



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

GASTEIZKO  
INGENIARITZA  
ESKOLA  
ESCUELA  
DE INGENIERÍA  
DE VITORIA-GASTEIZ

# 7 Gestión y planificación de procesos.

Introducción a los Sistemas Operativos,  
2025-2026

Pablo González Nalda

Dept. de Lenguajes y Sistemas Informáticos  
EU de Ingeniería de Vitoria-Gasteiz,  
UPV/EHU



24 de marzo de 2026

# Contenidos de la presentación

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

- 1 Procesos: Motivación
- 2 Control de procesos
- 3 Representación de los procesos
- 4 Llamadas al Sistema
- 5 Planificación de procesos y procesadores
- 6 Políticas de planificación
- 7 Planificación en multiprocesadores
- 8 Planificación de tareas de tiempo real
- 9 ¿Más preguntas?



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

# 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?



# Mecanismos para acelerar programas

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

En un ordenador ejecutamos muchos programas a la vez:

- concurrentemente, si sólo tiene un núcleo/procesador (se van **alternando, intercalando partes** de su ejecución) o
- en paralelo, si se ejecuta cada programa en uno de los núcleos o *cores* del procesador.

La mejora de rendimiento del sistema es notable incluso cuando sólo tenemos un núcleo respecto a ejecutar secuencialmente los programas. ¿Por qué?

Si sólo ejecuta **un programa** el ordenador, la bajada de rendimiento se debe a la diferencia de velocidad entre dispositivos de E/S y procesador.



# Mecanismos para acelerar programas

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

Vamos a analizar las técnicas para acelerar la ejecución en un sistema en el que se ejecuta un solo programa.

**E/S asíncrona** Anticipa la E/S al momento en que se necesita.

- Ordenación especial de las instrucciones.
- Incrementa la complejidad de la programación.
- Necesita sincronización posterior.

**Buffering** utilizar búferes para realizar las operaciones de E/S de manera transparente al programa.

- Gestión, previsión
- Ocupa memoria

**Spooling** uso de almacenamiento intermedio.

- Cuando se utilizan dispositivos de E/S lentos y no de forma interactiva.
- Se usa un dispositivo intermedio más rápido.
- Suele haber un hardware dedicado



# Ventajas e inconvenientes

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Diferentes ventajas, según la técnica que usemos

- solapa E/S con su propia ejecución.
- solapa E/S con la ejecución de otro trabajo.
- es fácil obtener varias copias del trabajo.

## Inconvenientes

- Espacio en disco.
- Tiempo de tratamiento.
- Se sigue sin resolver el problema: Un usuario, en general no tiene ocupada la CPU todo el tiempo.



# Solución

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

**Solución: si tenemos suficiente espacio en memoria cargamos varios programas.**

**Cuando un programa en ejecución necesita esperar por E/S, el SO lo para y pasa a ejecutar otro que pueda ejecutarse.**

# Contexto de ejecución

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

Los programas deben realizar su función independientemente de si se ejecutan solos (monoprogramación) o concurrentemente con otros (multiprogramación).

Para ello se debe crear un *Contexto de Ejecución* donde el SO mantiene por cada programa la información necesaria para ejecutar correctamente el programa.

El SO, cuando pasa de ejecutar un programa a ejecutar otro, realiza lo que se llama *cambio de contexto* (*Context Switch, CS*).

**PROCESO** = Información de contexto + programa(inst. y datos)

**PROCESO:** programa en estado de ejecución.

Información de contexto: Identificador, Tabla de Canales, Estado de los recursos del procesador (registros,...)



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?

# Control de procesos

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

**Flujo de ejecución:** código ejecutable que ocupa la CPU. Se determina con el registro PC (*Program Counter*), que indica dónde está en la RAM la instrucción de Lenguaje Máquina que se está ejecutando.

Un **proceso** es un conjunto de uno o más flujos de ejecución denominados **hilos**, junto al contexto de ejecución.

El **contexto de ejecución** es el conjunto de todos los datos que permiten al SO ejecutar el programa: código y datos en la RAM, pila y montículo, estado de la memoria y de la E/S (datos de gestión relativos al fichero), información del sistema sobre el programa en ejecución (tablas).

