CPU for a short time. This short-term is called Quantum. The processes in the queue are sorted in using the FIFO (First in First Out) algorithm. When the process is chosen, that process uses CPU for a short time; or the time left in order to finish if that time is lower than the quantum.

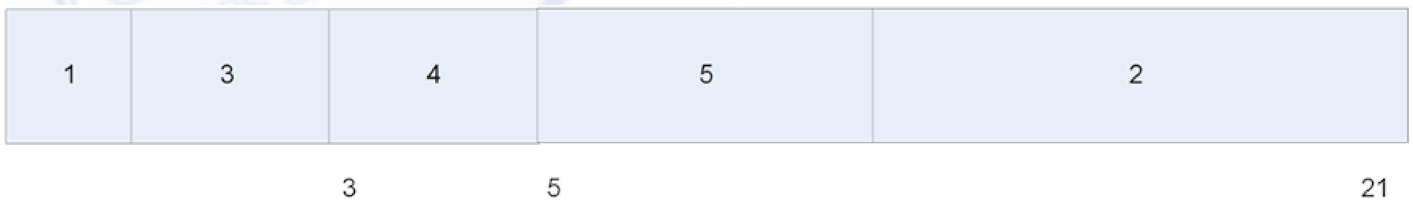
Next example shows how each algorithm would be executed:

Task	Time of use CPU
1	1
2	10
3	2
4	2
5	6

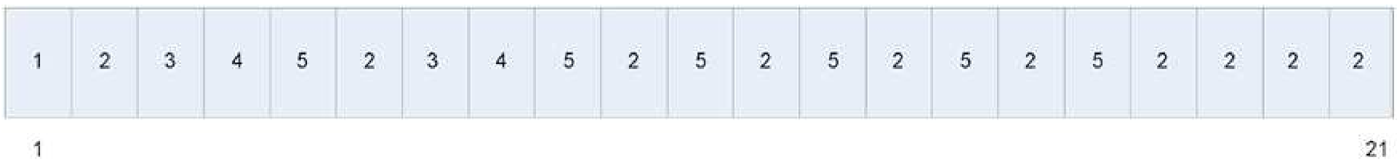
First In First Out



Shortest Job First



Round Robin



Try it 3

Calculate the return time and the average time for each one of the processes in each algorithm:

The return time is from the time the task is ready until the tasks are finished.

Task	FCFS	SJF	RR
1	1		
2	11		
3	13		
4	15		
5	21		
Average time	12.2		

Try it 4

Calculate the efficiency of each process in each algorithm.

The efficiency is the quotient between total time of the process and the time spent in finishing the task as a percentage.

Task	FCFS	SJF	RR
1	100 %		
2	90.90%		
3	15.38%		
4	13.33%		
5	28.57%		
Average efficiency	49.63%		

Try it 5

Calculate the waiting time for each process

Waiting time is the time that the process is waiting for some resource. It is the subtraction between return time and running time.

Task	FCFS	SJF	RR
1	0		
2	1		
3	11		
4	13		
5	15		
Average time	8		

Try it 6

Break down the results and determine what you can conclude from the outcome. Which algorithm shows us the best results?

6 Memory management

Memory management is used only in multitask systems. When we talk about memory management, we talk about main memory (RAM memory) management. It is well known that each program is loaded in main memory when they are running. These programs must share space with an operating system which is already loaded on the memory. Memory management tries to solve problems like:

- Loading programs and where it should be placed
- More than one program on memory
- Sharing memory areas by more than one program.
- Running programs bigger than placement in memory
- The efficient management of space of memory.

6.1 Virtual Memory

This technique is used to run programs which don't fit in main memory. In this case, the program is separated in blocks. Each block is called pages. Only a part of these pages is loaded in main memory; the rest are in a secondary storage. In Linux it is called swapping. So, thanks to the virtual memory, computers are able to run programs greater than its main memory. Some benefits of the virtual memory are:

