

# **UNIDAD 2**

# **INSTALACIÓN DE**

# **SISTEMAS OPERATIVOS**

**Módulo Sistemas Informáticos**  
**Curso 2020-2021**

# Contenido

2

1. Introducción: Qué es un sistema operativo
2. Estructuras y arquitecturas de sistemas operativos. Evolución y tipos
3. Funciones del sistema operativo

# 1. Introducción.

## Qué es un sistema operativo

3

- ***Un Sistema Operativo (SO) es un conjunto de programas y funciones que gestionan y coordinan el funcionamiento del hardware y del software, ofreciendo al usuario una forma sencilla de comunicarse con el ordenador.***
- El SO es el **software básico** del ordenador.
- Gestiona todos los recursos del sistema y proporciona la base para la creación y ejecución del software de aplicación.
- Es el intermediario entre el hardware y el software diseñado para ese SO.

# Introducción

4

- El SO es el primero que se carga en la memoria del ordenador al ser puesto en funcionamiento, y a continuación todos los programas que se precisen.
- Previo a la carga del SO se realiza el *POST (Power On Self Test)*, que se encarga de reconocer y testear el correcto funcionamiento del procesador, memoria, unidades de disco y otros dispositivos.



## 2. Estructuras y arquitecturas de sistemas operativos. Evolución

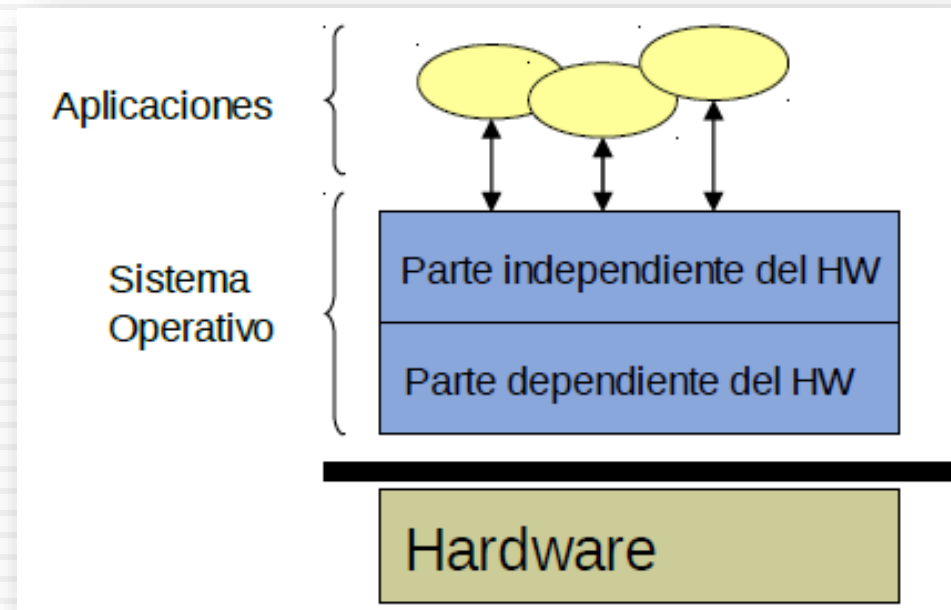
5

- Un SO se divide en varias **capas**.
- Cada una se comunica con las capas inmediatamente inferior y superior:
  - ▣ **Gestión del procesador** – Indica las instrucciones que debe llevar a cabo la CPU.
  - ▣ **Gestión de la memoria** – Asignación y liberación de memoria a los procesos, y control de violación de acceso.
  - ▣ **Gestión de procesos** – Planificador de procesos (**scheduler**) e intercambios de estado (**dispatcher**).
  - ▣ **Gestión de E/S** – Sincroniza la comunicación con el exterior.
  - ▣ **Gestión de la información** – Manejo del sistema de archivos (creación, permisos de acceso,...).

# Estructuras y arquitecturas de sistemas operativos. Evolución

6

- Las 2 primeras capas son dependientes del **Hardware**.



# Estructuras y arquitecturas de sistemas operativos. Evolución

7

- La **evolución** de los SO ha sido consecuencia de la evolución del hardware.
- De forma genérica podemos hablar de varias generaciones de SO:
  1. **Primera generación**
    - Entre 1940 y finales de los 50.
    - Tecnología: Válvulas de vacío
    - Los ordenadores no tenían sistemas operativos.
    - Para programarlos había que modificar unos interruptores mecánicos.
    - Con la construcción del Univac I (*Universal Automatic Computer*) aparece el concepto de sistema operativo.
    - Este ordenador funcionaba con el sistema operativo EXEC I, que posteriormente también se utilizó en el Univac II.

## 2. Segunda generación

- Desde 1960 a 1965.
- Tecnología: Transistores
- Procesamiento por lotes o batch.
- Esta tarea la realizan los primeros sistemas operativos: **Monitores**.
- La información se introducía en el ordenador mediante tarjetas perforadas o unidades de cinta magnética.
- Nueva versión del Univac: 1107, llamado también Univac 7
- En 1961 se desarrolla el sistema operativo CTSS, uno de los primeros que usó el tiempo compartido.
- El sistema operativo EXEC 8 se desarrolló en 1964 para el Univac 1108 que también usó el tiempo compartido



## 3. Tercera generación

- Desde 1966 a 1971.
- Tecnología: Circuitos integrados.
- Macro ordenadores o mainframes de IBM. Sistemas operativos: OS/360, CP/CMS y el VM/CMS
- Otra empresa: DEC desarrolla una serie de ordenadores. El sistema operativo que utilizaban era el TOPS-10.
- A mediados de los 60 se desarrolla el sistema operativo MULTICS incluyendo muchas novedades. Sistema monolítico.

# Estructuras y arquitecturas de sistemas operativos. Evolución

10

## 4. Cuarta generación

- Desde 1971 a 1981.
- Tecnología: Procesador.
- Los sistemas operativos son multiusuario y multitarea y la interfaz de usuario sigue siendo en modo texto y se siguen usando muchos de los sistemas operativos de la generación anterior (algunos sustituidos por sus versiones nuevas)
- UNIX, desarrollado en C por la empresa AT&T Bell, surgió a principios de los 70.  
Es un sistema operativo multiusuario y multitarea. Desde su inicio se diseñó con características especiales para las comunicaciones, compartir archivos, uso remoto de equipos... lo que ha contribuido a su gran éxito.
- Surge otro sistema operativo muy extendido que llegó a convertirse en un verdadero estándar para muchos fabricantes por su facilidad de manejo y su fiabilidad, el CP/M.

## 5. Quinta generación

- Comenzó en 1981... hasta la actualidad
- Tecnología: PC de IBM
- Llevaba inicialmente el sistema operativo PC-DOS, que se basó el en CP/M.
- Aparece MS-DOS de Microsoft, interfaz texto
- Competidor de MS-DOS: DR-DOS -> Novell DOS
- Aparecen diferentes versiones de Windows: 1.0 , 2.0, 2.1, 3.0, 3.1 y 3.11 ,se instalaban sobre MS-DOS.
- A partir de Windows 95 se instalan ya independientes de MS-DOS
- Windows 98, Windos Me, Windows XP, Vista, Windows 7 y 8.
- Sistemas operativos de red: Novell NetWare, le ganan terreno los sistemas operativos en red de Microsoft como Windows NT, Windows 2000, Windows 2003 Server, Windows 2008 Server y el más nuevo Windows 2012 Server.

# Estructuras y arquitecturas de sistemas operativos. Evolución

12

## □ Quinta generación...

- ▣ IBM lanza el sistema operativo OS 400 para los ordenadores AS 400.
- ▣ A principios de los 90 surge **Linux**, que es una versión del sistema operativo UNIX para PC, siendo, junto Windows, uno de los sistemas operativo más extendidos.
- ▣ En 1992 la empresa Sun Microsystem, actualmente adquirida por Oracle, lanza al mercado el sistema operativo **Solaris**, que es una versión de Unix y que ha liberado parte del código.
- ▣ Los ordenadores de la empresa Apple, los Macintosh o Mac, funcionan bajo el sistema operativo Mac OS en sus distintas versiones.
- ▣ Actualidad: Sistemas operativos de móviles, tablets, etc...: Android, IOS, Windows RT, etc....

