

# Sistemas Operativos

Grado en Ingeniería Informática del Software

## Tema 2: Gestión de procesos

## ¿Qué aprenderemos en este tema?

- Qué hace el Sistema Operativo para ejecutar
  - Un programa
  - Varios programas a la vez

} **Procesos, hilos,  
Multiprogramación  
Tiempo Compartido...**
- Cómo lo hace
  - Algoritmos
  - Estructuras de datos

} **Ciclo de vida  
BCP, Tabla de Procesos...**

1. **Conceptos básicos de gestión de procesos**
2. Elementos de un proceso
3. Ciclo de vida de un proceso
4. Hilos o procesos ligeros
5. Planificación de procesos e hilos

# *1.- Conceptos básicos de gestión de procesos*

## *El concepto de proceso*

- Un proceso es **cualquier programa en ejecución**.
- Un proceso **necesita** ciertos **RECURSOS** para realizar satisfactoriamente su tarea:
  - Tiempo de CPU.
  - Memoria.
  - Archivos.
  - Dispositivos de E/S.
- Los recursos se asignan a un proceso:
  - Cuando se crea.
  - Durante su ejecución.

# *1.- Conceptos básicos de gestión de procesos*

## *El concepto de proceso*

- Un proceso **es algo vivo**. En un instante un proceso viene dado por:
  - su código
  - los datos que tiene en ese instante
  - el valor de la pila de ejecución
  - el valor de todos los registros del procesador (incluyendo el contador de programa con la siguiente instrucción a ejecutar)
- Toda esta información constituye el ***estado*** del proceso.

# *1.- Conceptos básicos de gestión de procesos*

## *Funciones del gestor de procesos*

- Las funciones del SO como gestor de procesos son:
  - **Creación y eliminación** de procesos.
  - **Planificación de procesos** (procurando la ejecución de múltiples procesos maximizando la utilización del procesador).
  - Establecimiento de mecanismos para la **sincronización y comunicación** de procesos.
  - Manejo de **bloqueos mutuos**.
- Y siempre tratando de minimizar la **sobrecarga** del SO

# *1.- Conceptos básicos de gestión de procesos*

## *Multitarea*

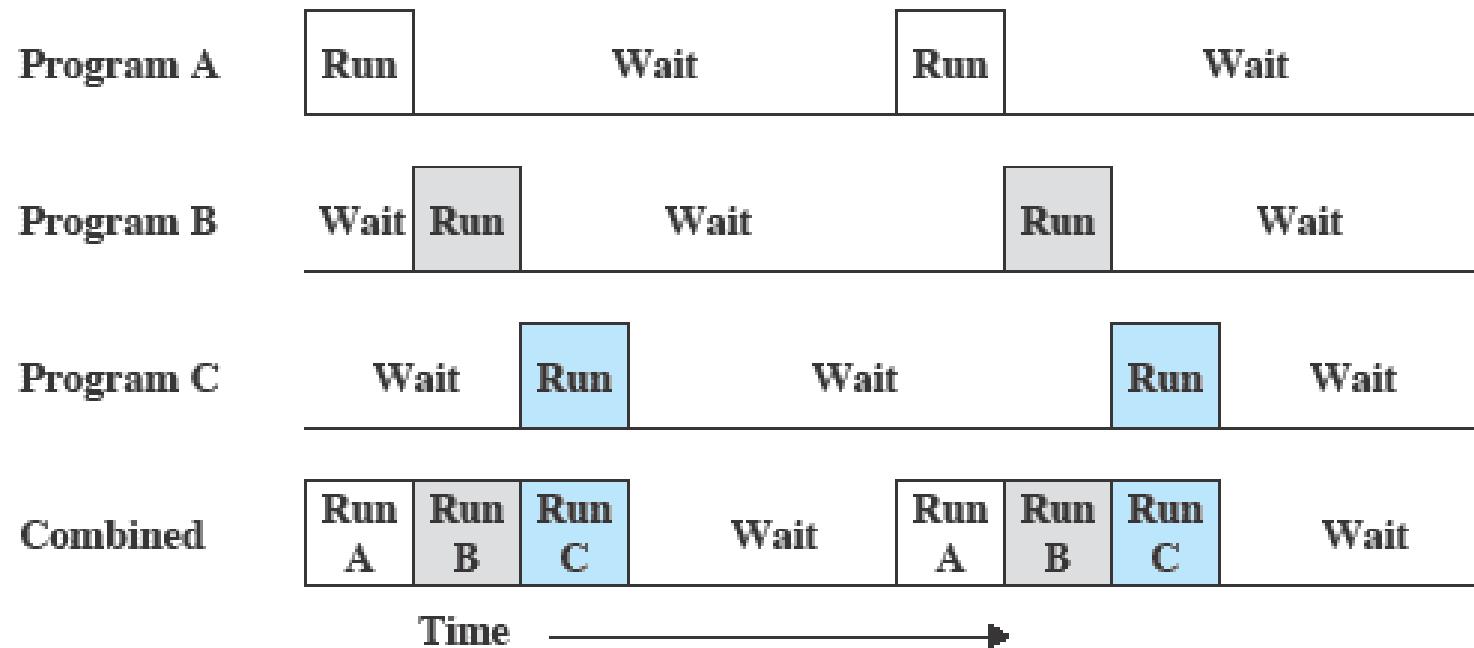
- **¿Cómo se ejecutan varios procesos simultáneamente en una máquina de un único procesador?**

*1. Paralelismo entre procesador y E/S  
(alternancia en el proceso entre cómputo y E/S)*

*2. Reparto del tiempo de procesador entre los procesos*

# 1.- Conceptos básicos de gestión de procesos

## Multitarea



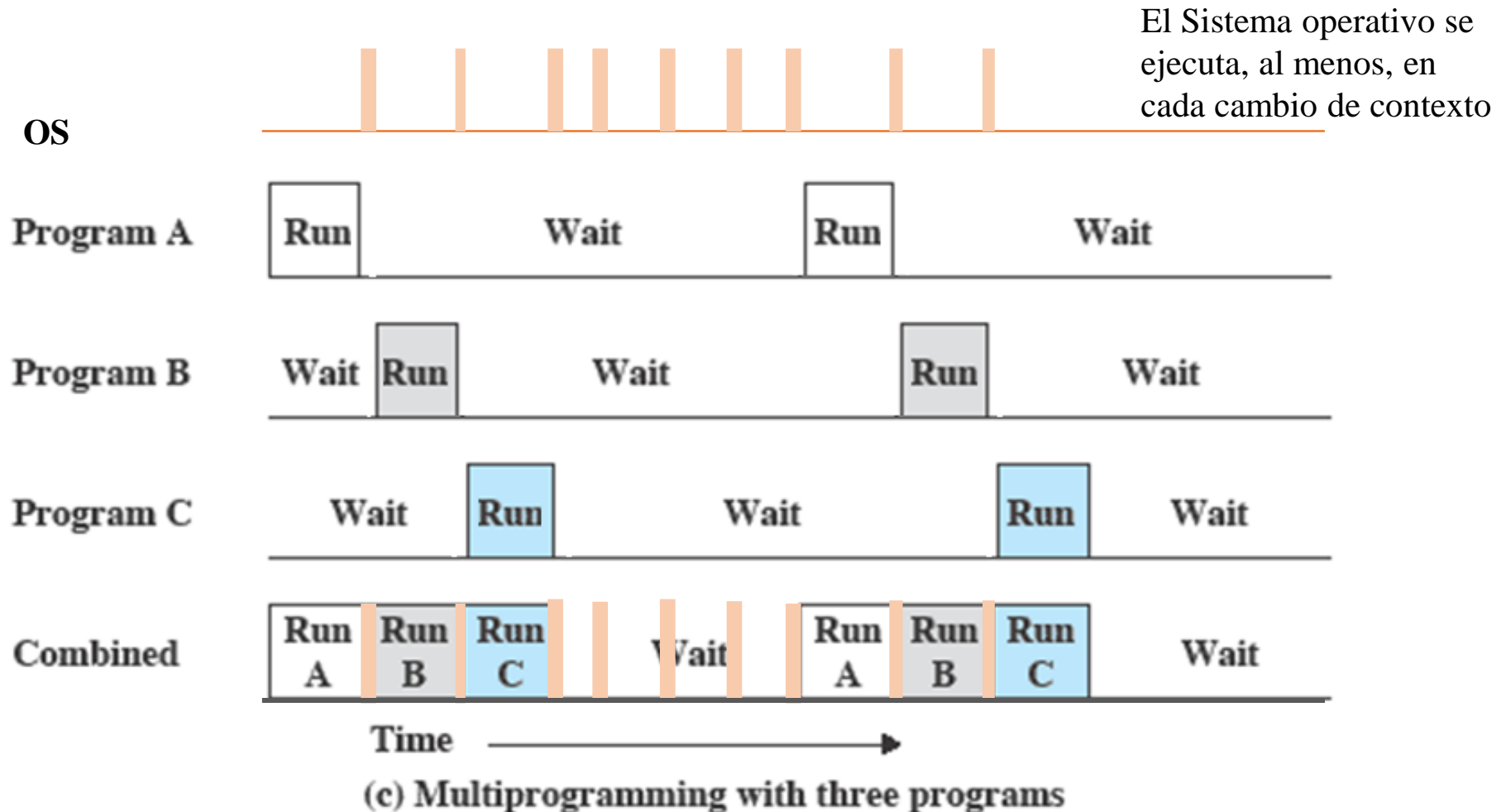
(c) Multiprogramming with three programs

*Es una situación realista?*



# 1.- Conceptos básicos de gestión de procesos

## Multitarea



1. Conceptos básicos de gestión de procesos
- 2. Elementos de un proceso**
3. Ciclo de vida de un proceso
4. Hilos o procesos ligeros
5. Planificación de procesos e hilos

## 2.- Elementos de un proceso

¿Qué información acerca de un proceso necesita manejar el SO para ejecutarlo en multitarea?

