

# Gestión de los recursos de un sistema operativo

# Objetivos

- Identificar los procesos y sus estados.
- Determinar las características y elementos de los procesos.
- Planificar la ejecución de los procesos.
- Interpretar las técnicas de gestión de memoria.
- Diferenciar las técnicas de gestión de memoria.
- Conocer la gestión de entrada/salida del sistema operativo.

# Contenidos

- 1. Procesos y flujos.
- 2. Hebras y estados de los procesos.
- 3. Transición de los procesos.
- 4. Bloque de control de procesos.
- 5. Algoritmos de planificación.
- 6. Memoria RAM y memoria virtual.
- 7. Intercambio.
- 8. Paginación, segmentación y swapping.
- 9. Programas reubicables, reentrantes, residentes y reutilizables.

# Contenidos

- 10. Gestión de entrada/salida: tipos de periféricos.
- 11. Comunicación con el sistema: interfaces de usuario.
- 12. Clasificación de los periféricos.
- 13. Gestión de la información.

# 1. Procesos y flujos

---

# Procesos y flujos.

Un **proceso** es un concepto manejado por el sistema operativo y que referencia un programa en ejecución.

Según el contexto dentro de cada SO a los procesos se les denomina flujos de control, tareas, *threads* o hilos.

La CPU gestiona la solicitud de recursos de los procesos, además de realizar funciones de sincronización de todos los procesos, para que se ejecuten en el orden adecuado y según la prioridad asignada.

Una **estructura de datos** se asocia al proceso de ejecución de los procesos cuando son ubicados sus datos que lo componen y sus datos asociados en memoria.

El **bloque de control de proceso (BCP)** contiene para cada proceso la siguiente información: estado actual del proceso, identificador del proceso, prioridad del proceso, ubicación en memoria y recursos utilizados.

## 2. Hebras y estados de los procesos

---

# Hebras y estados de los procesos.

Una **hebra o hilo** es un punto de ejecución de un proceso. Un proceso tendrá siempre un hilo, en la que corre el programa, pero puede tener más hilos.

Los hilos representan un método para mejorar el rendimiento y eficacia de los sistemas operativos. Los hilos de un mismo proceso compartirán recursos, como memoria, archivos, recursos hardware, etc...



# Hebras y estados de los procesos.

Un estado puede estar en diferentes estados:

**En ejecución:** El procesador está ejecutando instrucciones del programa que lo compone y tiene concedido el tiempo de uso del CPU en un instante concreto.

**Preparado, en espera o activo:** Un proceso está preparado para ser ejecutado.

**Bloqueado:** El proceso está retenido, por causas múltiples.

El PID (identificador del proceso).

# Hebras y estados de los procesos.



# 3. Transición de los procesos

---

# Transición de los procesos.

Una vez que un programa se ha lanzado y se ha convertido en proceso, puede atravesar varias fases o **estados** hasta que finaliza o termina.

Cuando un proceso se lanza, nunca se ejecuta directamente, sino que se coloca en la **cola de procesos** en un estado denominado **preparado**. Cuando la CPU le asigna su tiempo, el proceso pasa de preparado a **ejecución**. Estos dos estados se alternarán en caso de que esté ejecutando más de un proceso en el sistema.

Los cambios de estado en los que se puede encontrar un proceso se denominan **transiciones**.

# Transición de los procesos.

Los diferentes estados tiene una relación directa con las **prioridades**, que son aquellas que el administrador del sistema o el propio sistema, asignan cada proceso. De ello dependerá que un proceso se ejecute en más o menos tiempo.

Gracias a los **algoritmos de planificación** podemos decidir qué proceso ha de ejecutarse en cada momento y por qué.

Cuando se están ejecutando varias tareas a la vez, es necesario compartir el trabajo de la CPU. Las porciones de tiempo de los procesos en ejecución se denominan **quantum**.

## 4. Bloque de control de procesos

---

# Bloque de control de procesos.

El **BCP** de cada proceso almacena información como:

- **Estado actual del proceso:** Ejecución, bloqueado, preparado.
- **Identificador del proceso:** Dependiendo del sistema operativo, a cada proceso se le asigna un PID.
- **Prioridad del proceso:** Es asignada por el planificador.
- **Ubicación en memoria:** Dirección de memoria en la que se carga el proceso.
- **Recursos utilizados:** Recursos de hardware y software para poder ejecutarse.