# Tipos de hilos

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

*Kernel Level Thread* El SO gestiona los hilos

*User Level Thread* La aplicación gestiona los hilos

Biblioteca Pthreads (*Posix Threads*): biblioteca estándar que simplifica el desarrollo de aplicaciones basadas en hilos. Además asegura la portabilidad de las aplicaciones a diferentes entornos.

# Estados de un proceso

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

Para repartir las CPUs entre los procesos se define en qué estado se encuentran. Los principales son *ejecutándose, bloqueado y preparado*.

Los programas durante su ejecución (con forma de proceso) van pasando por los diferentes estados por partes. Durante la E/S de un proceso, que se queda en estado Bloqueado, otro proceso dispone del procesador para ejecutarse hasta que necesita bloquearse.



# Transiciones entre estados

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

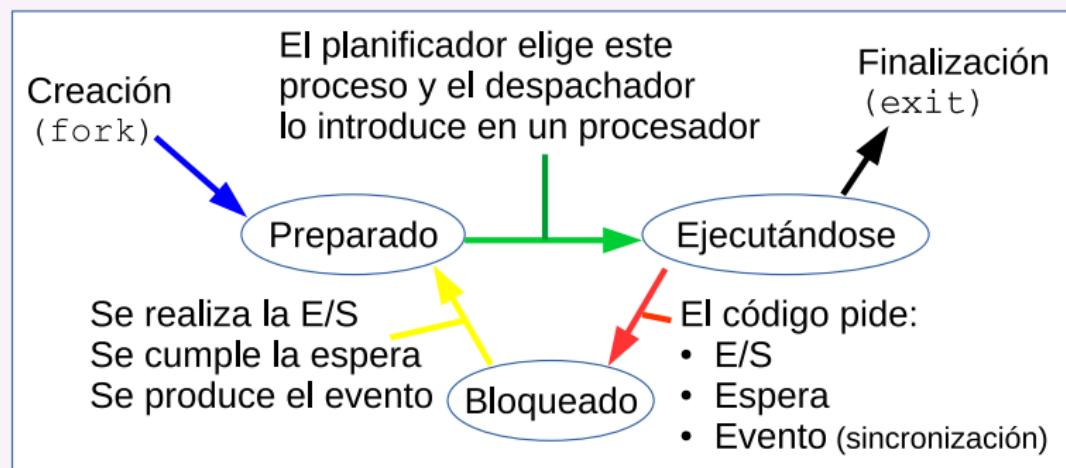
Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Gráfico básico:



# Transiciones entre estados

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

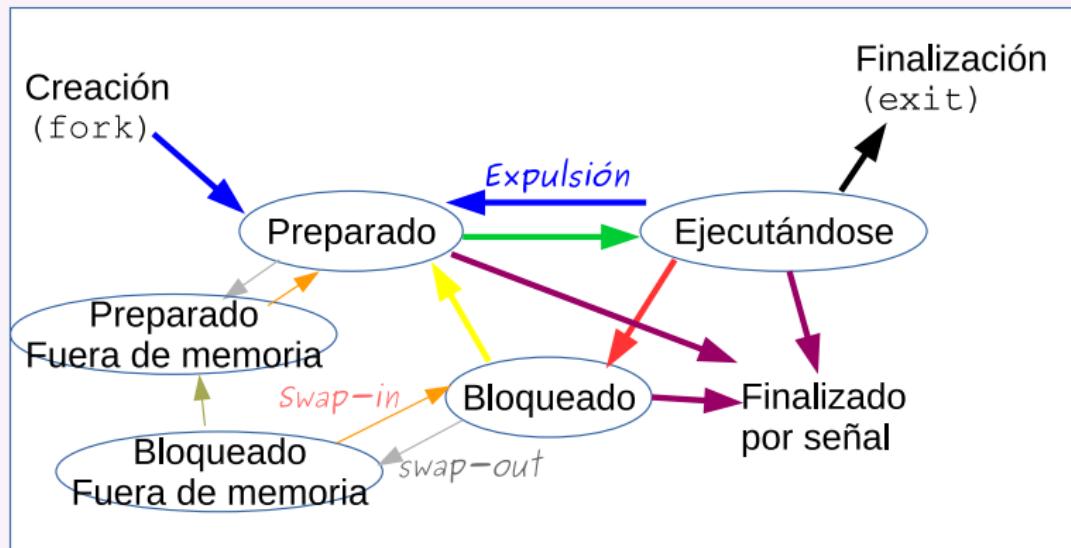
Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Gráfico extendido:



# Cambio de contexto

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

Un **cambio de contexto** (*context switch, CS*) consiste en guardar el estado intermedio de ejecución de un proceso que está en estado *ejecutándose* (datos en CPU que se van a borrar) y sustituirlo por otro que estaba en estado *preparado*.

Esta tarea la lleva a cabo el **planificador**, como último paso tras decidir qué proceso debe ser el siguiente en ejecutarse durante cierto tiempo.

No es una tarea excesivamente pesada puesto que se copian punteros o apunadores (direcciones de memoria donde comienzan las estructuras de datos).



# Interrupciones

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

**Las interrupciones no son procesos, no son programas independientes. Son partes del kernel que se ejecutan de forma asíncrona. Un proceso que debe dejar de ejecutarse por pasar a ejecutarse una interrupción se mantiene en el estado Ejecutándose.**

**La ejecución de código del kernel tampoco se realiza con el formato de proceso, ya que se crearía demasiada sobrecarga.**



# Expulsión

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

Un proceso puede salir **voluntariamente** (debido a su código) del estado *Ejecutándose* por una E/S, una espera de tiempo o una espera a un evento, pasando a **Bloqueado**. En general, por una *Llamada al Sistema* el proceso pasa a *Bloqueado* a la espera de cumplir una cierta operación.

Un proceso puede salir **forzosamente (expulsión)** del estado *Ejecutándose* y volver directamente a **Preparado** cuando:

**Expulsión por tiempo:** se le acaba el *quantum o tiempo máximo de ejecución ininterrumpida*

**Expulsión por evento:** otro proceso cambia a estado *Preparado* y se quiere primar la interactividad



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

### 1 Procesos: Motivación

### 2 Control de procesos

### 3 Representación de los procesos

### 4 Llamadas al Sistema

### 5 Planificación de procesos y procesadores

### 6 Políticas de planificación

### 7 Planificación en multiprocesadores

### 8 Planificación de tareas de tiempo real

### 9 ¿Más preguntas?



# PCB: Bloque de control del proceso

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

**PID** entero identificador único

**estado** Posición en la cola de preparados, o bloqueado o ejecutándose (+)

**datos para planif.** Prioridad, *quantum*...

**contexto de ejecución** estado de CPU y pila

**contabilidad** gestión de consumo, para planificación y *sysadmin*.

**MMU** [Unidad de Gestión de memoria](#) (tablas de páginas...)

**E/S** Peticiones pendientes, dispositivos y tabla de canales

No tiene por qué ser una estructura real, puede estar distribuida en diferentes tablas del [kernel](#).

# Proceso nulo o *idle*

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

Proceso nulo o *idle*: proceso vacío con operaciones NOP o [HLT](#) (*No Operación* o *HaLT*) que ocupan o paran el procesador cuando no hay otro proceso que hacer. Las operaciones NOP y HLT ponen al procesador en una situación de ahorro de energía, junto con el *Gobernador* con los estados C1 y C3.

[Discusión del proceso nulo en StackExchange](#)

You can see various implementations of idle tasks for example in arch/x86/kernel/process.c in the Linux kernel: the basic one just calls HLT, which stops the processor until an interrupt occurs (and enables the [C1 energy-saving mode](#))

[ACPI](#) configura y monitoriza componentes hardware y realiza gestión de energía por ejemplo *durmiento* componentes no usados.



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?