- Un proceso es:
  - **Sección de texto** (código del programa)
  - **Sección de datos**, que contiene **variables globales y memoria dinámica**
  - **Pila** (stack), que contiene **datos temporales** (parámetros de subrutinas, direcciones de retorno y variables locales)
  - **Situación actual del procesador:**
    - Valor del **contador de programa**
    - Valor de la **palabra de estado del procesador**
    - Contenido del resto de **registros** del procesador.
  - Información de otros recursos
    - Ficheros abiertos, dispositivos e/s asociados, ...
  - Información de control:
    - Identificadores, Prioridad, estado, ...

## 2.- Elementos de un proceso

### *El bloque de Control de Procesos*

#### EL PCB (Bloque de control de procesos)

- Contiene los **elementos** de un proceso.
- Hace **posible interrumpir** un proceso en ejecución y más tarde retomar la ejecución como si no hubiera ocurrido.
- Creado y gestionado por el Sistema Operativo.
- Herramienta clave para el **soporte de múltiples procesos**.

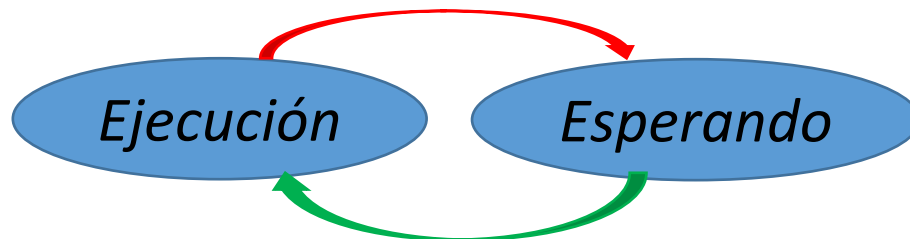
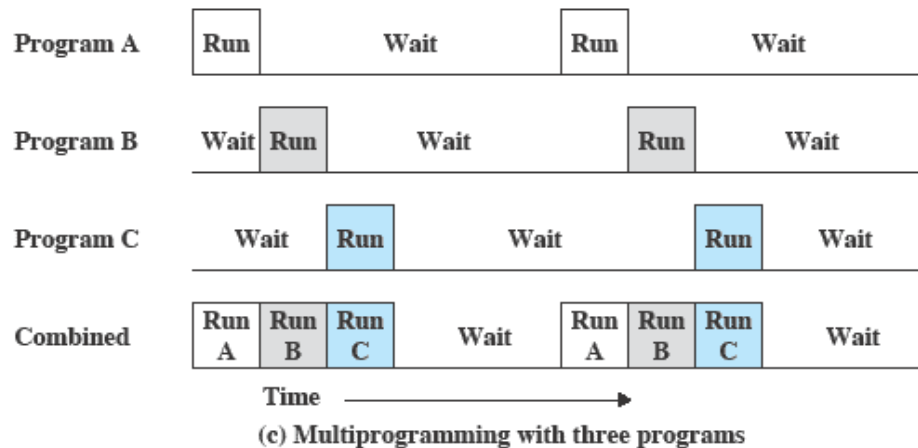
PCB
Identificación
Estado
Prioridad
Copia PC
Copia PSW
Copia otros registros
Punteros a memoria
Información E/S
Información contabilidad
...

# *Contenido*

1. Conceptos básicos de gestión de procesos
2. Elementos de un proceso
- 3. Ciclo de vida de un proceso**
4. Hilos
5. Planificación de procesos e hilos

### 3.- Ciclo de vida de un proceso

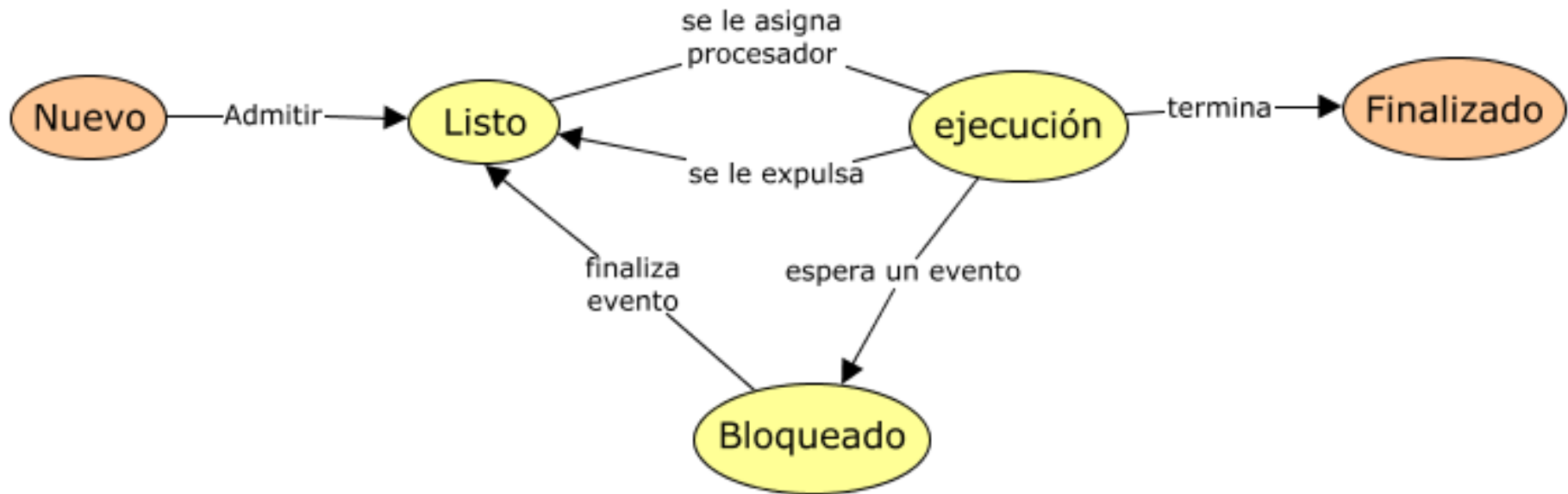
¿En qué estados puede estar un proceso?



### 3.- Ciclo de vida de un proceso

*Modelo de 5 estados*

¿Cuáles son las situaciones que implican un cambio de estado?



## 3.- Ciclo de vida de un proceso

### *Modelo de 5 estados*

- A medida que un **proceso se ejecuta cambia de estado**. Cada proceso puede estar en uno de los estados:
  - **Nuevo** (new): el proceso se está creando.
  - **En ejecución** (running): el proceso está en la CPU ejecutando instrucciones.
  - **Bloqueado** (waiting, en espera): proceso esperando a que ocurra un suceso (ej. terminación de E/S o recepción de una señal).
  - **Listo** (ready, Preparado): esperando que se le asigne a un procesador.
  - **Terminado** (terminated): finalizó su ejecución, por tanto no ejecuta más instrucciones y el SO le retirará los recursos que consume.
- Nota: **Sólo un proceso puede estar ejecutándose en un procesador en un instante dado**, pero muchos procesos pueden estar listos y esperando.



## 3.- Ciclo de vida de un proceso

### *Modelo de 5 estados*

- Los sucesos que pueden dar lugar a una **transición de estados** en este modelo son los siguientes:
  - **Ninguno a nuevo:** se crea un nuevo proceso para ejecutar un programa
  - **Nuevo a preparado:** el sistema está preparado para aceptar un proceso más porque dispone de recursos para ello.
  - **Preparado a ejecución:** el sistema elige uno de los procesos en estado preparado para llevarlo a ejecución.
  - **Ejecución a terminado:** el proceso que se está ejecutando es finalizado por el SO si indica que terminó, se abandona o se cancela.
  - **Ejecución a preparado:** el proceso ha agotado su tiempo de ejecución, cede voluntariamente su tiempo de ejecución o se interrumpe para atender a otro de mayor prioridad.