# Estructuras y arquitecturas de sistemas operativos. Tipos

13

## □ Clasificación de sistemas operativos

### ▣ *Atendiendo a diferentes criterios:*

1. Tiempo de respuesta
2. Número de usuarios
3. Número de procesos
4. Número de procesadores
5. Trabajo en red

# Tipos de sistemas operativos:

## Según el **tiempo de respuesta**

14

- *Tiempo que transcurre desde que un proceso de un usuario llega al sistema hasta que el usuario obtenga una respuesta o unos resultados.*

### 1. **Procesamiento por lotes**

- Los procesos se ejecutan secuencialmente uno tras otro.
- Mientras un proceso no finalice su ejecución no se podrá comenzar a ejecutar el proceso siguiente.
- El usuario no obtiene los resultados de forma inmediata y no existe interacción con éste.
- Tiempos de respuesta son altos por lo que actualmente no son utilizados.

# Tipos de sistemas operativos:

## Según el **tiempo de respuesta**

15

### 2. **Interactivos**

- Los procesos se ejecutan y pueden pedir información al usuario a través de la pantalla y recibirla por teclado.
- Habitualmente tiempos de respuesta menores que en el procesamiento por lotes.
- Utiliza la técnica del tiempo compartido (***time sharing***) en la que el procesador divide su tiempo entre todos los procesos de manera que parece que cada proceso es el único que se ejecuta en ese momento en el sistema.

# Tipos de sistemas operativos:

## Según el **tiempo de respuesta**

16

### 3. **Tiempo real**

- Los procesos requieren un tiempo de respuesta muy bajo, es decir, inmediato.
- Se utilizan en campos donde un tiempo de respuesta alto implicaría graves consecuencias como por ejemplo en tráfico aéreo, procesos industriales, sistemas espaciales...
- El tiempo de respuesta es crucial y repercute en grandes pérdida económicas o de vidas humanas.



# Tipos de sistemas operativos:

## Según el **número de usuarios**

17

- *Según el número de usuarios que pueden utilizar simultáneamente el sistema informático, podemos dividir los sistemas operativos en:*

### **1. Monousuario**

Sólo un usuario, para el que están disponibles todos los recursos tanto hardware como software, puede utilizar el sistema informático.

### **2. Multiusuario**

Varios usuarios pueden utilizar el sistema informático *simultáneamente*, lo que conlleva que se tengan que repartir recursos entre ellos.

# Tipos de sistemas operativos:

## Según el número de procesos

18

- *Hace referencia al número de procesos, es decir, programas que se pueden estar ejecutando de manera simultánea.*
- Atendiendo a este criterio:
  1. **Monoprogramación (monotarea)**
  2. **Multiprogramación (multitarea)**

# Tipos de sistemas operativos:

## Según el número de procesos

19

### 1. **Monoprogramación (monotarea)**

- ▣ El sistema operativo sólo es capaz de gestionar la ejecución de un proceso a la vez.
- ▣ Hasta que no termine la ejecución de ese proceso no podrá empezar la ejecución del siguiente.
- ▣ Los procesos se ejecutan de forma secuencial.

# Tipos de sistemas operativos:

## Según el número de procesos

20

### 2. **Multiprogramación (multitarea)**

- ▣ El SO es capaz de gestionar varios procesos ejecutándose a la vez, de forma simultánea en la CPU.
- ▣ La multitarea real (en sentido estricto) sólo se produce en sistemas que cuentan con más de un procesador (multiproceso), ya que, en un mismo instante, un procesador sólo puede estar ocupado por un único proceso.

# Tipos de sistemas operativos:

## Según el **número de procesadores**

21

- *Hace referencia al número de procesadores que tenga instalado el sistema informático.*
- Aunque el equipo tenga instalado más de un procesador es necesario que tenga instalado un sistema operativo que los pueda utilizar para aprovechar esa ventaja.
- Por ejemplo, si sobre un superordenador instalamos un sistema operativo monotarea sólo se ejecutaría un proceso a la vez.

# Tipos de sistemas operativos:

## Según el número de procesadores

22

### □ *Atendiendo a este criterio:*

#### 1. **Monoproceso**

Sólo pueden trabajar con un único procesador, aunque existan más en el sistema. Ej. MS-DOS, Windows 98.

#### 2. **Multiproceso**

Pueden gestionar varios procesadores. Si un sistema operativo multiproceso está montado sobre un sistema que tiene un solo procesador, aunque el sistema operativo pueda gestionar más de un procesador, se comportaría como un sistema operativo monoproceso.

Si el sistema informático tiene más de un procesador, un sistema operativo multiproceso puede utilizar varios procesadores de simultáneamente, por lo tanto, ejecutaría varios procesos de forma simultánea.

# Tipos de sistemas operativos:

## Según el número de procesadores

23

### **Multiproceso**

*Pueden ser de dos tipos:*

- **Simétricos:** el sistema operativo distribuye los procesos entre los distintos procesadores por igual.  
Hace un reparto equitativo de la carga de procesamiento.
- **Asimétricos:** se distribuyen los procesadores de forma que a cada uno se le asigna un trabajo diferente.  
La carga de cada procesador no tiene por qué ser igual a la de los otros.

# Tipos de sistemas operativos:

## Según el **trabajo en red**

24

- Dependiendo de la forma en que el sistema operativo puede trabajar y utilizar los recursos de la red podemos encontrar los siguientes tipos de sistemas operativos:

### 1. **Centralizados**

Con este tipo de modelo los ordenadores *mainframe* se encargaban de todo el procesamiento y los usuarios manejaban únicamente terminales *tontos* (no disponían de memoria ni procesador).

Actualmente se siguen utilizando los sistemas centralizados (como los *Terminal Services* de Microsoft) pero los terminales dejan de ser *tontos*.



# Tipos de sistemas operativos:

## Según el trabajo en red

25

### 2. En red

Estos sistemas operativos mantienen a dos o más ordenadores unidos a través de algún medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información el sistema.

En este entorno, cada ordenador mantiene su propio sistema operativo y tiene su propio usuario (o grupo de usuarios).

# Tipos de sistemas operativos:

## Según el trabajo en red

26

### 3. Distribuidos

En un sistema distribuido los recursos que utiliza el usuario pueden estar “distribuidos” por diferentes equipos de una red, pero todo de forma transparente al usuario, que tendrá la sensación de trabajar con recursos locales a la propia máquina.

Así, los ficheros que utiliza un usuario, o el procesador que ejecuta sus procesos pueden estar situados en cualquier ordenador de la red sin que el usuario sea consciente de ello.

### 3. Funciones del sistema operativo

27

□ *Las funciones principales del SO son:*

1. Control de ejecución de programas.
2. Administración de memoria.
3. Administración de periféricos (E/S).
4. Gestión de la información.
5. Otras funciones: Control de concurrencia, gestión de usuarios, control de errores, etc.

## Funciones del sistema operativo:

### 1. Control de ejecución de programas (procesos)

28

- Un **proceso** es un programa en ejecución (o preparado para su ejecución) situado en memoria.
- Cada vez que se manda ejecutar un programa se crea un proceso.
- El sistema operativo debe realizar una gestión adecuada de los recursos del sistema para la correcta ejecución de los procesos.
- Cada proceso tendrá una estructura de datos llamada **PCB** o **Bloque de Control de Proceso**.

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

29

- Dependiendo del sistema operativo, el **BCP** puede contener la siguiente información:
  1. **Estado** actual del proceso (en **ejecución**, **preparado** o **bloqueado**).
  2. **Identificación** del proceso. Cada proceso tiene asignado un PID asignado por el SO.
  3. **Prioridad** del proceso:
    - Prioridades **estáticas** – No se modifican durante la ejecución del proceso.
    - Prioridades **dinámicas** – Pueden ser modificadas:
      - Por los usuarios que tengan privilegios suficientes.
      - Por el SO - Si un proceso tiene una prioridad muy baja puede darse el caso de que nunca se ejecute (**inanición**). Para evitar eso se aumenta la prioridad del proceso a medida que se incrementa el tiempo de espera.

*Continúa...*

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

30

- 4. **Zona de memoria asignada:** cada proceso debe tener una zona de memoria independiente que no puede ser interferida por otros procesos.
  - 5. **Recursos asociados al proceso:** Cada proceso debe de tener a su disposición determinados recursos hardware y algunos software.
- 
- Como puede haber varios procesos ejecutándose concurrentemente tenemos una **Tabla de Procesos** que contiene una entrada por cada proceso.