En programas multi hilo, el BCP puede contener además el PPID (Process Parent IDentification). Este dato referencia el PID del proceso padre dentro del BCP.

# 5. Algoritmos de planificación

---



# Algoritmos de planificación.

- Algoritmo de la rueda:

A cada uno se le asigna el mismo intervalo de tiempo de ejecución.

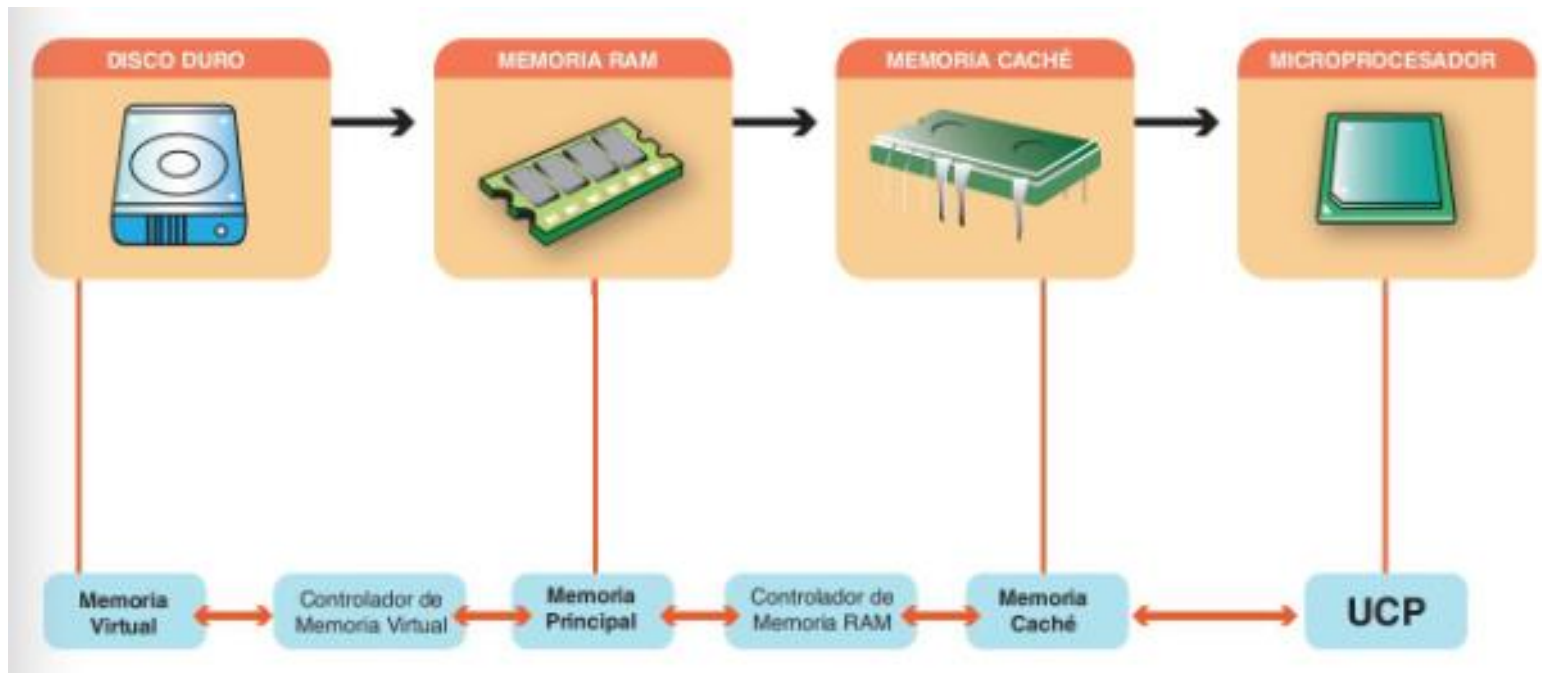
- Algoritmo FIFO

Al primer proceso que llega se le asignan tiempos de CPU hasta que termina completamente.