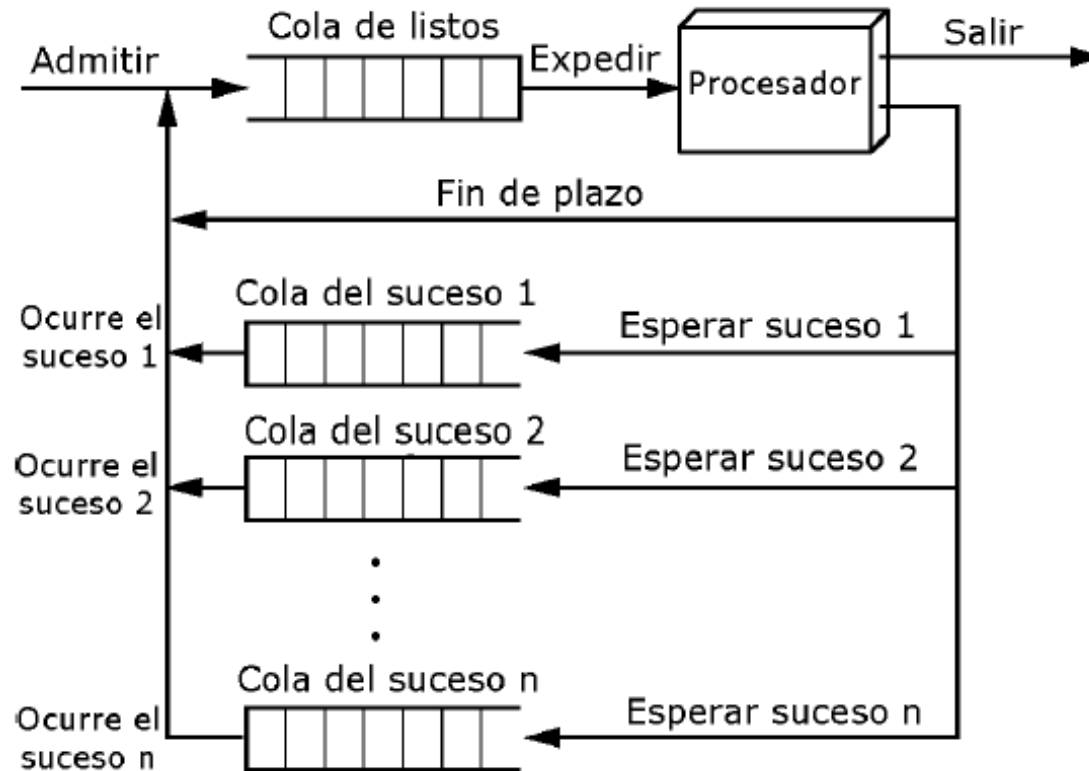
### *3.- Ciclo de vida de un proceso*

#### *Modelo de 5 estados*

- Los sucesos que pueden dar lugar a una transición de estados en este modelo son los siguientes (continuación):
  - **Ejecución a bloqueado:** el proceso solicita algo por lo que debe esperar.
  - **Bloqueado a preparado:** se produce el suceso por el que el proceso estaba esperando.
  - **Preparado a terminado** (no aparece en la figura): un padre puede terminar con un proceso hijo en cualquier momento, o bien, si el padre termina todos sus hijos se pueden terminar.
  - **Bloqueado a terminado:** el mismo criterio que el anterior.

### 3.- Ciclo de vida de un proceso Implementación

El sistema operativo mantiene una serie de colas (no necesariamente FIFO) para saber qué procesos hay en cada estado



### *3.- Ciclo de vida de un proceso*

#### *Gestión de interrupciones*

#### **Cuestiones sobre la gestión de los procesos del sistema**

- 1. ¿Qué ocurre cuando se produce una interrupción?**
2. ¿Qué ocurre si la interrupción implica un cambio de proceso?
3. ¿Qué tipo de interrupciones provocan necesariamente un cambio de proceso en ejecución y cuáles no?
4. ¿Qué implica la creación de un proceso?

### 3- Process Lifecycle

#### OS interrupt handling

Hemos visto en el tema 1 cómo el hardware guarda el Contador de Programa y registro de estado y luego se salta al código encargado de manejar la interrupción.

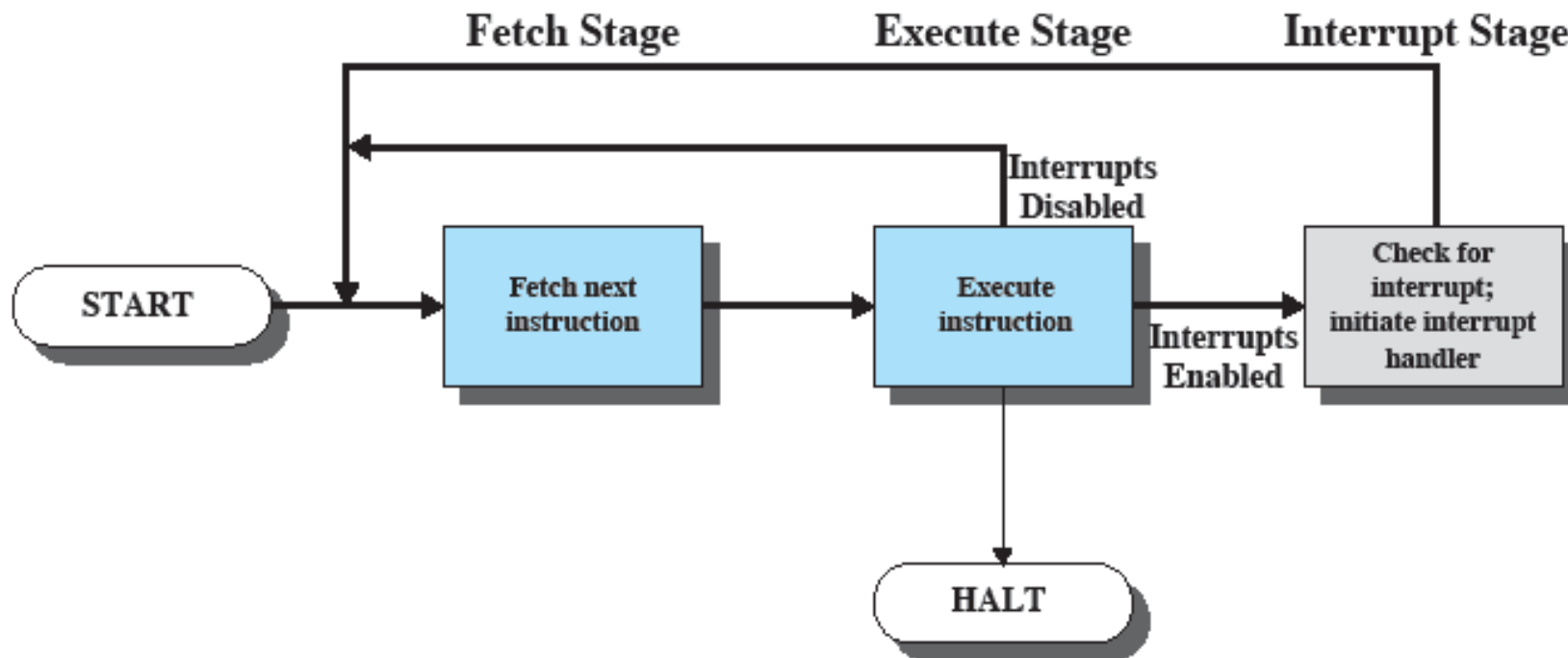


Figure 1.7 Instruction Cycle with Interrupts

### *3.- Ciclo de vida de un proceso*

#### *Gestión de interrupciones*

#### **Cuestiones sobre la gestión de los procesos del sistema**

1. ¿Qué ocurre cuando se produce una interrupción?
- 2. ¿Qué ocurre si la interrupción implica un cambio de proceso?**
3. ¿Qué tipo de interrupciones provocan necesariamente un cambio de proceso en ejecución y cuáles no?
4. ¿Qué implica la creación de un proceso?

### *3.- Ciclo de vida de un proceso*

#### *Gestión de interrupciones*

Tras ejecutar lo necesario para tratar la interrupción en cuestión, antes del IRET:

1. Se **planifica** (decide) que proceso pasará a ejecutarse (de entre los que estén listos)
2. Se realiza un **cambio de contexto**
  1. Se guardan los datos de la pila de control (que contienen los valores del PC y PSW), así como el resto de registros del procesador en el BCP del proceso interrumpido.
  2. Se cargan los valores de los registros del procesador del nuevo proceso desde el PCB al procesador. El PC y PSW se copiarán a la pila de control.

