

# PLANIFICACIÓN

# Objetivos del sistema

- Tiempo rápido de respuesta
- Productividad (Rendimiento)
- Eficiencia del procesador

# Objetivos de la planificación

- ser justa
- elevar al máximo la producción
- aumentar al máximo el número de usuarios interactivos
- ser predecible
- reducir al mínimo el gasto extra
- equilibrar el aprovechamiento de los recursos
- lograr un equilibrio entre la respuesta y el aprovechamiento

# Objetivos de la planificación

- evitar aplazamiento indefinido
- imponer prioridades
- dar preferencia a los procesos que ocupan recursos decisivos
- dar un mejor trato a los procesos que muestren un comportamiento deseable
  - tasas bajas de paginación
- degradarse paulatinamente con las cargas pesadas

# Niveles de planificación (tiempo)

- **Largo plazo**
  - agregar procesos al conjunto de procesos a ejecutar.

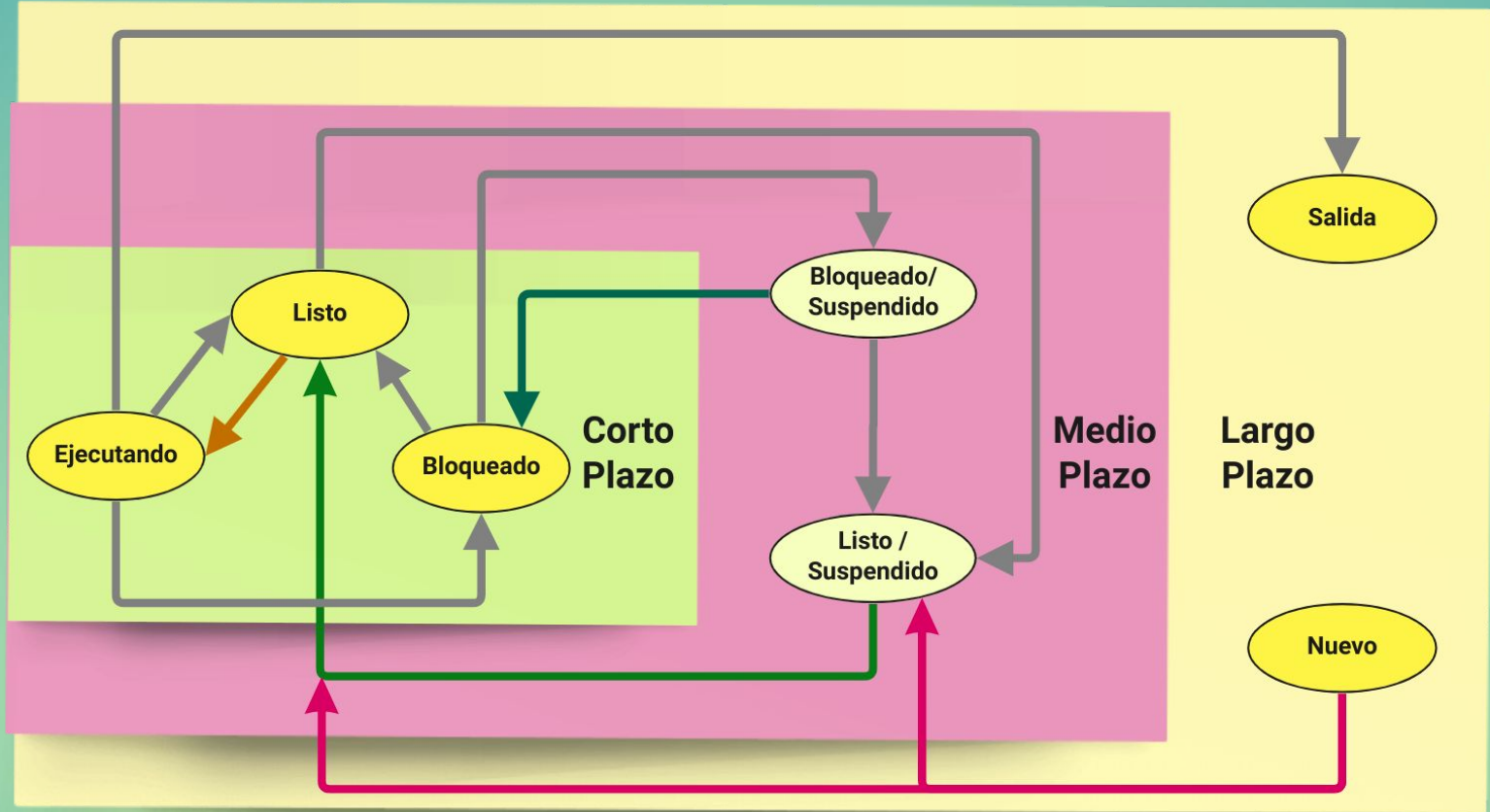
==> se realiza al tomar la decisión de crear un proceso nuevo.
- **Mediano plazo**
  - agregar procesos al conjunto de los que están total o parcialmente en memoria.