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

31

- Dependiendo de si el proceso lo ha lanzado un usuario o es un proceso del sistema operativo, podemos distinguir:
  1. **Procesos del sistema:** generalmente se ejecutan al iniciar el sistema y suelen estar en ejecución para proporcionar servicios a los usuarios.
  2. **Procesos de usuarios:** los manda ejecutar el usuario. Si el sistema es multiusuario puede haber varios procesos de distintos usuarios ejecutándose a la vez.

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

32

### □ Planificación de CPU-Multiprocesamiento

- Existen ordenadores que disponen de varios procesadores por lo que son capaces de ejecutar al mismo tiempo varias tareas si disponen de sistemas operativos multiproceso.
- En los ordenadores convencionales, normalmente, sólo se dispone de un procesador por lo que en cada momento sólo se estará procesando una tarea (*actualmente, los procesadores disponen de varios núcleos que, en determinadas actuaciones pueden actuar como si fueran varios procesadores*)
- No obstante, es posible crear la ilusión de multiprocesamiento por medio de interrupciones, cambiando el proceso activo cada cierto tiempo aunque no se haya finalizado la tarea en ejecución.



# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

33

El sistema operativo se encarga de compartir la CPU entre los distintos procesos de la manera más eficiente posible, persiguiendo los siguientes **objetivos**:

- ▣ *Equidad*
- ▣ *Eficiencia*
- ▣ *Bajo tiempo de respuesta*
- ▣ *Rendimiento alto*
- ▣ *Minimización del tiempo de respuesta*

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

34

Los estados en los que se puede encontrar un proceso son:

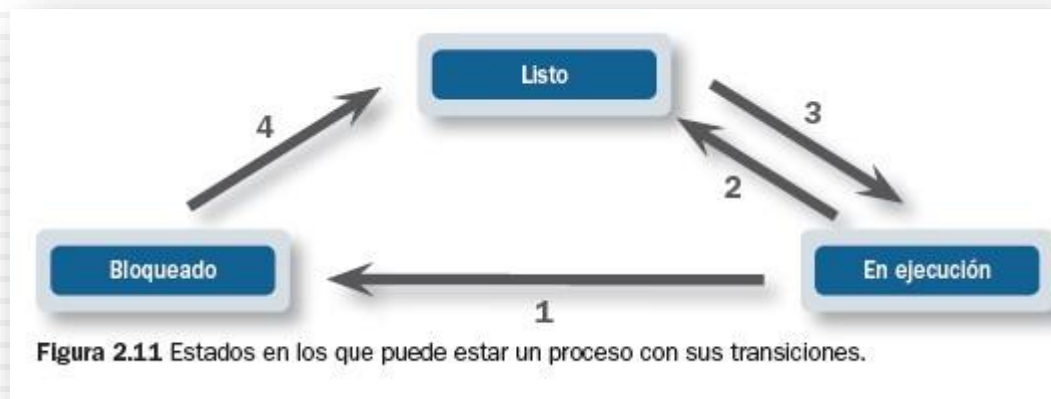
- ▣ **Listo (en espera o preparado):** el proceso está preparado para ejecutarse, es decir, está en espera de que el proceso que se está ejecutando deje libre la CPU.
- ▣ **Bloqueado:** el proceso está esperando un recurso que está siendo utilizado por otro proceso en ese momento.
- ▣ **En ejecución:** el proceso está ejecutando sus instrucciones en ese momento, es decir, está ocupando la CPU.

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

35

### □ Cambios de estado



1. **De ejecución a bloqueado:** ocurre cuando el programa que está en ejecución necesita algún elemento, señal, datos, etc., para continuar ejecutándose.
2. **De ejecución a listo:** ocurre cuando un proceso ha utilizado el tiempo asignado por la CPU para su ejecución y tiene que dejar paso al siguiente proceso.
3. **De listo a ejecución:** ocurre cuando al proceso que está preparado para ejecutarse se le asigna la CPU para que continúe (o comience) con su ejecución.
4. **De bloqueado a listo:** ocurre cuando el proceso ya puede disponer del recurso por el que se había bloqueado.

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

36

### □ Algoritmos de planificación del procesador

- **Planificación a Corto Plazo – dispatcher:** decide qué proceso de los que están preparados para ejecutarse pasa a utilizar el procesador.
- Entre las distintas planificaciones existen dos tipos principales:
  - **Apropiativa:** una vez que se le ha otorgado la CPU a un proceso, le puede ser retirada.
  - **No apropiativa:** Una vez que se le ha otorgado la CPU a un proceso, no le puede ser retirada.

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

37

Todos los algoritmos deben **maximizar la utilización de la CPU** evitando que esté libre en algún momento y **minimizar el tiempo de respuesta** de cada proceso.

**En cada algoritmo sabremos para cada proceso:**

- **Tiempo de entrada o de llegada al sistema ( $T_0$ ):** es el momento en que el proceso entra en el sistema.
- **Tiempo de ejecución ( $T_x$ ):** tiempo que el proceso necesita para su ejecución total.

**Para cada proceso nos interesa obtener:**

- **Tiempo de respuesta o tiempo de retorno ( $T_R$ ):** tiempo que pasa desde que el proceso llega al sistema hasta que se obtienen los resultados.
- **Tiempo de espera ( $T_E$ ):** tiempo que el proceso está esperando en el sistema.

$$T_E = T_R - T_x$$

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

38

□ **Los algoritmos de planificación más importantes son:**

**1. FCFS**

*(First Come First Served o  
FIFO – First In First Out)*

**2. SJF (Shortest Job First)**

**3. SRTF (Shortest Remaining Time First)**

**4. Round-Robin (por rondas)**

**5. Algoritmos con prioridad**

# Funciones del sistema operativo:

## Control de ejecución de programas (procesos)

39

1. **FCFS** (First Come First Served o **FIFO** – First In First Out): Al primer proceso que llegue se le asignan tiempos o ciclos de CPU hasta que termina completamente. No apropiativo
2. **SJF** (Shortest Job First): Se ejecuta primero el que dure menos tiempo. Es no apropiativo, es decir si llega un proceso que tarde menos en ejecutarse se termina primero el actual.
3. **SRTF** (Shortest Remaining Time First, o **SRT**) – Es como el SJF, pero apropiativo.
4. **Round-Robin**: Asigna por rondas tiempos de ejecución (determinado por el **quantum**) a los diferentes procesos de forma secuencial (empleando una cola). Appropriativo
5. **Algoritmos con prioridad**: Cada proceso tiene una prioridad de ejecución asignada. En estos algoritmos para evitar que un proceso esté en un estado de bloqueo indefinido o **inanición** se suele aumentar su prioridad a medida que pasa el tiempo en el sistema. Puede ser apropiativo o no apropiativo