El IRET de la interrupción restaura estos valores a los registros

### 3.- Ciclo de vida de un proceso

#### *Cambio de contexto*

#### **Cambio de contexto**

*Operaciones realizadas por el SO para hacer que un proceso libere el procesador y otro obtenga el procesador para su ejecución.*

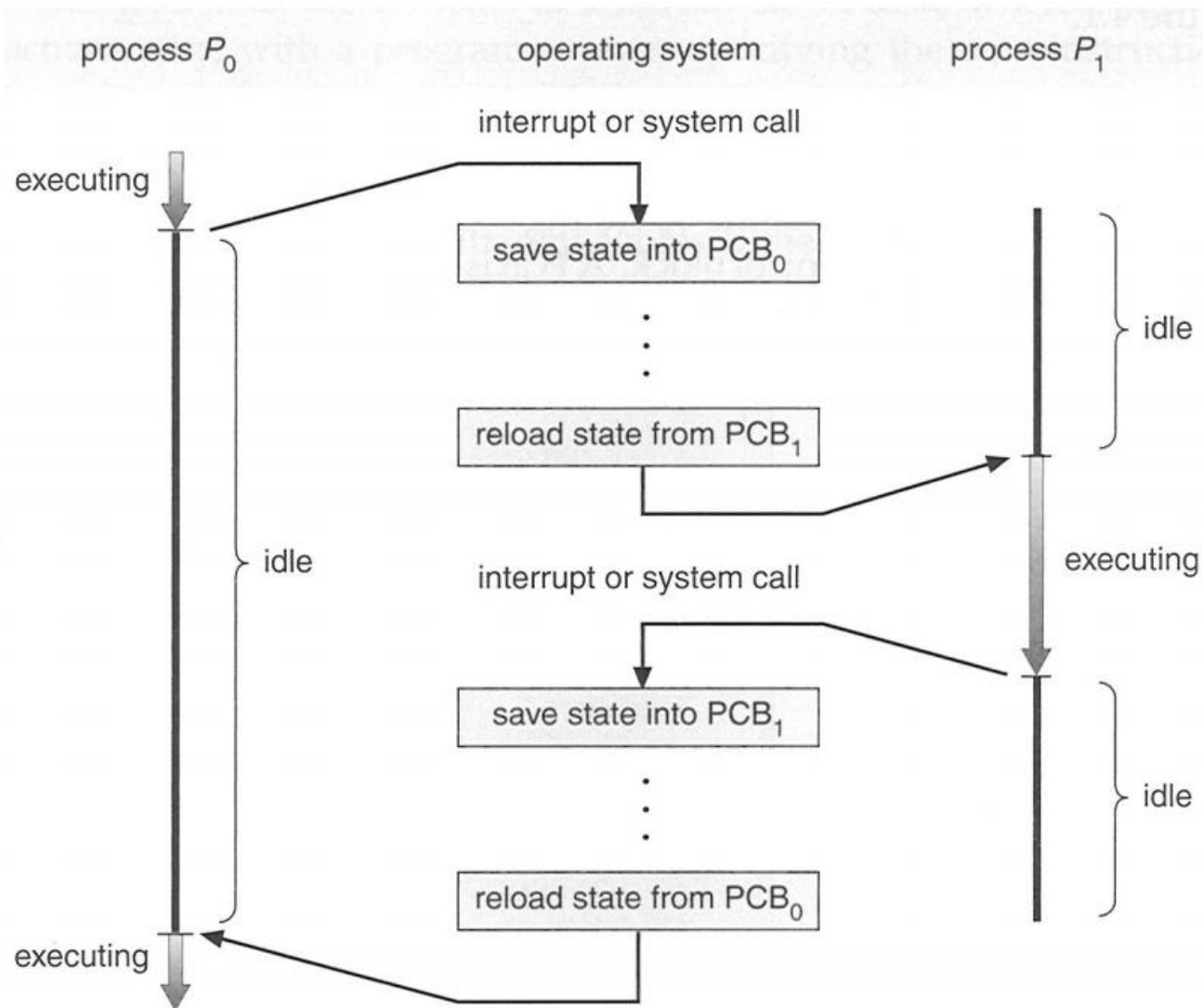
Implica

- Salvar estado del procesador en PCB del proceso que abandona la CPU.
- Restaurar el estado del procesador a partir del PCB del proceso al que se le asigna el procesador.



### 3.- Ciclo de vida de un proceso

#### Cambio de contexto



### *3.- Ciclo de vida de un proceso*

#### *Gestión de interrupciones*

#### **Cuestiones sobre la gestión de los procesos del sistema**

1. ¿Qué ocurre cuando se produce una interrupción?
2. ¿Qué ocurre si la interrupción implica un cambio de proceso?
- 3. ¿Qué tipo de interrupciones provocan necesariamente un cambio de proceso en ejecución y cuáles no?**
4. ¿Qué implica la creación de un proceso?

### 3.- Ciclo de vida de un proceso

#### *Causas del cambio de proceso*

#### **Algunas interrupciones causan el cambio de proceso**

- El proceso actual no puede continuar porque:
  - Termina.
  - Se bloquea esperando por algo (el fin de una operación lenta, por una señal de otro proceso, ...)
- El proceso actual puede continuar, pero el sistema decide que ya se ha ejecutado demasiado tiempo (*time-out*).

#### **Otras interrupciones no causan el cambio de proceso**

- El proceso actual puede continuar y el sistema decide que todavía no se ha ejecutado demasiado tiempo.

### *3.- Ciclo de vida de un proceso*

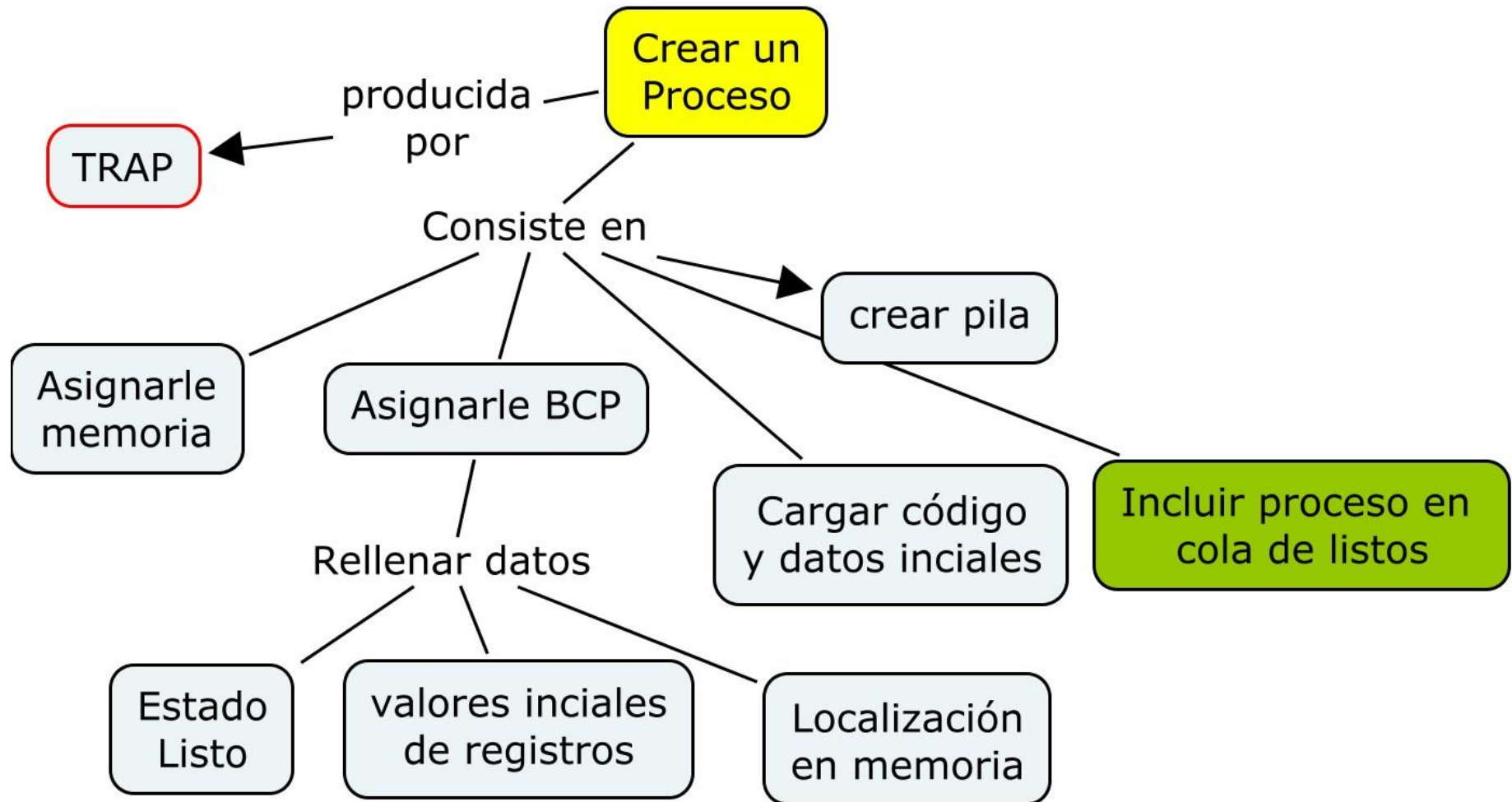
#### *Gestión de interrupciones*

#### **Cuestiones sobre la gestión de los procesos del sistema**

1. ¿Qué ocurre cuando se produce una interrupción?
2. ¿Qué ocurre si la interrupción implica un cambio de proceso?
3. ¿Qué tipo de interrupciones provocan necesariamente un cambio de proceso en ejecución y cuáles no?
4. **¿Qué implica la creación de un proceso?**