==> Fc swapping (memoria de intercambio a principal para poder ejecutar).
- **Corto plazo**
  - qué proceso pasará del estado listo al estado en ejecución.
- **Entrada / Salida**
  - cual de todas las solicitudes de espera de E/S será atendida por un dispositivo disponible

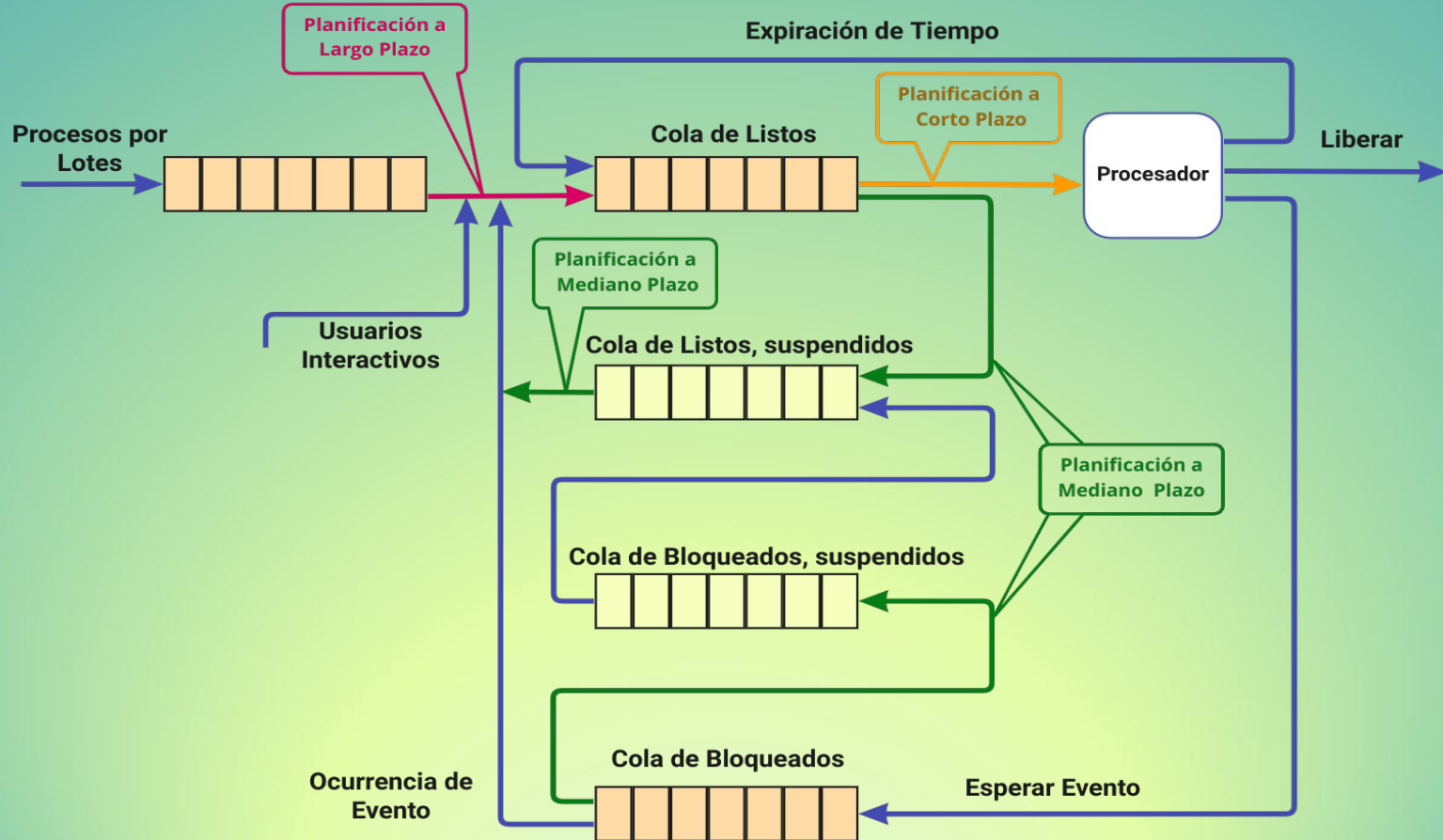
# Planificación y transición de estado de procesos



# Anidamiento de funciones de planificación



# Diagrama de colas de planificación





# Planificación de largo plazo

- Determina qué programas son admitidos por el sistema para procesar.
- Controla el grado de multiprogramación (cantidad de procesos en ejecución)
- A más procesos, menor porcentaje de tiempo para ejecutar cada proceso.