# Ejemplo algoritmo FIFO (FCFS)

40

## ACTIVIDAD RESUELTA

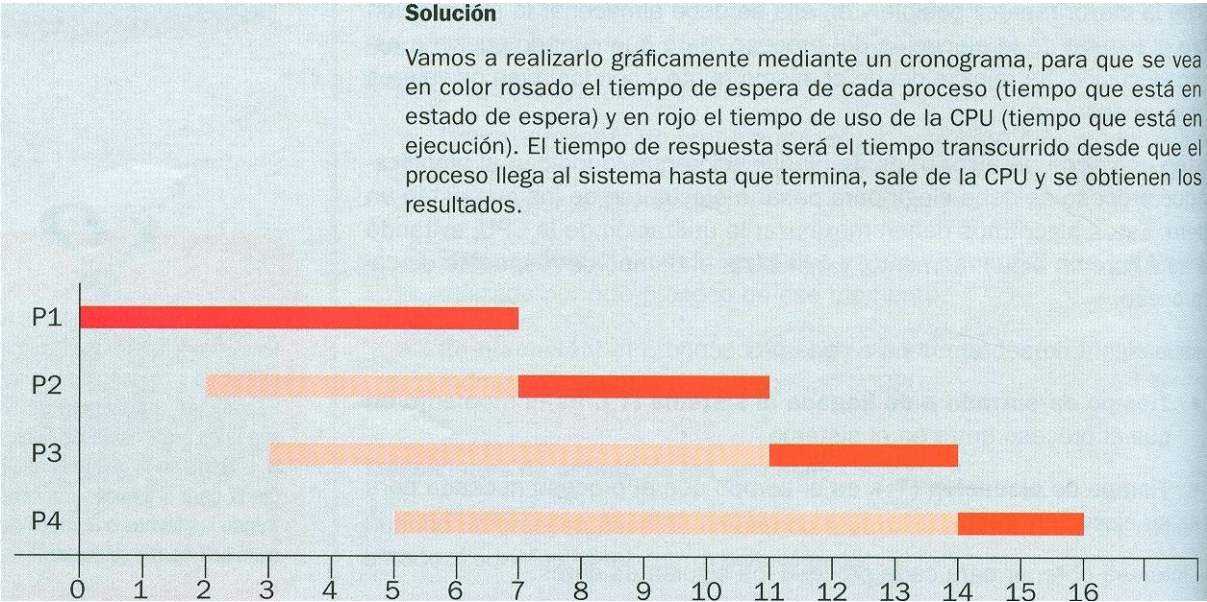
Para los siguientes procesos, con el tiempo de llegada y el tiempo de ejecución necesario que se indica, calcula los tiempos de espera y respuesta de cada proceso y los tiempos medios, utilizando el algoritmo FIFO o FCFS.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	5	9
P3	8	11
P4	9	11
<b>TIEMPOS MEDIOS</b>	<b>5,5</b>	<b>9,5</b>

### Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.





# Ejemplo algoritmo SJF

41

## ACTIVIDAD RESUELTA

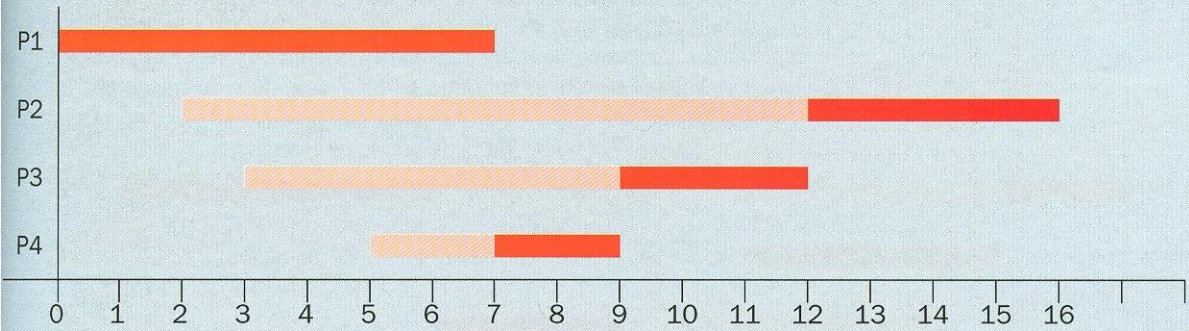
Para los siguientes procesos, con el tiempo de llegada y el tiempo de ejecución necesario que se indica, calcula los tiempos de espera y respuesta de cada proceso y los tiempos medios, utilizando el algoritmo SJF.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	10	14
P3	6	9
P4	2	4
<b>TIEMPOS MEDIOS</b>	<b>4,5</b>	<b>8,5</b>

## Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.





# Ejemplo algoritmo SRTF

42

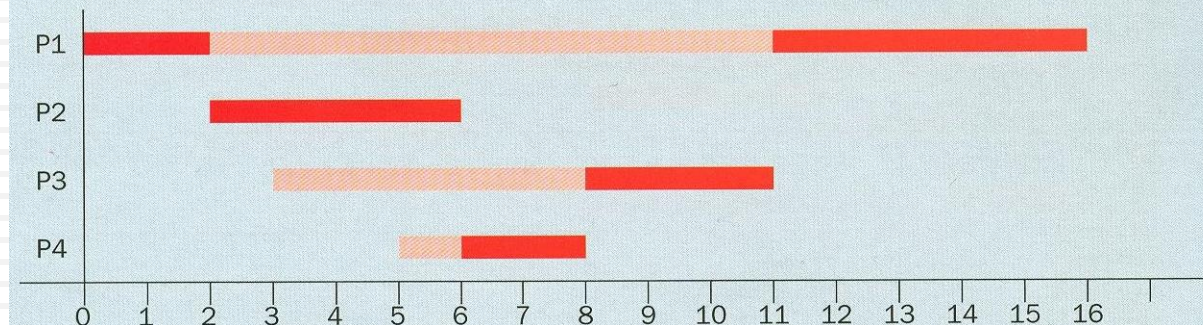
## ACTIVIDAD RESUELTA

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	9	16
P2	0	4
P3	5	8
P4	1	3
<b>TIEMPOS MEDIOS</b>	<b>3,75</b>	<b>7,75</b>

## Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.





# Ejemplo algoritmo con prioridades

ACTIVIDAD RESUELTA

Para los siguientes procesos, con el tiempo de llegada y el tiempo de ejecución necesario que se indica, calcula los tiempos de espera y respuesta de cada proceso y los tiempos medios, utilizando el algoritmo **por prioridades no expulsivo**.

PROCESO	TIEMPO DE LLEGADA	PRIORIDAD	TIEMPO DE EJECUCIÓN
P1	0	4	7
P2	2	2	4
P3	3	1	3
P4	5	3	2

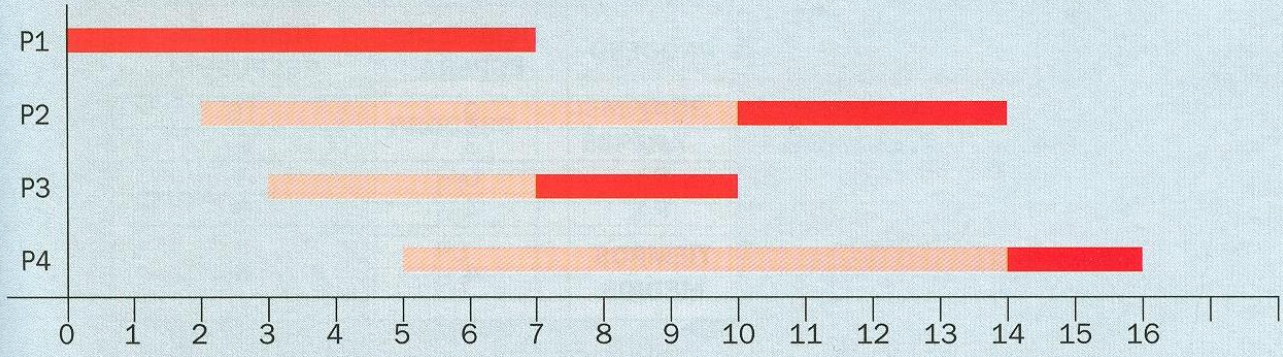
No expulsivo = no apropiativo

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	8	12
P3	4	7
P4	9	11
TIEMPOS MEDIOS	5,25	9,25

La prioridad se suele indicar con un número entero de modo que cuanto más bajo sea, mayor prioridad tendrá el proceso.

Solución

Vamos a realizarlo gráficamente mediante un cronograma, para que se vea en color rosado el tiempo de espera de cada proceso (tiempo que está en estado de espera) y en rojo el tiempo de uso de la CPU (tiempo que está en ejecución). El tiempo de respuesta será el tiempo transcurrido desde que el proceso llega al sistema hasta que termina, sale de la CPU y se obtienen los resultados.



# Ejemplo algoritmo Round Robin (I)

44

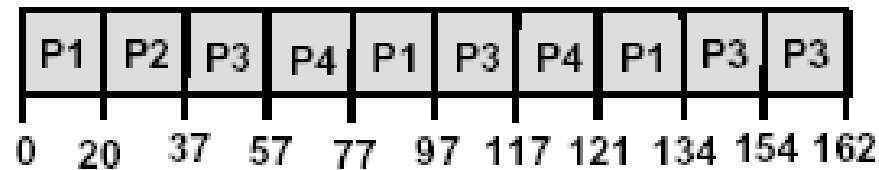
- Se debe establecer un tiempo o quantum ( $q$ ), tras el cual el proceso abandona la CPU y da paso al siguiente, siguiendo el orden FIFO.