### 3.- Ciclo de vida de un proceso

#### Creación de procesos



# *Contenido*

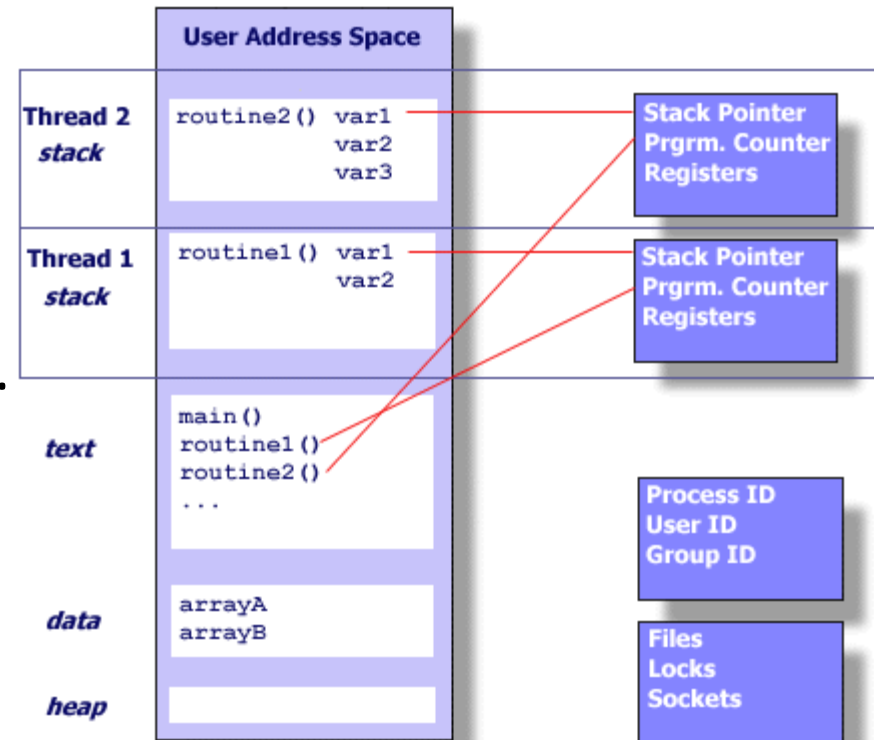
1. Conceptos básicos de gestión de procesos
2. Elementos de un proceso
3. Ciclo de vida de un proceso
- 4. Hilos**
5. Planificación de procesos e hilos
6. Servicios del sistema para gestión de procesos e hilos

## 4.- Hilos

- **Hilo (thread, hebra): Flujo de ejecución dentro de un proceso**

Varios hilos en la misma tarea

- Cada hilo dispone de:
  - Contador de Programa.
  - Valores de registros del procesador.
  - Pila.
- Los hilos comparten
  - Imagen en memoria, archivos abiertos, señales, etc.



# 4.- Hilos

*Aplicaciones... ¿Por ejemplo?*

## Aplicaciones

- Programación paralela (separación tareas, modularidad)
- Servidores de información

*¿Ventajas... Por qué?*

## Ventajas

- Facilidad de comunicación entre hilos.
- Se incrementa la velocidad de ejecución del trabajo
  - El cambio entre hilos de la misma tarea tiene poco coste.
  - El coste de creación y destrucción de hilos es mucho menor que el de procesos
- Se mejora por tanto el rendimiento.

## Inconvenientes

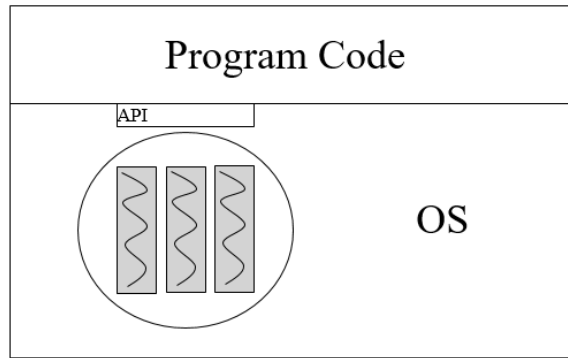
- Compartición del espacio de memoria (acceso concurrente a recursos compartidos)
- Mayor dificultad de programación

## Implementación

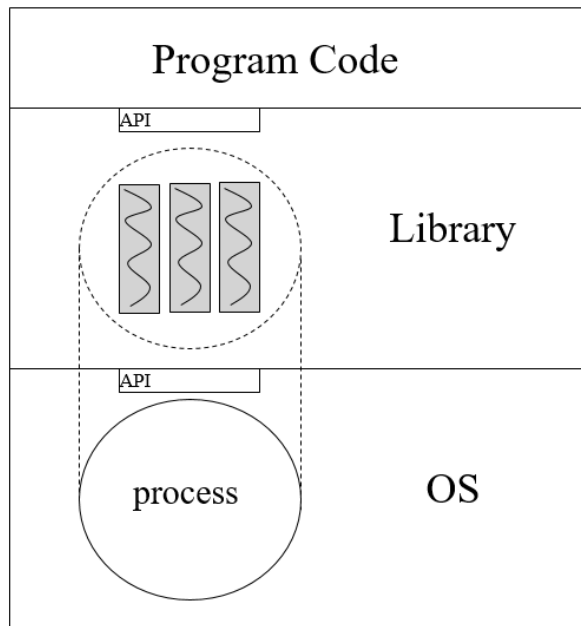
- A nivel de núcleo (KLT)
- A nivel de usuario (ULT)
- Combinación de ambos



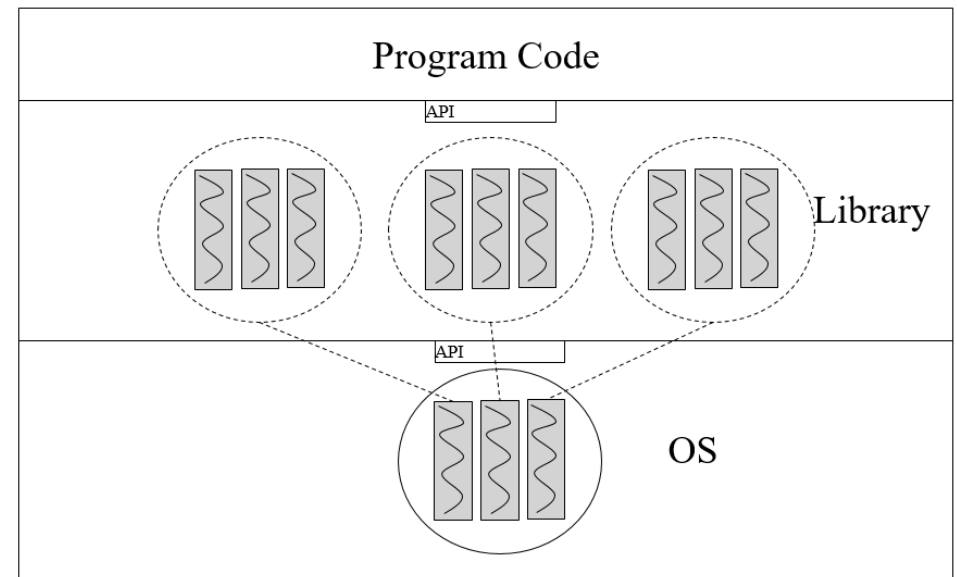
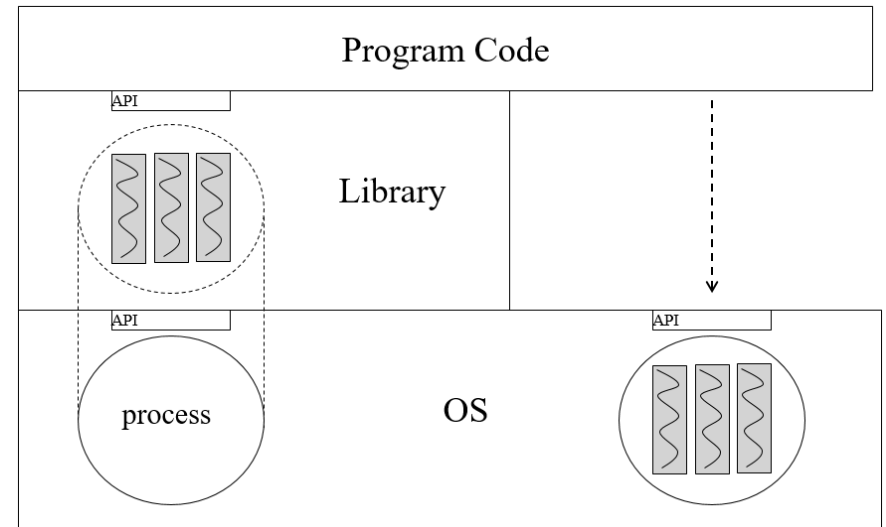
# 4- Hilos KLT y ULT



*KLT Implementation*



*ULT Implementation*



## 4.- Hilos

### Combinación ULT y KLT

	Ventajas	Desventajas
ULT (Hilos a nivel de usuario )	<ul style="list-style-type: none"><li>• Más políticas disponibles</li><li>• Gestión de operaciones más eficiente (creación, cambio de contexto, bloqueo)</li><li>• No se necesita al núcleo para sincronizar hilos.</li></ul>	<ul style="list-style-type: none"><li>• El bloqueo de un hilo bloquea al resto.</li><li>• No hay paralelismo real</li></ul>
KLT (Hilos a nivel de núcleo)	<ul style="list-style-type: none"><li>• El bloqueo de un hilo no bloquea al resto.</li><li>• Es posible el paralelismo real</li></ul>	<ul style="list-style-type: none"><li>• Gestión de operaciones menos eficiente</li><li>• Necesario cambio de modo para cambiar entre hilos</li></ul>