# Planificación de mediano plazo

- Intercambio
- Basada en la necesidad de manejar multiprogramación

# Planificación de corto plazo

- Conocido como despachador.
- Invocada cuando ocurre un evento que puede llevar al bloqueo del proceso actual o que da la oportunidad para ocupar el procesador en favor de otro proceso:
  - interrupciones de reloj
  - interrupciones de E/S
  - llamadas al sistema operativo
  - señales

# **Planificación de monoprocesadores**

## **Algoritmos**

# Criterios de la planificación de corto plazo

- Orientado al usuario:

- Tiempo de respuesta: *Tiempo transcurrido entre que se emite un pedido y la respuesta aparece en la salida.*

- Orientado al sistema:

- utilización efectiva y eficiente del procesador.

- Relativo a las prestaciones:

- se pueden medir, como el tiempo de respuesta y el rendimiento.

- No relativo al rendimiento:

- no se pueden medir, cualitativos, como la previsibilidad.

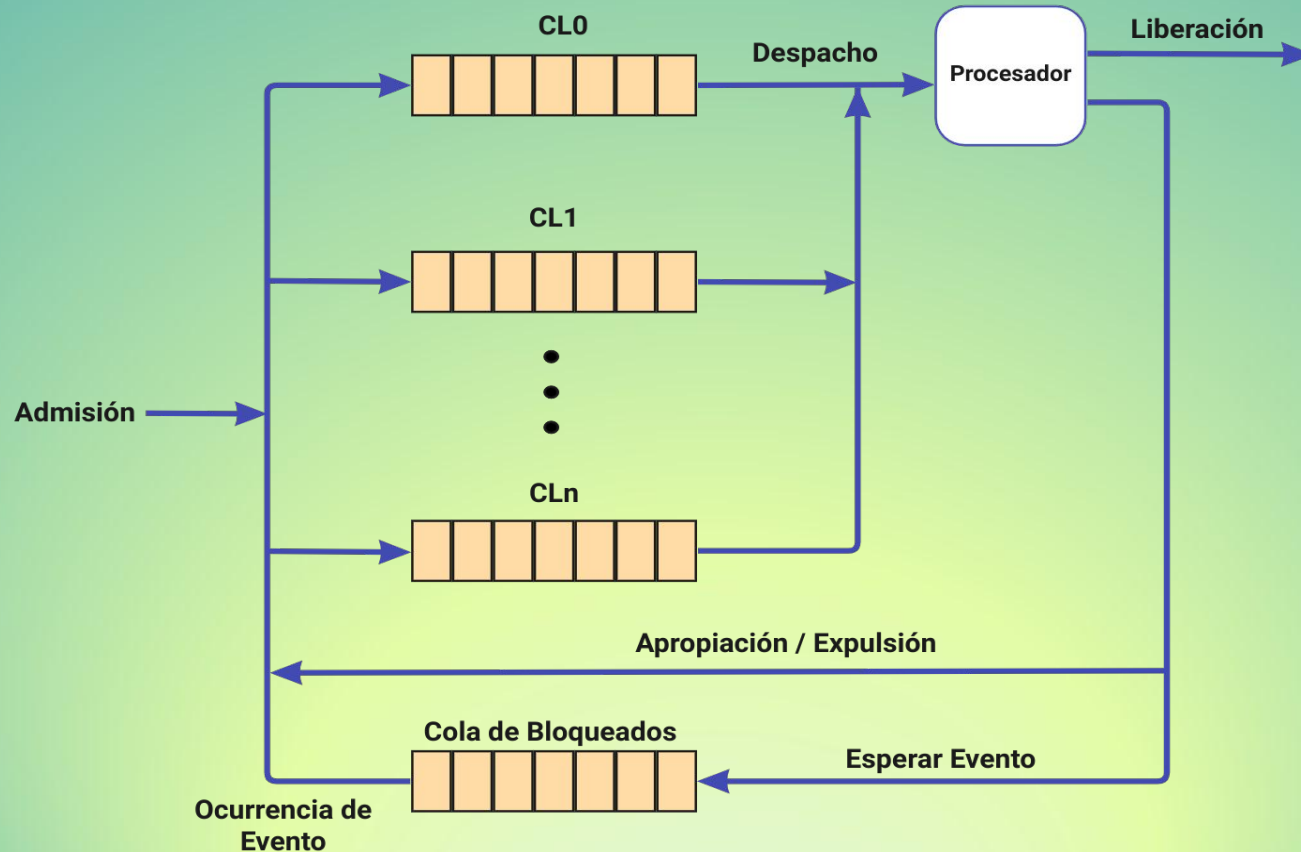
# Criterios de la planificación de corto plazo

	Usuario	Sistema
cuantitativo	Tiempo de retorno Tiempo de respuesta Plazos	Productividad Utilización del procesador
cualitativo	Previsibilidad	Equidad Prioridades Equilibrio de recursos