- ◆ Ejemplo:

<u>Proceso</u>	<u>Ráfaga CPU</u>
P1	53
P2	17
P3	68
P4	24

- Con  $q=20$

✓ Diagrama de Gant para la planificación:



✓ 9 cambios de contexto

# Ejemplo algoritmo Round Robin (II)

45

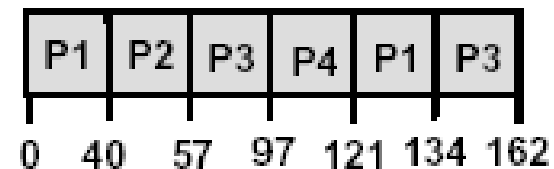
*Si no hay procesos en espera, el que está usando la CPU seguirá hasta que llegue algún otro proceso.*

- ◆ Ejemplo:

<u>Proceso</u>	<u>Ráfaga CPU</u>
P1	53
P2	17
P3	68
P4	24

- Con  $q=40$

✓ Diagrama de Gant para la planificación:



✓ 5 cambios de contexto

## Funciones del sistema operativo:

### 2. Administración de la memoria

46

- *La memoria principal es un recurso que el SO tiene que gestionar.*
- El **objetivo** del gestor de memoria es compartir la memoria entre un conjunto de procesos.
- El **espacio físico de direcciones** son las direcciones “reales” de la memoria física.
- El **espacio lógico de direcciones** son las direcciones relativas o lógicas, que el sistema operativo con ayuda del hardware convierte en direcciones reales o físicas.

# Funciones del sistema operativo:

## Administración de la memoria

47

- Un programa se dice que es **reubicable** si puede ser cargado y ejecutado en cualquier espacio de memoria.
- **Reasignar** es convertir las direcciones lógicas de un programa en direcciones físicas.
  - ▣ Existen dos formas de hacerlo:
    - **Reasignación estática** – Se realiza durante la carga del programa en memoria.  
El cargador sumará a cada dirección lógica la dirección de carga, obteniendo así las direcciones reales.
    - **Reasignación dinámica** – Se realiza durante la ejecución del programa.  
Es necesaria la existencia de un registro base o registro de reubicación, cuyo valor se suma a toda dirección generada por el proceso cuando se lleva a memoria.  
Se realiza en tiempo de ejecución

# Funciones del sistema operativo:

## Administración de la memoria

48

### □ **Sistemas de gestión de memoria**

▣ *Los sistemas de gestión de memoria pueden dividirse en:*

■ **Monoprogramación:** esquema de memoria más simple.

Consiste en mantener la memoria ocupada con un solo proceso.

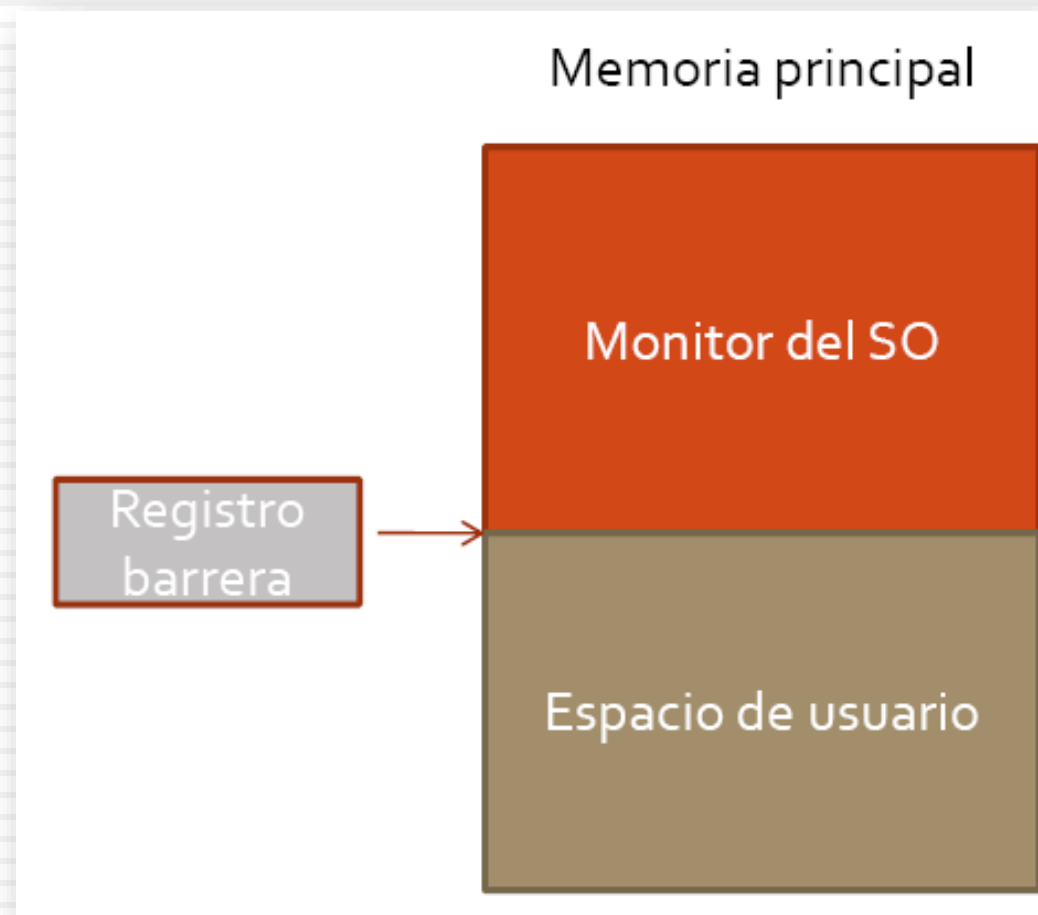
■ **Multiprogramación:** varios procesos se ejecutan al mismo tiempo, por lo que será necesario repartir la memoria entre ellos.



# Funciones del sistema operativo: Administración de la memoria

49

## □ Monoprogramación



# Funciones del sistema operativo:

## Administración de la memoria

50

### □ Multiprogramación

- Una forma de conseguir la **multiprogramación** consiste en dividir la memoria disponible en **regiones** o **particiones**, cada una de las cuales se asignará a un proceso o a partes de un proceso.
- Si todas las partes de un programa se cargan juntas en la memoria se dice que la asignación es **contigua** y si, por el contrario, está cargado en diferentes partes, se dice que su asignación es **no contigua**.