#### 1 / 10 / 100 relación

- *Creación de un hilo en ULT: 1 unidad de tiempo*
- *Creación de un hilo en KLT : 10 unidades de tiempo*
- *Creación de proceso: 100 unidades de tiempo*

# *Contenido*

1. Conceptos básicos de gestión de procesos
2. Elementos de un proceso
3. Ciclo de vida de un proceso
4. Hilos
- 5. Planificación de procesos e hilos**

## 5.- Planificación de procesos e hilos

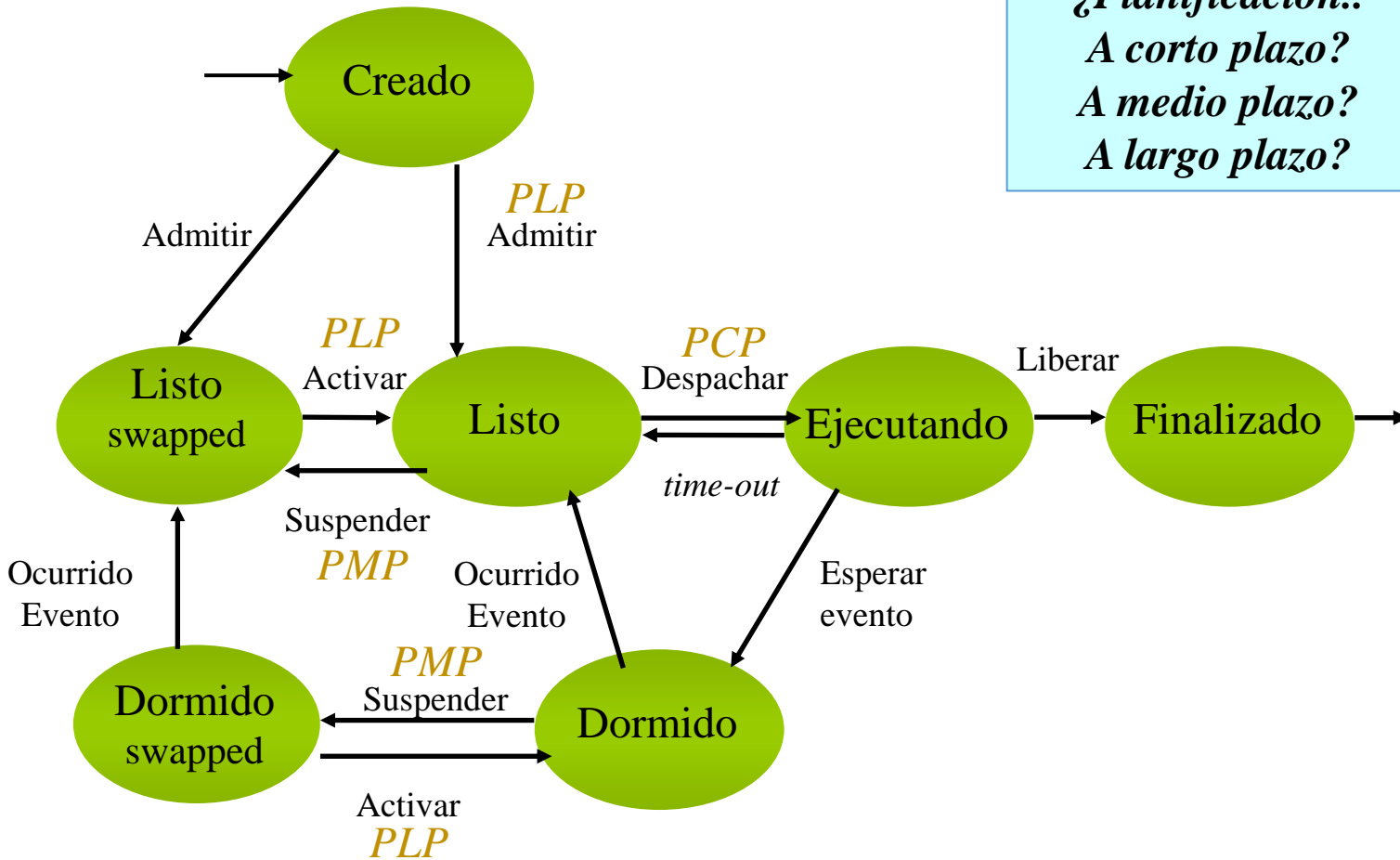
**Planificación:** *Reparto del tiempo del procesador entre los procesos que se pueden ejecutar.*

- **Niveles de planificación (clásicamente)**
  - **Corto plazo** asigna el procesador
  - **Medio plazo** trata la suspensión de procesos a memoria secundaria (modelo de 7 estados). Reduce el ***grado de multiprogramación*** del sistema
  - **Largo plazo** añade procesos al sistema (creados o *expulsados*)
- **Planificador:** Módulo que decide qué proceso pasa de listo a ejecutando. (Planificador a corto plazo)
- **Activador** (despachador) módulo que pone en ejecución el proceso planificado.

# 5.- Planificación de proceso e hilos

## Modelo de 7 estados

¿Planificación..  
A corto plazo?  
A medio plazo?  
A largo plazo?



## 5.- Planificación de proceso e hilos

### Objetivos de la planificación

## Métricas de rendimiento de algoritmos de planificación

### Proceso (o thread)

- Tiempo de ejecución o retorno (*turnaround* time) (Tt)
- Tiempo de espera (*waiting* time) (Tw)
- Tiempo de respuesta (*response* time) (Tr)

### Objetivo general

- Minimizar métricas de Proceso
- Maximizar métricas de sistema

### Sistema

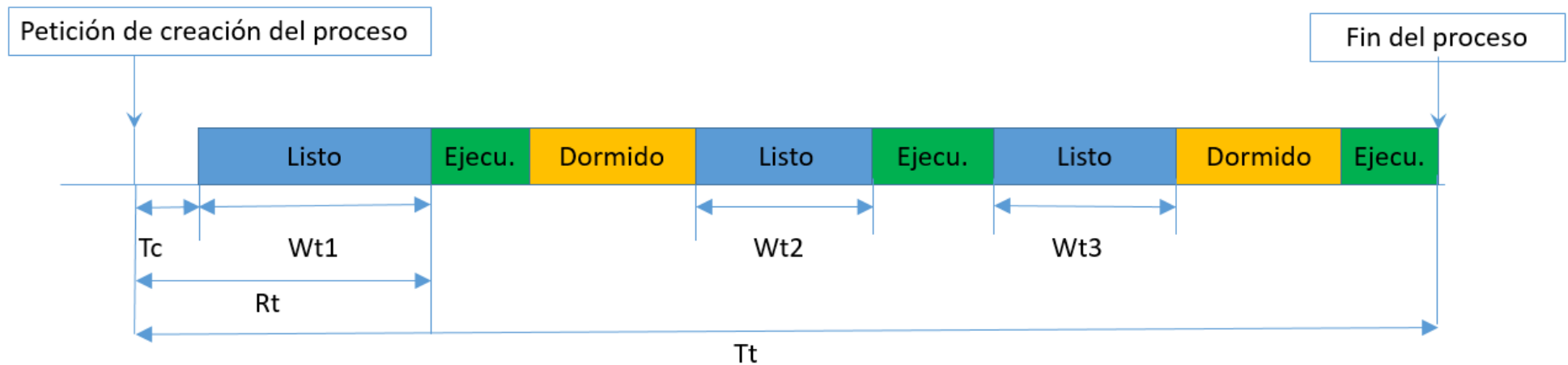
- Uso del procesador (C)
- Tasa de trabajos completados (P)

### Otros objetivos...

- Imparcialidad
- Reparto equitativo de procesador
- Eficiencia de procesador (maximizar uso)
- Predictibilidad de ejecución (tiempo real)
- Equilibrio en uso de recursos
- Minimizar varianza en tiempo de respuesta
- Reducir tiempo de cambio entre procesos

# 5.- Planificación de proceso e hilos

## Objetivos de la planificación



- $Tc$ : Tiempo usado por el SO. para crear el proceso
- $Tr$ : Response Time (tiempo de respuesta)
- $Tw$ : Waiting time.  $Wt = Wt1 + Wt2 + Wt3$  (tiempo de espera)
- $Tt$ : Turnaround Time (tiempo de retorno)

## *5.- Planificación de proceso e hilos*

### *Objetivos de la planificación*

- El cambio de contexto se considera como la operación más costosa en un sistema operativo.
- Por lo tanto, un objetivo principal en el diseño del sistema operativo es evitar los cambios de contexto todo lo posible.



## 5.- Planificación de proceso e hilos

### Políticas de planificación

- **No Expulsivas**
  - FCFS (FIFO)
  - SJF (Primero el más corto)
- **Expulsivas**
  - SRTF (Menor tiempo Restante)
  - Turno Rotatorio (RR)
  - Prioridades
    - Estáticas
    - Dinámicas
      - Con envejecimiento
      - Favorecer procesos interactivos

*¿Ventajas e Inconvenientes?*

*¿Cuál elegirías para tu diseño? ¿Por qué?*

*Políticas actuales: combinación de políticas expulsivas*

*Colas multinivel con Prioridades dinámicas + Turno rotatorio*

## *5.- Planificación de proceso e hilos*

### *Política de planificación FCFS o FIFO no expulsivo*

- Los procesos listos para ejecución están organizados en una cola FIFO
  - Cuando un proceso pasa a listo para ejecución se coloca al final de la cola.
- Los procesos abandonan voluntariamente la CPU
  - El proceso realiza una llamada al sistema y como consecuencia de ella se duerme.
  - El proceso termina.
- Cuando la CPU queda libre el planificador elige al proceso que está en la cabeza de la cola de listos.