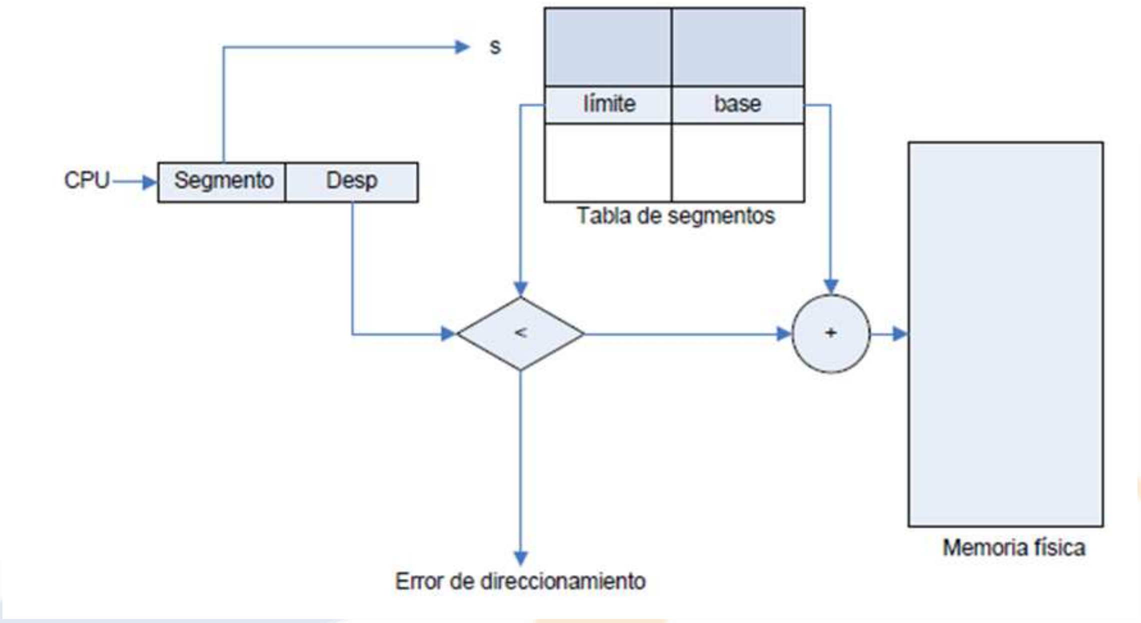
- It allows running programs that are greater than the size of the main memory.
- It allows more programs to be loaded in main memory
- It enables programs to run faster
- It reduces the exchange frequency because CPU is able to run a program while another is exchanging pages with the virtual memory.

It is possible that running processes, some gaps in memory are created because there are some programs which are greater than another. And likewise, can be opposite. Avoiding this technique is used. The memory addresses are formed by two numbers, the first number points out the initial memory of the process, and the second number the offset; where it is allocated in main memory. For instance, if we have (54, 9) the final memory address is 63.

6.2 Segmentation memory

A similar technique is used in segmentation memory process where memory is divided in variable block sizes. Each block is able to change its size depending on the necessity of the program.

Each segment is formed by a name and a size. Any address is formed by a number which specifies the number of the segment and the offset inside of this segment. The address is checked out and eventually the address is formed in order to access the main memory.



7 I/O management

This management is in charge of peripheral operations in a computer:

- At the beginning each process had to verify if there was an I/O process.
- Later, interrupts are used. When an I/O operation is required, CPU is taken by these operations. This is called interrupts because CPU interrupts its normal working in order to attend the I/O process.
- DMA (Direct Memory Access): this technique is carried out for a device called DMA. CPU can carry on with its process and when DMA finishes its I/O process, an interrupt is sent to the CPU in order to warn it.

Get faster

In order to get faster in writing, hard disk cache is used. The data is written in cache before. Hard Disk will write the data in the storage in order for CPU to carry on with its process.

8 File System management

The information is organized in hard disk in order to be available at any time to be checked out.

- File: they are objects in order to save data
- Directories: they are containers in order to keep the information organized.

The way of organizing the information and their rules depend on each operating system. For instance, the rules for the name and the difference between upper case and lower case. And most of the operating systems use extension. This is very useful in order to know what kind of file it is.

The attributes like partition size, maximum file size and the cluster size are very important. The hard disk is divided in little portions in order to save the information. All the storage on the hard disk is divided with the same cluster size. When we save some file in hard disk, and if its size is lower than the size of the cluster, one portion (one cluster) will be needed. However, if your file size is greater than the cluster size the file will need more than one cluster. The portions of the file will be saved where the operating system can. If these portions are so far in the hard disk, more time will be required in order to load the file. In order to minimize the distance between portions you can defrag the hard disk.

Besides, each operating system has a different way in order to organize the files. For instance, in windows as well-known FAT-16, FAT-32 and NTFS and Linux use ext4.

Try it 7

What is a partition?

____

____

____

Find information of each system file about:

	FAT 16	FAT 32	NTFS	Ext 4
Max. File Size				
Max Partition Size				