# Funciones del sistema operativo:

## Administración de la memoria

51

### □ **Multiprogramación: asignación contigua**

1. Regiones o particiones de tamaño fijo (**MFT**)
2. Regiones o particiones de tamaño variable (**MVT**)

### □ **Multiprogramación: asignación no contigua**

1. Regiones del mismo tamaño (**Paginación**)
2. Regiones de diferente tamaño (**Segmentación**)

# Funciones del sistema operativo:

## Administración de la memoria

52

### □ **MFT: particiones de tamaño fijo**

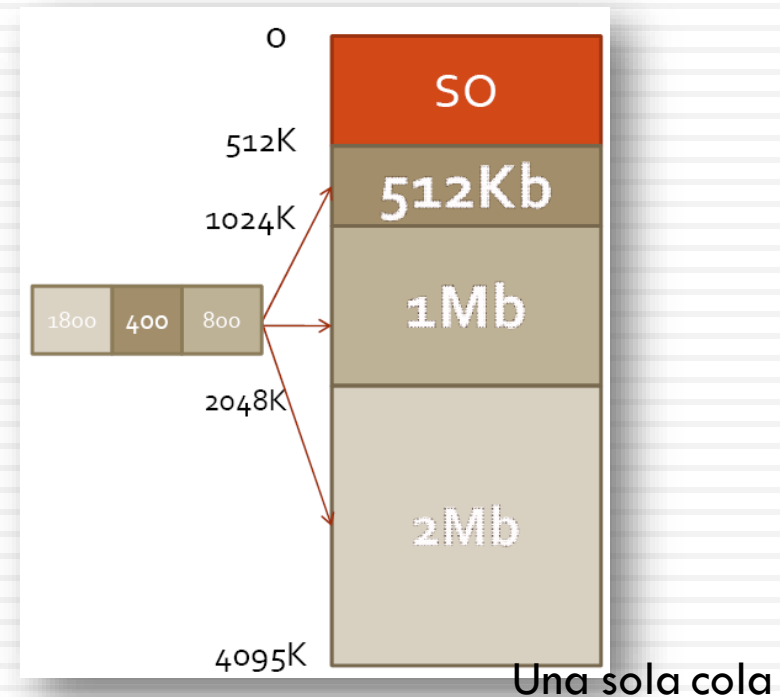
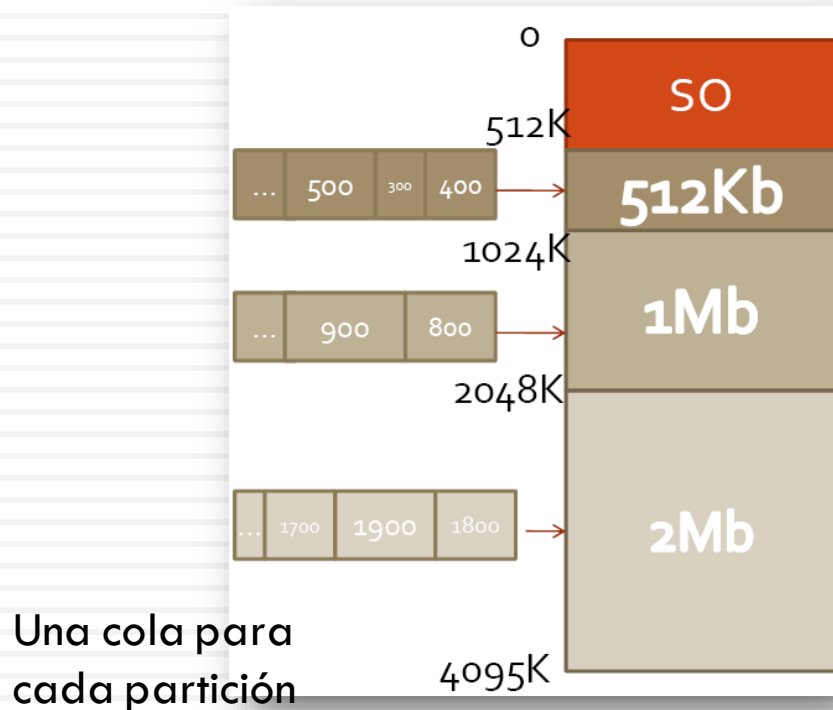
- ▣ **Regiones Fijas** – Los tamaños de región son fijos y no cambian durante la ejecución.
- ▣ Las regiones no tienen porque tener el mismo tamaño.
- ▣ Cuando queda disponible una región se busca en la cola el primer trabajo que se pueda situar en esa región, se selecciona y se le asigna la región, incluso si hay trabajos de mayor prioridad (pero mayor tamaño que no pueden utilizar la región por ser demasiados grandes).

# Funciones del sistema operativo: Administración de la memoria

53

## □ MFT: particiones de tamaño fijo

- ▣ Las técnicas de asignación se basan en:
  - Establecer una cola para cada partición de memoria.
  - Establecer una cola única para toda la memoria.



# Funciones del sistema operativo:

## Administración de la memoria

54

### □ **MFT: particiones de tamaño fijo**

▣ Los algoritmos de planificación pueden ser:

- **Primer ajuste** (First fit) – El primero suficientemente grande.
- **Mejor ajuste** (Best fit) – El hueco mas pequeño de todos suficientemente grande.
- **Exploración de cola** – Se asigna memoria a los trabajos hasta que, finalmente los requerimientos de memoria del trabajo siguiente no pueden ser satisfechos (ningún bloque de memoria disponible es suficientemente grande).  
Si el planificador desciende por la cola de trabajos para ver si algún trabajo de menor prioridad se puede colocar en el hueco de memoria disponible se dice que tiene exploración de cola.

# Funciones del sistema operativo:

## Administración de la memoria

55

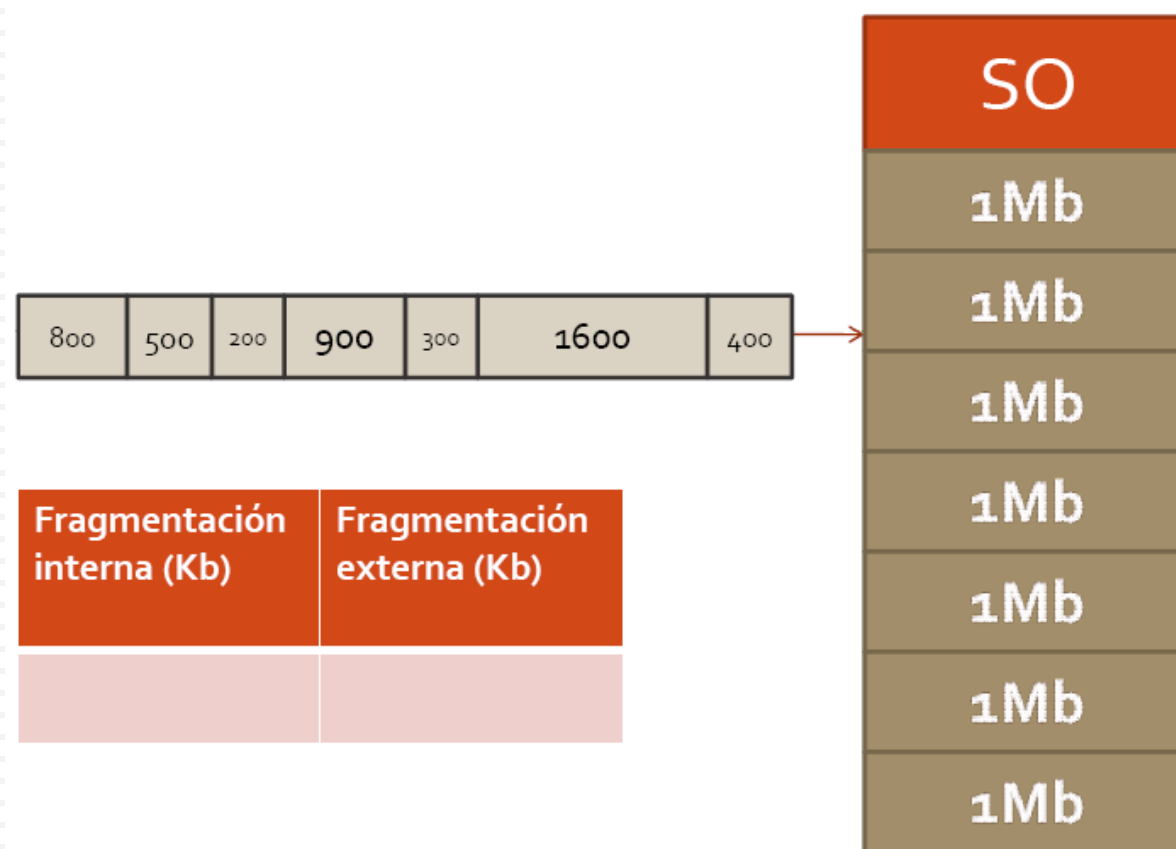
### □ **MFT: particiones de tamaño fijo. Problema:**

- Desaprovechamiento de memoria: **fragmentación**
- Dos **tipos**:
  - **Fragmentación interna**: aquella parte de la memoria que no se está usando pero que es interna a una partición asignada a una tarea.
  - **Fragmentación externa**: cuando una partición disponible no se emplea porque es muy pequeña para cualquiera de las tareas que esperan.

# Funciones del sistema operativo: Administración de la memoria

56

- **MFT: particiones de tamaño fijo.**  
**Ejemplo fragmentación:**





# Funciones del sistema operativo:

## Administración de la memoria

57

- **MFT: particiones de tamaño fijo.**  
**Ejemplo fragmentación:**

Fragmentación interna (Kb)	Fragmentación externa (Kb)
3044 Kb	1024 Kb

Se vé que hay un trabajo de 1600 Kb que no se podrá ejecutar nunca

los otros sin embargo si se ejecutarán pero en conjunto desperdiciarán 4068 Kb, casi la mitad de la memoria disponible



# Funciones del sistema operativo:

## Administración de la memoria

58

### □ **MVT: particiones de tamaño variable**

- ▣ Inicialmente, *toda la memoria está disponible y se considera como un gran bloque.*
- ▣ El sistema operativo mantiene una tabla en la que se indica **qué partes de la memoria están disponibles y cuales están ocupadas.**
- ▣ Cuando llega un trabajo y precisa memoria buscamos un hueco que sea suficientemente grande para ese trabajo.
- ▣ Si encontramos uno, **le asignamos sólo la memoria que sea necesaria,** manteniendo el resto disponible para futuras solicitudes.