## *5.- Planificación de proceso e hilos*

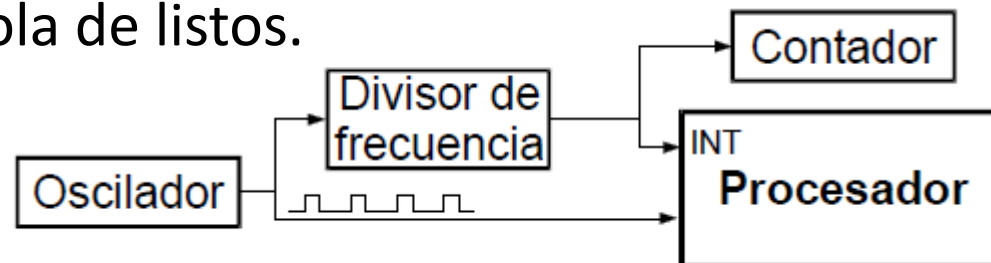
### *Política de planificación SJF (Primero el más corto)*

- Se elige el **proceso con menor duración total**. Favorece a los trabajos cortos frente a los largos.
- **Exige conocer el tiempo de ejecución antes** de ejecutar el trabajo. Sólo útil en ambientes de producción donde se ejecutan frecuentemente los mismos trabajos.
- Política no expulsiva.
- Puede provocar inanición de los procesos largos.
- Mejora la media de los tiempos de respuesta, pero aumenta la varianza.
- Política en desuso. Diseñada para los sistema de procesamiento por lotes anteriores a la multiprogramación.

## 5.- Planificación de proceso e hilos

### Política de planificación Cíclica o Turno Rotatorio

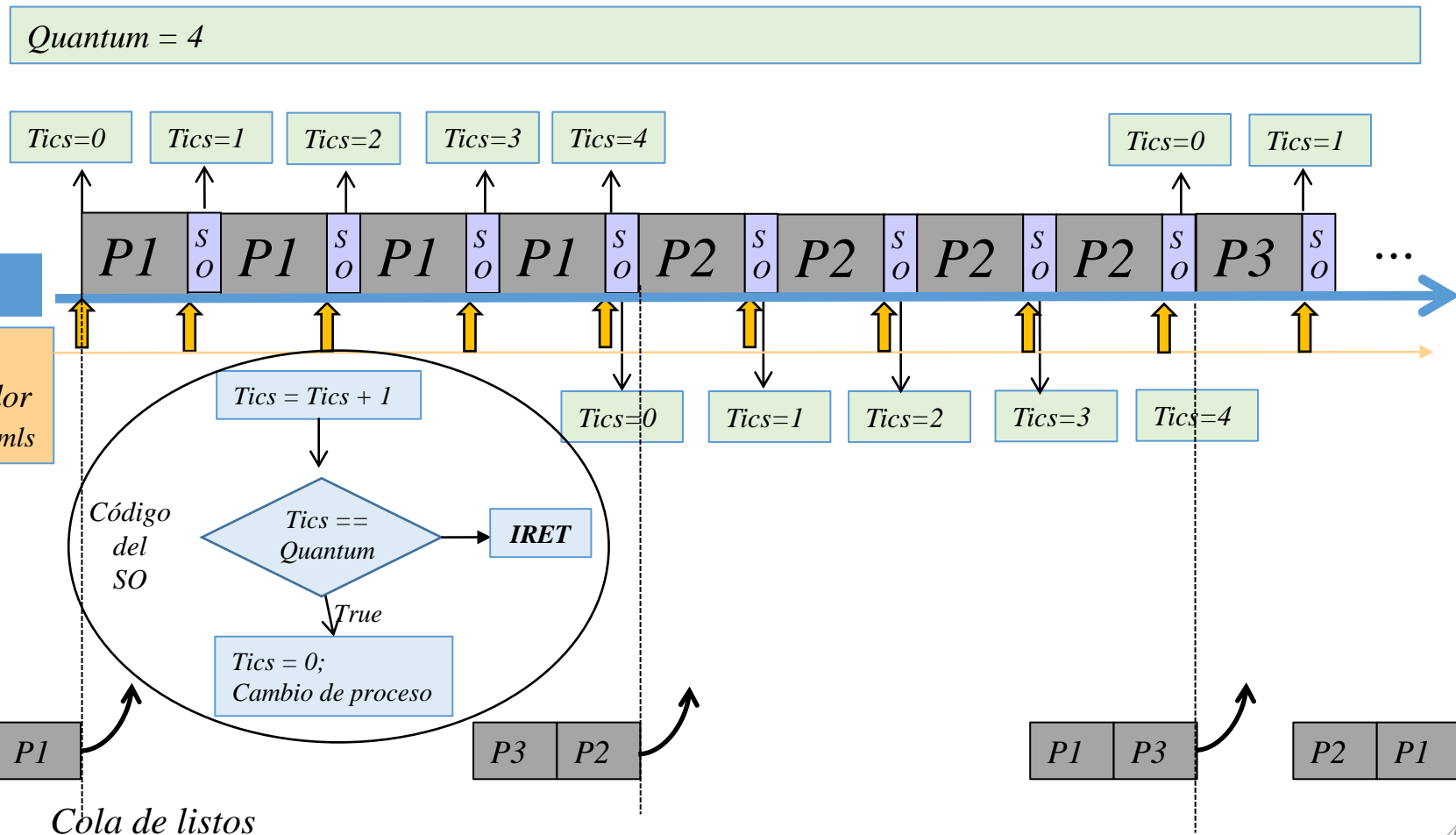
- Consigue un reparto equitativo del tiempo del procesador
- El sistema asigna un *cuanto* o rodaja al proceso que ha entrado en la CPU.
- Los procesos listos para ejecución están organizados en una cola FIFO
  - Cuando un proceso pasa a listo para ejecución se coloca al final de la cola.
- Los procesos abandonan la CPU porque:
  - El proceso realiza una llamada al sistema y como consecuencia de ella se duerme.
  - El proceso termina.
  - El proceso agota su cuanto.
- Cuando la CPU queda libre el planificador elige al proceso que está en la cabeza de la cola de listos.



# 5.- Planificación de proceso e hilos

## Política de planificación Cíclica o Turno Rotatorio

*Expulsión por fin de cuanto de tiempo*



## *5.- Planificación de proceso e hilos*

### *Política de planificación basada en prioridades*

- Cada proceso tiene asignada una prioridad:
  - Asignada automáticamente por el sistema.
  - Seleccionada por el usuario.
- La cola de listos está ordenada por prioridad. El planificador elige al proceso de mayor prioridad.

Tipos de prioridad:

- **Estática:** fija durante toda la vida del proceso.
  - Fácil de implementar.
  - Introduce poca sobrecarga en el sistema.
  - No se adecua a cambios en el entorno.
  - Puede provocar inanición.

## 5.- Planificación de proceso e hilos

### *Política de planificación basada en prioridades*

- **Dinámica:** varía durante toda la vida del proceso.
  - Más difícil de implementar.
  - Introduce mayor sobrecarga en el sistema.
  - Evita inanición.
- Ejemplo: *prioridad con envejecimiento*. Se asigna una prioridad inicial y se aumenta a medida que pasa el tiempo sin ejecutarse. Cuando se ejecuta se vuelve a la prioridad inicial.
- Ejemplo: *Aumenta* la prioridad cuando el proceso realiza operaciones de e/s y la *disminuye* cuando agota el cuanto.
- Esta política se combina con otras (cíclica, FIFO).

## *5.- Planificación de proceso e hilos*

### *Políticas de planificación de sistemas en tiempo real*

- Los procesos se tienen que ejecutar en **instante predeterminados**.
- Hay dos tipos de procesos:
  - **A plazo fijo**: se ejecutan una vez en un instante determinado.
  - **Periódico**: se ejecutan repetidamente cada cierto tiempo.
- Se asocia a cada proceso el instante en que se debe ejecutar.
- Los procesos que no han alcanzado su hora de ejecución están en una cola de espera; los que han alcanzado su hora de ejecución pasan a la cola de listos.
- En los sistemas críticos se asigna a cada proceso una franja horaria de ejecución; no se debe rebasar en ningún caso la hora límite de ejecución.