# Llamadas al sistema para identificación del proceso

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Identificación de procesos

```
1 int getpid(); // Devuelve el PID del propio proceso
```

```
int getppid(); // identificador (PID) del proceso padre
```

```
int getuid(); // identificador del usuario al que pertenece  
              del proceso (uid)
```



# Llamadas al sistema para lanzar procesos

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Clonar procesos

```
int fork();
```

## Cargar nuevo ejecutable en un proceso

```
execvp("ls", "ls", "-l", NULL);
```



# Llamadas al sistema para control del proceso

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Finalizar procesos

```
void exit(int estado);
```

Unix guarda el código de finalización (estado) hasta que el proceso padre ejecute `wait()`

`wait()` bloquea al proceso que lo llama hasta que uno de sus hijos finalice. Si no tiene hijos devuelve -1 sin bloquear al proceso que llama. Devuelve el identificador del proceso hijo finalizado y el código de retorno devuelto por el proceso hijo.

```
int wait(int *estado);
```

# Llamadas al sistema para señales

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

Las señales son activaciones asíncronas de funciones del proceso, es decir, cuando un proceso recibe la señal 14 se ejecuta el código de esa señal en paralelo al código del programa. Las funciones están en una tabla y por defecto tienen un comportamiento por defecto. Ver en man 7 signal

Se manda la señal nombreS al proceso pid.

```
int kill(int pid, int nombreS);
```

signal modifica la tabla de señales. Relaciona la función fun con la señal señal

```
int signal (int nombreS, void fun());
```

# Llamadas al sistema para control de tiempos

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Llamadas para control de tiempos y bloqueo

```
void unsigned int sleep (unsigned int segundos);  
// Unix bloquea el proceso durante ese tiempo o hasta que  
    llegue una señal  
  
void pause ();  
// Unix bloquea el proceso hasta que llegue una señal
```



# Llamadas al sistema para señales y tiempos

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Gestión de la señal SIGALRM

```
1 unsigned alarm (unsigned segundos);
// Unix manda SIGALRM al proceso que ha hecho la llamada,
pasados los segundos indicados
```

Este código haría que el programa se bloqueara 3 segundos al ejecutar la función esperar (3):

```
1 void fnula() {
}
2
3 void esperar (unsigned seg) {
    signal(SIGALRM, fnula);
    alarm(seg);
    pause();
}
```



# Llamadas al sistema para control de prioridades

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Llamadas al sistema para control de prioridades

```
int nice (int valor);
```

Vale para modificar las prioridades de los procesos en el acceso a la CPU. De forma análoga, se puede cambiar la prioridad de acceso a disco y red.

Siempre es posible bajar la prioridad de los procesos, sin embargo, para subirla es necesario ser *root* o administrador.



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?

# Planificación de proceso

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

- Gestión de tiempo y de recursos (multiprocesador)
- Parámetros de rendimiento (latencia)
- La política debe ser adecuada al comportamiento de los programas (cálculo/interactivos)
- Problemas del tiempo real y su interacción con las tareas de tiempo compartido
- Multiprocesador y cachés
- Planificación a corto plazo (entrada a la CPU)
- Planificación a largo plazo (prioridad inicial y *quantum*)
- Planificación a medio plazo (*swapping* o intercambio de procesos)

# Comportamiento de los programas

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## Intervalo de CPU

El *Intervalo de CPU* es el tiempo ininterrumpido o consecutivo de ejecución de un programa en la CPU (tiempo de servicio) hasta el paso a Bloqueado o expulsado.

Gráfico de distribución de probabilidades del intervalo de CPU (*heavy-tailed*)



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

## 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?

## Políticas de planificación

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

## Representación de los procesos

## Llamadas al Sistema

Planificación de  
procesos y  
procesadores

## Políticas de planificación

- Selección de un proceso para entrar en CPU
- Expulsión por evento y por tiempo

### Planificación en multiprocesadores

## Planificación de tareas de tiempo

- Selección del proceso que debe pasar al estado Ejecutándose, en función de ciertos criterios
  - Planificación al abandonar un proceso la CPU, o políticas expulsoras
  - Composición de políticas para diferentes tipos de procesos (planificación multinivel)

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

*First come, first served:* FIFO en la cola de preparados, el  
proceso que lleve más tiempo preparado.