# Funciones del sistema operativo:

## Administración de la memoria

59

### □ **MVT: particiones de tamaño variable**

#### ▣ Técnicas de asignación:

- **Primer ajuste** (First fit) – El primero suficientemente grande.
- **Mejor ajuste** (Best fit) – El hueco mas pequeño de todos suficientemente grande.
- **Peor ajuste** (Worst fit) – Se asigna al hueco mas grande.

#### ▣ **No existe fragmentación interna.**

#### ▣ **Existe fragmentación externa.**

- Para resolverlo se puede realizar **compactación**. Se mueven todos los trabajos hacia un extremo de la memoria. Todos los huecos se mueven en dirección opuesta, produciendo un gran hueco de memoria disponible. Solo es posible realizar una compactación si la reasignación de memoria es dinámica. Conlleva mucho trabajo de procesador.

# Funciones del sistema operativo:

## Administración de la memoria

60

### □ **Multiprogramación: asignación no contigua**

#### 1. **Regiones del mismo tamaño (Paginación)**

- La paginación admite que la memoria de un programa sea **no contigua**, permitiendo de este modo la asignación a un programa de memoria física en cualquier lugar donde esté disponible.
- La memoria física se fragmenta en bloques de tamaño fijo llamados marcos de página (**frames**).

# Funciones del sistema operativo:

## Administración de la memoria

61

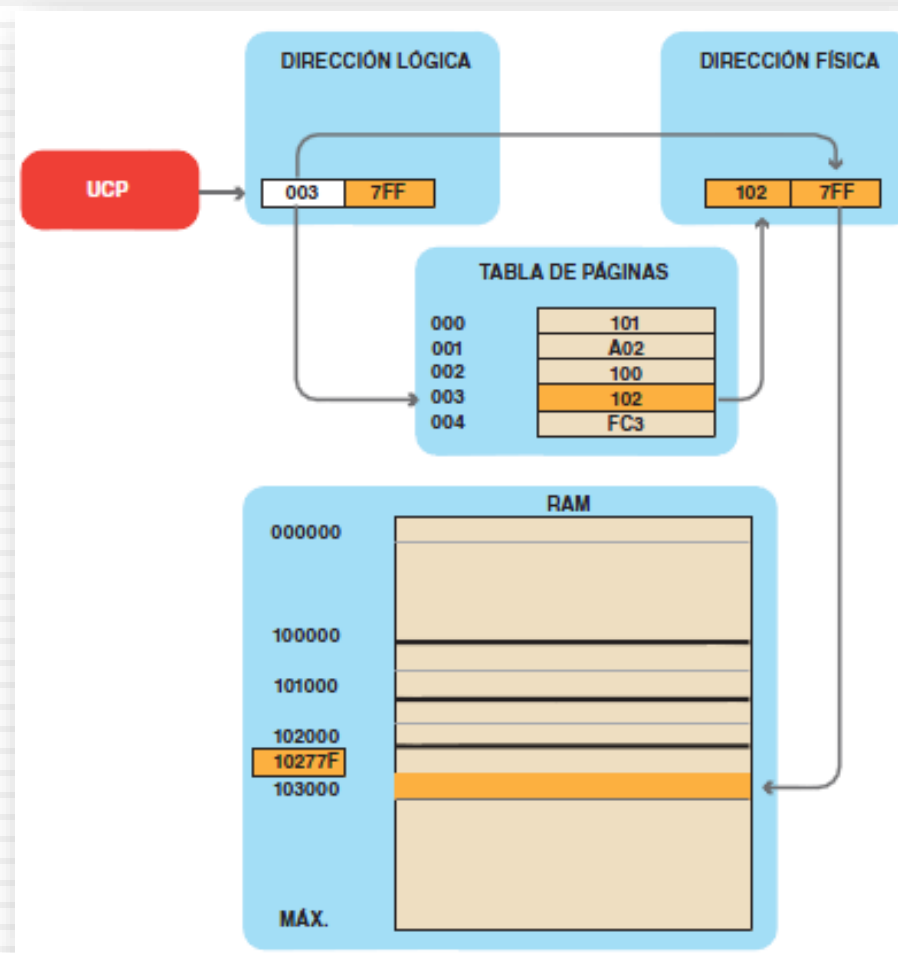
### □ **Multiprogramación: asignación no contigua. Paginación**

- ▣ Cuando hay que ejecutar un programa se cargan sus páginas en cualquiera de las celdas que haya disponibles.
- ▣ El SO mediante una **tabla de páginas** relaciona la pagina lógica con la pagina física o frame.
- ▣ **Suele existir fragmentación interna en la última página asignada.**
- ▣ **No existe fragmentación externa.**

# Funciones del sistema operativo: Administración de la memoria

62

## □ Paginación



# Funciones del sistema operativo:

## Administración de la memoria

63

### □ **Multiprogramación: asignación no contigua**

#### 2. **Regiones de diferente tamaño (Segmentación)**

- Es una técnica similar a la paginación que permite definir los bloques de memoria de **tamaño variable**.
- El sistema operativo debe de mantener para cada segmento cual es su dirección base (donde comienza) y el límite (tamaño máximo que tiene el segmento).
- **No existe fragmentación interna.**
- **Existe fragmentación externa**

## Funciones del sistema operativo:

### 3. Gestión de E/S. Periféricos

64

- Otra de las funciones del SO es la de controlar todos los dispositivos de entrada/salida (periféricos), proporcionando una interfaz entre los dispositivos y el resto del sistema que sea sencilla y fácil de usar.
- *Las operaciones de E/S se pueden realizar mediante una gran variedad de dispositivos externos que proporcionan diferentes formas de intercambiar datos entre el mundo exterior y el computador.*



# Funciones del sistema operativo:

## Gestión de E/S. Periféricos

65

□ Los periféricos se clasifican en:

1. **Periféricos tipo bloque.** La información que se lee o escribe en estos dispositivos es de tamaño fijo (bloque).

Son **direccionables**, se puede leer o escribir cualquier bloque, sin pasar por los anteriores.

*Ejemplos: discos duros, discos ópticos, disquetes, etc.*

2. **Periféricos tipo carácter:** trabajan con secuencias de bytes sin importar su longitud. **NO** son **direccionables**.

*Ejemplos: teclado, ratón, tarjeta de red, etc.*

# Funciones del sistema operativo:

## Gestión de E/S. Periféricos

66

- Los **dispositivos físicos** que el sistema operativo tiene que gestionar para el intercambio de información se clasifican según su función en:
  1. **Soportes de almacenamiento:** Memoria auxiliar del ordenador (*discos duros, unidades de CD-ROM, etc.*).
    - *Interfaz tipo texto.*
    - *Interfaz tipo grafico.*
  2. **Interfaces:** Permiten la comunicación entre el usuario y el SO (teclado, ratón, pantalla, etc.).
  3. **Soportes de transmisión** – Buses y canales encargados de transmitir la información entre los diferentes componentes del ordenador.

# Funciones del sistema operativo:

## Gestión de E/S. Periféricos

67

- Los dispositivos de E/S tienen 2 partes:
  1. **Mecánica:** Lleva a cabo las funciones físicas.
  2. **Electrónica:** A través de ésta se accede a la parte mecánica (***controlador del dispositivo***).

Proporciona una interfaz estándar que se adapta a las peculiaridades del hardware que esta por debajo.

# Funciones del sistema operativo:

## Gestión de E/S. Periféricos

68

### □ Acceso Directo a Memoria (DMA)

- Permite que algunos dispositivos (*en particular los de bloque*) accedan a memoria **sin intervención de la CPU** (*más rendimiento*).
- La CPU proporciona información sobre la dirección de memoria para acceder y la cantidad de datos a transferir; después se despreocupa.
- Una vez terminado, el controlador genera una **interrupción** para que la CPU tome el control de nuevo.