## 5.- Planificación de proceso e hilos

### Política de planificación en POSIX

- Se aplica a procesos e hilos
- Políticas basadas en prioridades. Cada proceso tiene una prioridad (entre 0 y 31).
- Se elige al hilo de mayor prioridad (menor valor numérico)
- Políticas disponibles (usando la llamada *sched\_setscheduler*)
  - FIFO (SCHED FIFO)
  - Round Robin (SCHED RR)
  - Otras (SCHED OTHER)
- **Conviven todas:** en el sistema cada proceso elige la política, por lo que existen las tres a la vez

## *5.- Planificación de proceso e hilos*

### *Política de planificación en POSIX*

- Planificación FIFO
  - Una cola FIFO por cada prioridad
  - Se expulsa al proceso si llega otro con mayor prioridad o si hay E/S.
  - Si llega otro con mayor prioridad el expulsado pasa a ser el primero de la cola asociada a su prioridad.
  - Un proceso dormido que pasa a listo se introduce al final de la cola de su prioridad.

## *5.- Planificación de proceso e hilos*

### *Política de planificación en POSIX*

- Planificación Cíclica
  - Una cola FIFO por cada prioridad, y a los procesos se les asigna una rodaja de tiempo.
  - Cuando un proceso acaba su rodaja, se le inserta al final de su cola.
  - Cuando un proceso es expulsado por otro de mayor prioridad, se introduce al principio de la cola sin inicializar su rodaja de tiempo.
- Otra política
  - Además de Fifo y cíclica, el estándar establece que se puede implementar otra a elegir.

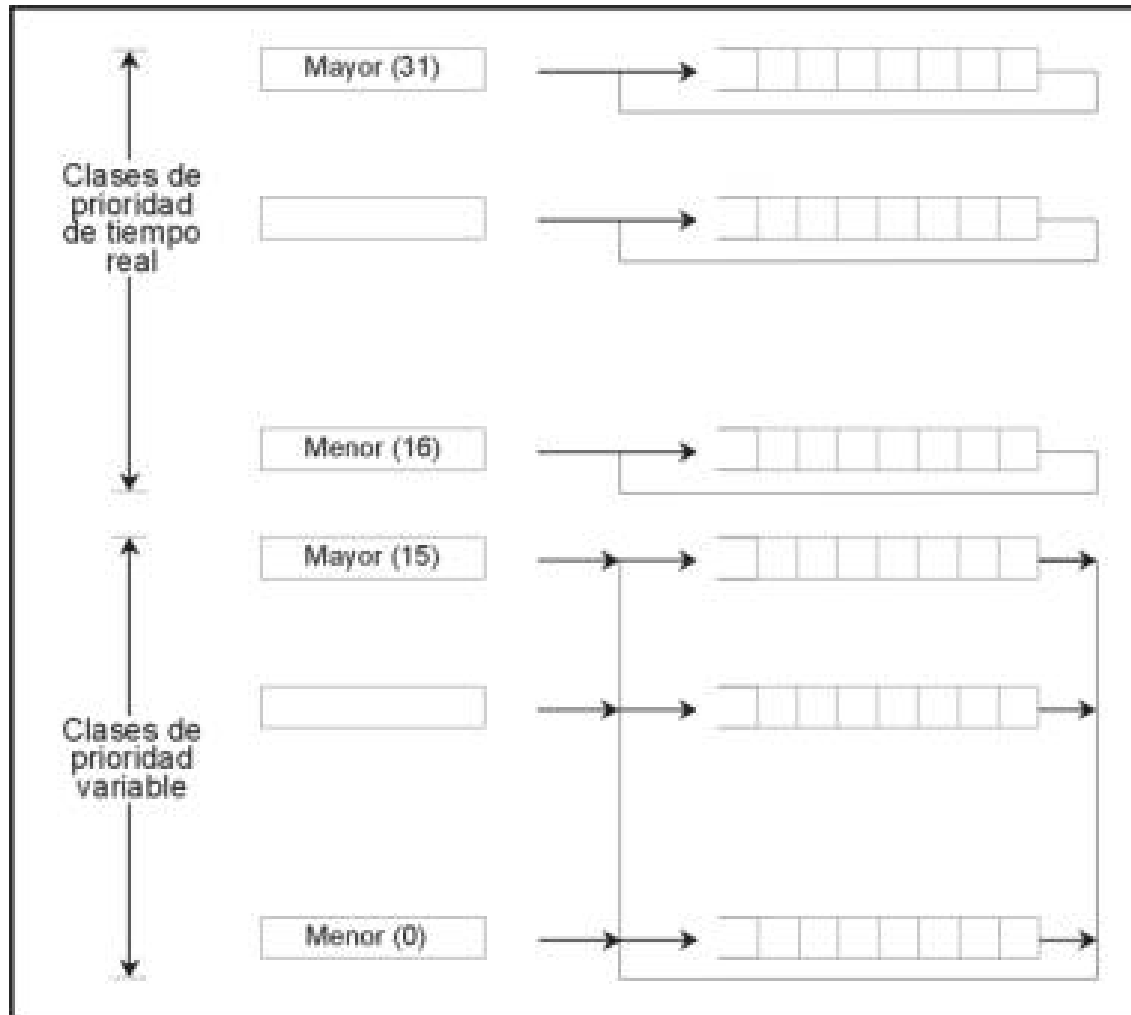
## *5.- Planificación de proceso e hilos*

### *Política de planificación en Win32*

- Planificación cíclica con prioridades y con expulsión. Se planifican hilos
- 32 niveles de planificación (de 0 a 31 el máximo)
  - 16 niveles con prioridades de tiempo real (16 a 31). Prioridad fija
  - 15 niveles con prioridades variables (1 a 15). Prioridad variable según el comportamiento del hilo. Se decrementa la prioridad si el hilo agota la rodaja de tiempo, se incrementa si realiza una operación de E/S bloqueante.
  - Un nivel de sistema (0)
  - Existe prioridad base para el proceso y para los hilos

## 5.- Planificación de proceso e hilos

### Política de planificación en Win32



## *5.- Planificación de proceso e hilos*

### *Política de planificación en sistemas multiprocesadores*

#### Ejecución del SO en multiprocesadores

- Multiprocesamiento Asimétrico
  - Un procesador dedicado a la ejecución del SO.
  - Ralentiza pero el SO es más simple
    - No hay que gestionar concurrencia de operaciones del SO
- Multiprocesamiento Simétrico
  - El SO se ejecuta en todos los procesadores cuando sea necesario
  - Problemas de concurrencia de servicios al SO
  - Código del SO más complejo
  - Ejecución más eficiente

## *5.- Planificación de proceso e hilos*

### *Política de planificación en sistemas multiprocesadores*

#### Criterios para la planificación

- **Afinidad al procesador**

- **Afinidad.** Proceso listo es afín al procesador en el que se ha ejecutado anteriormente. En siguiente ejecución podrá aprovechar al máximo la información de la caché.

- (llevan una caché por procesador)

- **Afinidad estricta**

- Posibilidad de indicar en un proceso la afinidad a un procesador o grupo (para procesos críticos)

- **Reparto de carga**

- Equilibrio de carga de ejecución de los procesadores.

## *5.- Planificación de proceso e hilos*

### *Política de planificación en sistemas multiprocesadores*

Dos posibilidades de planificación de procesos

- Cola (de listos) única para todos los procesadores
- 1 cola por procesador



## *5.- Planificación de proceso e hilos*

### *Política de planificación en sistemas multiprocesadores*

#### **Planificación con cola única**

- Si procesador libre
  - Elige proceso de la cola
    - Por prioridad y
    - Por Afinidad (es afín si se ha ejecutado en ese procesador la última vez)
- NO hay procesador libre
  - Si llega proceso a listos (cuando nace o sale de bloqueado) pasa a procesador si algún procesador ejecuta proceso menos prioritario (lo expulsa)
    - Tiene en cuenta afinidad

#### **Ventaja**

- Equilibrio de carga

#### **Inconveniente**

- Acceso protegido a cola implica cuello de botella

## *5.- Planificación de proceso e hilos*

### *Política de planificación en sistemas multiprocesadores*

#### **Planificación con 1 cola por procesador**

- Cada proceso se asigna a un procesador
  - Cada procesador coge de su cola
- 
- Ventaja
    - Desaparece el cuello de botella
  - Inconveniente
    - Reequilibrio de carga no es automático
      - Periódicamente
      - Si procesador queda libre
    - Se cambian procesos de una cola a otra.
    - Se tiene en cuenta afinidades (estricta y de caché)

## *5.- Planificación de proceso e hilos*

### *Política de planificación en sistemas multiprocesadores*

#### Políticas de planificación de hilos

1. No se tienen en cuenta relaciones entre hilos del mismo proceso
2. Se tienen en cuenta las relaciones entre hilos

##### **2.1Tiempo compartido**

- Aplicaciones se multiplexan en el tiempo
  - Se asigna un tiempo a cada aplicación (conjunto de sus hilos) en un procesador. Todos los hilos se ejecutan en el mismo procesador.
  - Se benefician de la afinidad

##### **2.2 Espacio compartido (co-planificación)**

- Los hilos de una aplicación se reparten entre grupos de procesadores
- Se benefician del paralelismo real

En cualquier tipo se puede emplear cola global o colas locales

## *5.- Planificación de proceso e hilos*

### *Política de planificación en sistemas multiprocesadores*

#### **Sistemas actuales**

- Multiprocesamiento simétrico
  - El SO se ejecuta en todos los procesadores
- Planificación
  - Windows (cliente)
    - 1 cola global
  - Windows (servidores)
    - Colas locales
  - Linux
    - Colas locales