Problema: *efecto convoy.*

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

*Smallest job first:* el proceso que lleve menos tiempo terminarlo.

Problema: estimar la duración de un proceso o de su siguiente intervalo de CPU.

# Prioridades

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

## Cola de preparados ordenada por prioridades.

- Para cada prioridad se aplica FCFS.
- Inserción ordenada en la cola.
- Prioridades según usuario, características y necesidades del proceso.
- Prioridades estáticas, o dinámicas para afinar y evitar inanición.
- Ajuste de prioridades periódica o aperiódicamente (en las transiciones de estado)



# Expulsión

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

## No expulsora

Una política **no** es expulsora si la única forma de que un proceso abandone el estado *ejecutándose* es bloqueándose o terminando el propio programa.

# Expulsión por evento

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

Si hay expulsión por evento y un proceso A entra en la cola de Preparados, el proceso B que estaba en *ejecutándose* se intercambia por el A. B pasa a *preparado* y A entra a ejecutarse.

Beneficia los tiempos de respuesta.

La expulsión por evento es necesaria cuando hay procesos de tiempo real (que deben dar resultado en un tiempo máximo).

# Expulsión por tiempo

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

También se llama Turno Circular o *Round-Robin*. Se puede aplicar a FCFS o a una política de prioridades. Los sistemas que la usan se denominan de *tiempo compartido*.

Se define un *quantum* ( $q$ ) o tiempo máximo de ejecución ininterrumpida, valor general o una cantidad de tiempo para cada proceso.

La Rutina de Atención a la Interrupción de Reloj se encarga de determinar cuándo es el final de quantum del proceso en estado *ejecutándose*.



# Planificación multinivel

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Selección de un proceso  
para entrar en CPU

Expulsión por evento y por  
tiempo

Planificación multinivel

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo

Procesos con diferentes necesidades: interactivos, tiempo real,  
*batch*, procesos del sistema...

Una cola por cada tipo de procesos y una política para  
combinarlas: elegir la cola de más importancia no vacía o  
distribuir proporcionalmente a la prioridad.



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Criterios de planificación  
en multiprocesadores

Políticas de planificación  
en multiprocesadores

Planificación de  
tareas de tiempo  
real

## 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?

# Multiprocesadores

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Criterios de planificación  
en multiprocesadores

Políticas de planificación  
en multiprocesadores

Planificación de  
tareas de tiempo  
real

## Ordenador multiprocesador

Conjunto de procesadores con un espacio de direccionamiento  
de memoria física: **SMP** (*Shared-memory Multi-Processor*)

En un ordenador multiprocesador todos los *cores* o núcleos  
comparten la RAM.

No son multicomputadores, cúmulos o *clusters*, en los que la  
memoria RAM es propia de cada procesador. Lo mismo ocurre  
en las tarjetas gráficas.

Las memorias cachés L1 y L2 son importantes por afectar al  
rendimiento.

Tener el **hilo** como entidad planificable mejora el  
rendimiento.

# Tipos de planificación en multiprocesadores

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Criterios de planificación  
en multiprocesadores

Políticas de planificación  
en multiprocesadores

Planificación de  
tareas de tiempo  
real

**Planificación temporal** Política de planificación para cada CPU  
**Planificación espacial** Política de planificación para asignar  
procesos a los procesadores



# Criterios de planificación en multiprocesadores

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Criterios de planificación  
en multiprocesadores

Políticas de planificación  
en multiprocesadores

Planificación de  
tareas de tiempo  
real

## Afinidad al procesador Mantener cada proceso siempre en el mismo procesador

- para aprovechar la *huella* (código y datos del proceso en la caché) y mejorar el rendimiento
- pero pueden quedar procesadores ociosos

## Reparto de carga Cambiar el proceso de procesador para ejecutarlo en el primero libre

- para equilibrar el trabajo entre procesadores y mejorar el rendimiento (no ociosos)
- pero sobrecarga el bus de acceso a RAM y desaprovecha cachés