## 6. Memoria RAM y memoria virtual

---

# Memoria RAM y memoria virtual.



# 7. Intercambio

---

# Intercambio.

La parte del sistema operativo que administra la memoria es el **administrador de memoria**. Su labor es llevar un registro de las partes de memoria que se están utilizando y de las que no. De esta forma, reserva espacio de memoria para los nuevos procesos y libera el espacio de los procesos que finalizan.

También se encarga de gestionar el intercambio de datos entre memoria y disco, siempre y cuando los procesos sean tan grandes que no quepan de una sola vez en memoria.

# Intercambio.

El **planificador** tiene en cuenta los requerimientos de memoria de cada uno de los procesos y las particiones de memoria disponibles. Estos requerimientos de uso de memoria se almacenan en el **BCP**.

Las particiones presentan dificultades en su diseño, para solucionarlo los procesos podrán utilizar memoria no contigua, mediante **técnicas de paginación**, dichas técnicas utilizan un sistema de traducción de direcciones lógicas a físicas mediante una **tabla de páginas**. Esta tabla presenta dos cuestiones: el tamaño (que puede ser demasiado grande) y el tiempo de asignación (que debe ser de corta duración).

# Intercambio.

En contraposición, está la concepción por parte del usuario de considerar la memoria como un conjunto de segmentos de diferentes tamaños, sin ninguna ordenación entre ellos. Este esquema corresponde a la **segmentación**.

La idea de permitir ejecutar los procesos que no están cargados totalmente en memoria, e incluso que sus tamaños superen al de la memoria física instalada da lugar al concepto de **memoria virtual**.

## 8. Paginación, segmentación y swapping

---



# Paginación, segmentación y swapping.

Son técnicas de gestión de memoria, que en general permiten ejecutar programas de un tamaño superior a la capacidad de la memoria RAM utilizando el disco duro como una «ampliación» de la memoria principal del equipo.

# Paginación, segmentación y swapping.

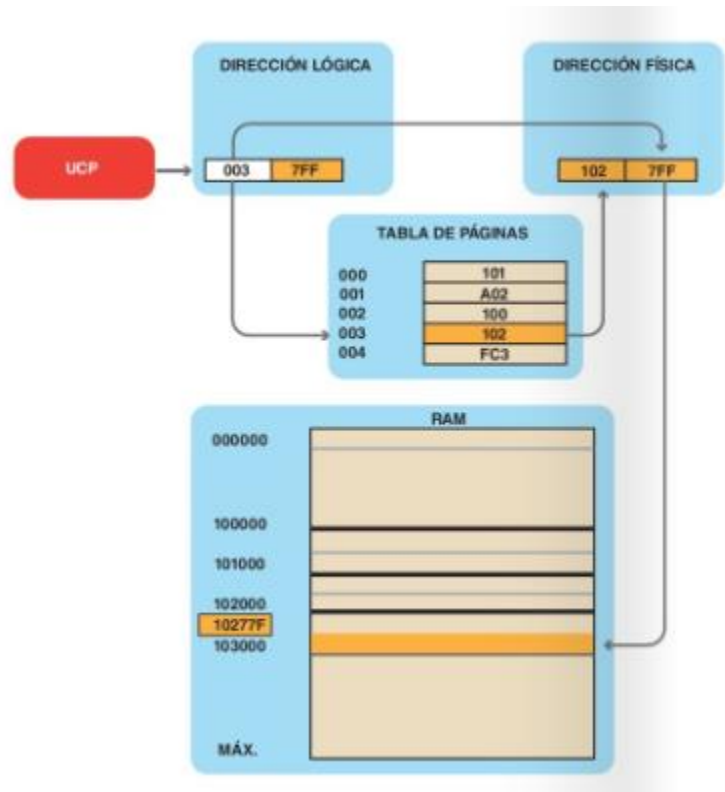
## Paginación

Es una técnica que consigue en dividir la memoria interna o RAM en zonas iguales llamadas **frames**, y los programas en partes del mismo tamaño denominado **páginas**.

La paginación es una técnica de reasignación o re direccionamiento dinámico, con la consideración de que la tabla de páginas se puede almacenar en registros especiales destinados a tal efecto o en una parte de la propia memoria.