# Funciones del sistema operativo:

## Gestión de E/S. Periféricos

69

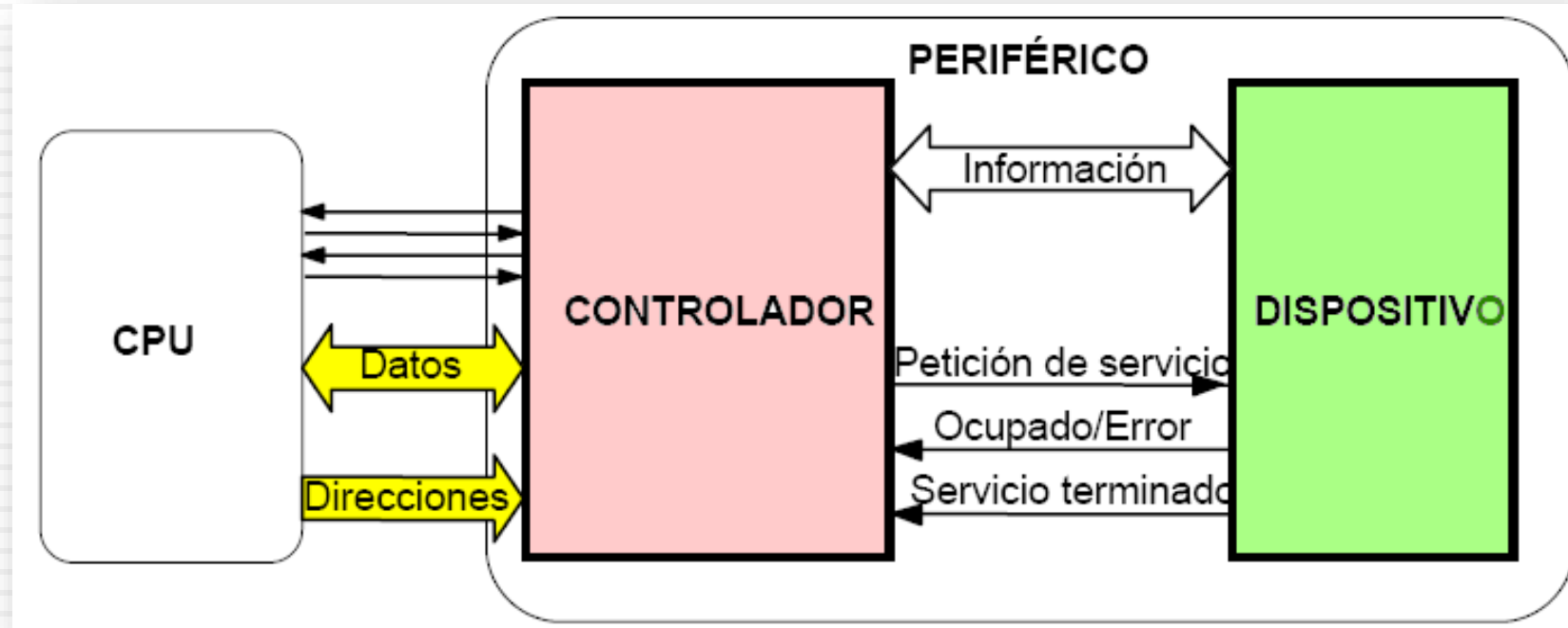
### □ **Driver (controlador)**

- ▣ Es un Software específico para cada periférico y sistema operativo (dependiente de la arquitectura en general).
- ▣ Aceptan solicitudes genéricas por parte del sistema operativo y las transforman en instrucciones dependientes del hardware.
- ▣ Si el dispositivo está ocupado, el controlador o manejador se encarga de gestionar **una cola de peticiones**.

# Funciones del sistema operativo:

## Gestión de E/S. Periféricos

70



## 4. Gestión de la información

- El sistema operativo gestiona la información mediante el **sistema de archivos**.
- Para cada archivo almacenado en el soporte, se indica el nombre, tamaño, tipo, fecha y hora de grabación, lugar del soporte en el que se encuentra, etc.
- Los nombres de archivos siguen el formato (**<nombre-archivo>[.<extensión>]**).

# Funciones del sistema operativo:

## Gestión de la información

72

- ❑ Algunos SO como LINUX distinguen entre mayúsculas y minúsculas.
- ❑ Otros como MS-DOS limitan el nombre a 8 caracteres para el nombre y 3 para la extensión.
- ❑ Las extensiones deben de incluirse en los nombres de archivos puesto que permiten diferenciar rápidamente el tipo de archivo de que se trata.
- ❑ En Linux no es imprescindible la extensión.



# Funciones del sistema operativo:

## Gestión de la información

73

### □ ***Tipos de archivos:***

- ▣ **Archivos regulares o estándares:** Contienen información del usuario.
- ▣ **Directorios:** Contienen referencias a otros directorios o archivos.
- ▣ **Archivos especiales:** Archivos del sistema (paginación, archivos de dispositivos, librerías, etc.).
- ▣ **Atributos:** Información que caracteriza al archivo (solo lectura, oculto, modificado, sistema).

# Funciones del sistema operativo:

## Gestión de la información

74

- *Los **sistemas de archivos** varían de un sistema operativo a otro, algunos de ellos son:*
  - ▣ **FAT16 y FAT32** – Empezó a utilizarse en MS-DOS y Windows.
  - ▣ **NTFS** - Utilizado por diferentes versiones de Windows, incluido Windows 7.
  - ▣ **Ext, Ext2, Ext3, Ext4, Swap** - Linux.
  - ▣ **HPFS** – OS/2.
  - ▣ **S5** – UNIX.
  - ▣ **CDFS** – Unidad de CD-ROM.

# Funciones del sistema operativo:

## Gestión de la información

75

### □ *Particiones de disco*

- Las **particiones** son divisiones lógicas de un dispositivo físico que permiten crear varias unidades dentro de un solo disco duro.

### ■ **Tipos:**

- **Primarias:** Puede existir un máximo de 4 por dispositivo. Una de ellas debe estar "activa". Destinadas a albergar la instalación de SO y datos.
- **Extendida:** Solución a la limitación del máximo de 4 particiones primarias. No esta destinada a crear un **sistema de archivos** en ella, sino que permite albergar un numero indeterminado de **particiones lógicas**.
- **Lógicas:** Creadas dentro de la partición extendida. Destinadas a almacenar datos. Si se instala un SO en una partición lógica, es posible que no se pueda accede desde este a las demás particiones lógicas.

# Funciones del sistema operativo:

## Gestión de la información

76

### □ Particiones de disco

The screenshot shows the 'Administración de equipos' (Computer Management) console, specifically the 'Administración de discos' (Disk Management) section. It displays a list of disks and their partitions.

Volumen	Disposición	Tipo	Sistema de archivos	Estado	Capacidad	Espacio disponible	% disponible
(G:)	Simple	Básico	FAT	Correcto (Partición primaria)	1,86 GB	1,85 GB	99 %
OS (C:)	Simple	Básico	NTFS	Correcto (Arranque, Archivo de paginación, Volcado, Partición primaria)	451,07 GB	222,52 GB	49 %
RECOVERY	Simple	Básico	NTFS	Correcto (Sistema, Activo, Partición primaria)	14,65 GB	8,41 GB	57 %
WD_Ana (F:)	Simple	Básico	NTFS	Correcto (Partición primaria)	465,73 GB	374,15 GB	80 %

Disco	Partición	Detalles
Disco 0 Básico 465,76 GB En pantalla	39 MB Correcto (Partición)	RECOVERY 14,65 GB NTFS Correcto (Sistema, Activo, Partición primaria)
Disco 1 Básico 465,73 GB En pantalla	WD_Ana (F:) 465,73 GB NTFS Correcto (Partición primaria)	
Disco 2 Extraíble 1,86 GB En pantalla	(G:) 1,86 GB FAT Correcto (Partición primaria)	
CD-ROM 0 DVD (D:)	No hay medios	

Legend: ■ No asignado ■ Partición primaria

## Funciones del sistema operativo:

### 5. Otras funciones del sistema operativo

- Otras funciones que también lleva a cabo el sistema operativo son:
  - ▣ *Gestión de los permisos y usuarios*
  - ▣ *Administración de la red*
  - ▣ *Gestión de la seguridad del sistema*
  - ▣ *Control de errores*
  - ▣ *Intérprete de comandos*