# Políticas de tiempo compartido en multiprocesadores

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Criterios de planificación  
en multiprocesadores

Políticas de planificación  
en multiprocesadores

Planificación de  
tareas de tiempo  
real

## Políticas de tiempo compartido

Cola global única Reparto de carga

Cola local por procesador Afinidad al procesador



# Políticas de espacio compartido en multiprocesadores

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Criterios de planificación  
en multiprocesadores

Políticas de planificación  
en multiprocesadores

Planificación de  
tareas de tiempo  
real

**Planif. en grupos** Se intentan planificar todos los hilos de una aplicación a la vez

**Particionado** Un grupo de procesadores son de uso exclusivo de una aplicación para la ejecución de sus hilos

El particionado puede ser

**fijo** mismo número de procesadores por aplicación

**flexible** número variable según las necesidades de la aplicación, como las prioridades

**estático** asignado al inicio de la ejecución

**dinámico** variable según el uso de las CPUs asignadas

Tener CPUs ociosas es un tipo de fragmentación interna



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

Tipos de tareas de tiempo  
real

Criterios de planificación  
de tareas de tiempo real

## 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?



# Planificación de tareas de tiempo real

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

Tipos de tareas de tiempo  
real

Criterios de planificación  
de tareas de tiempo real

Una tarea de tiempo real (TR) es un programa que debe dar un resultado en un tiempo determinado, sin importar la carga del sistema (ambulancia).

En sistemas de propósito general es muy difícil que las TR cumplan los plazos. Ha habido mejoras por puntos de expulsión de rutinas del núcleo.

# Tipos de tareas de tiempo real

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

Tipos de tareas de tiempo  
real

Criterios de planificación  
de tareas de tiempo real

¿El sistema admite un funcionamiento degradado (que no siempre se consiga hacer la tarea a tiempo)? ¿Hay parada segura?

**Plazo estricto TR crítico:** central nuclear, piloto automático

**Plazo no estricto Soft real time o no crítico, acrítico:**  
reproducción de audio o vídeo.

**Sistemas empotrados (embedded):** centralitas de un vehículo  
(frenada, estabilidad, sistemas de seguridad).

# Criterios de planificación de tareas de tiempo real

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

Tipos de tareas de tiempo  
real

Criterios de planificación  
de tareas de tiempo real

**La planificación debe ser viable. Con tareas periódicas es posible un análisis de viabilidad previo.**

**La viabilidad es necesaria en un sistema crítico. En uno acrítico se pueden minimizar los retrasos y el tiempo de retardo.**

# Políticas de planificación de tareas de tiempo real

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

Tipos de tareas de tiempo  
real

Criterios de planificación  
de tareas de tiempo real

Frecuencia monótona elegir la que va a terminar antes, SJF  
Plazo más cercano elegir la que *caduca* antes

Cuando no es posible establecer un plan de viabilidad, la  
planificación es dinámica



## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

## 1 Procesos: Motivación

## 2 Control de procesos

## 3 Representación de los procesos

## 4 Llamadas al Sistema

## 5 Planificación de procesos y procesadores

## 6 Políticas de planificación

## 7 Planificación en multiprocesadores

## 8 Planificación de tareas de tiempo real

## 9 ¿Más preguntas?

# ¿Más preguntas?

## CONTENIDOS

Procesos:  
Motivación

Control de  
procesos

Representación de  
los procesos

Llamadas al  
Sistema

Planificación de  
procesos y  
procesadores

Políticas de  
planificación

Planificación en  
multiprocesadores

Planificación de  
tareas de tiempo  
real

¿Más preguntas?

¿Más preguntas?



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

GASTEIZKO  
INGENIARITZA  
ESKOLA  
ESCUELA  
DE INGENIERÍA  
DE VITORIA-GASTEIZ

# 7 Gestión y planificación de procesos.

Introducción a los Sistemas Operativos,  
2025-2026

Pablo González Nalda

Dept. de Lenguajes y Sistemas Informáticos  
EU de Ingeniería de Vitoria-Gasteiz,  
UPV/EHU



24 de marzo de 2026