# Paginación, segmentación y swapping.

## Paginación



# Paginación, segmentación y swapping.

## Segmentación

Es una técnica similar a la paginación que permite definir los bloques de memoria de tamaño variable. Cada **segmento** puede variar desde 0 hasta un máximo permitido. La longitud de un segmento puede variar según las necesidades del programa.

El ordenador, puede organizar la memoria en bloques concretos y tener partes de ella destinadas a almacenar las estructuras de datos, que pueden crecer o menguar según las necesidades del usuario del programa. Para ello se utilizarán las pilas de memoria o **stacks**, en las que se gestionan las estructuras de datos necesarias.

La paginación se diferencia de la segmentación en que las paginas son de tamaño fijo y los segmentos no.

# Paginación, segmentación y swapping.

## Swapping

Es una técnica similar a la memoria virtual. Según el estado en el que se encuentre el proceso de cada usuario, la memoria se irá liberando de su proceso y pasará a la zona de **swap** mediante la técnica llamada **swap-out**. De esta forma, la memoria interna queda liberada para que en ella se pueda almacenar otro proceso del mismo usuario o de otro.

Si el usuario vuelve a solicitar su proceso para seguir ejecutándolo, se produce el denominado swap-in, que consiste en pasar el programa de la zona swap a la memoria interna.

# Paginación, segmentación y swapping.

## Swapping

Los fabricantes recomiendan que la zona del swap sea del 20% del espacio en disco o del doble de la capacidad de RAM del ordenador.

La diferencia entre la gestión de memoria virtual y el swapping es que, mediante la primera, puede llegar a ocurrir que el disco esté tan lleno que la gestión sea difícil o imposible, ya que el espacio destinado al intercambio suele ser espacio del disco duro en el que está instalado tanto el sistema operativo como las aplicaciones y los datos del usuario.

En el swapping no puede ocurrir esto, ya que esta zona siempre estará reservada.

## 9. Programas reubicables, reentrantes, residentes y reutilizables

---

# Programas reubicables, reentrantes, residentes y reutilizables.

Según cómo, dónde y cuándo se ubiquen en memoria, los programas pueden ser de varios tipos:

**Reubicables:** Una vez cargados en RAM para ejecutarse, pueden variar de situación, ya que la parte de RAM que ocupan puede ser necesaria para ubicar otro proceso (configuración interna del propio ordenador).

**Residentes:** Una vez cargados en memoria, permanecerán en ella hasta que se apague el ordenador.



# Programas reubicables, reentrantes, residentes y reutilizables.

**Reentrantes:** Si no se están ejecutando, dejan la memoria libre para otros procesos. Estos procesos, cuando se liberan, se suelen almacenar temporalmente en el disco duro. Son los procesos gestionados mediante la técnica de memoria virtual.

**Reutilizables:** Son programas que normalmente son utilizados por varios usuarios a la vez en memoria, independientemente del número de usuarios que los vayan a utilizar.

# 10.Gestión de entrada/salida: tipos de periféricos

---

# Gestión de entrada/salida: tipos de periféricos.

Una forma de clasificar los periféricos es si gestionan la información por bloques o por caracteres:

**Periféricos tipo bloque:** Son aquellos en que la información que se maneja es de tamaño fijo. La información entra o sale de memoria en forma de bloque.

**Periféricos tipo carácter:** Son los que sirven para introducir o sacar datos en memoria en forma de caracteres.

Un periférico está compuesto por un componente mecánico y uno o muchos electrónicos. El SO se encarga de conectar los distintos dispositivos con la memoria.

# 11. Comunicación con el sistema: interfaces de usuario

---

# Comunicación con el sistema: interfaces de usuario.

**Interfaces tipo texto**

**Interfaz tipo gráfico**

# 12. Clasificación de los periféricos

---

# Clasificación de los periféricos.

- **De entrada**
- **De salida**
- **De entrada y salida**

# 13. Gestión de la información

---



# Gestión de la información.

Cada sistema operativo utiliza su propio **sistema de archivos**. En general los tipos de archivos que gestiona todo el sistema operativo son tres:

**Archivos regulares:** Contienen información del usuario: programas, documentos...

**Directorios:** Son archivos que contienen referencias a otros archivos regulares.

**Archivos especiales:** Los que no son de ninguno de los dos tipos anteriores.