# Prioridades

- El planificador siempre seleccionará un proceso de mayor prioridad antes que uno de menor prioridad.
- Tiene múltiples colas de listos que representan cada nivel de prioridad.
- Los de prioridad menor pueden sufrir de inanición:
  - Entonces puede permitir a un proceso cambiar su prioridad basado en su antigüedad o historial de ejecución

# Planificación por prioridades





# Función de selección

- Determina qué proceso se elige
- Basada en:
  - prioridades
  - necesidades de recursos
  - características de ejecución de los procesos
    - **W** = tiempo consumido hasta el momento en el sistema, esperando y ejecutando.
    - **e** = tiempo consumido hasta el momento en ejecución.
    - **S** = tiempo total de servicio exigido por el proceso, incluido **e**.

# Modo de decisión

- Especifica los instantes de tiempo en que se aplica la función de selección. Hay dos categorías:
  - **Sin expulsión (no apropiativo, non preemptive)**
    - Una vez que el proceso pasa a estado de ejecución, continúa ejecutando hasta que *termina* o hasta que se *bloquea por E/S*.
  - **Con expulsión (apropiativo, preemptive)**
    - El proceso ejecutándose actualmente puede ser interrumpido y llevado al estado de listo por el sistema operativo.
    - Permite mejor servicio ya que ningún proceso monopoliza el procesador por mucho tiempo.

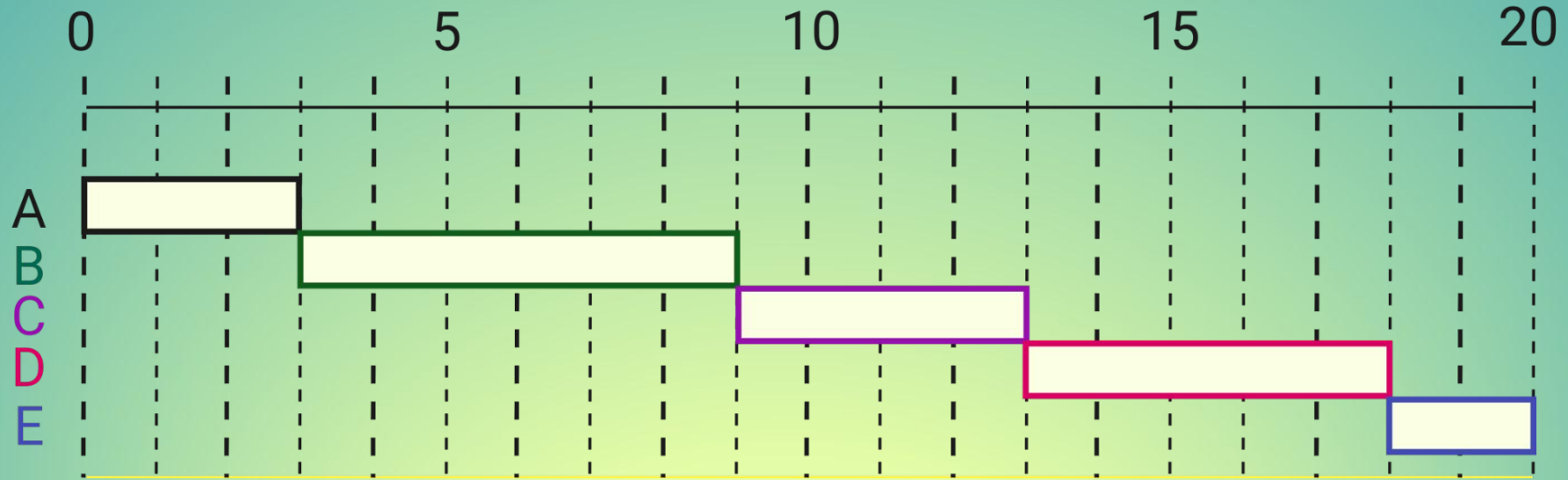
# Características de algunas políticas de planificación

	Función de selección	Modo de Decisión	Rendimiento	Tiempo de respuesta	Sobre-carga	Efecto sobre los procesos	Inanición
FCFS	$\max[w]$	no expulsiva	no especificado	puede ser alto especialmente si hay mucha diferencia entre los tiempos de ejecución de los procesos	mínima	penaliza procesos cortos; penaliza procesos con mucha e/s	no
Turno rotatorio(RR)	constante	expulsiva (por rodajas de tiempo)	puede ser bajo si la rodaja es demasiado pequeña	proporciona buen tiempo de respuesta para procesos cortos	mínima	tratamiento justo	no
SPN	$\min[s]$	no expulsiva	alto	proporciona buen tiempo de respuesta para procesos cortos	puede ser alta	penaliza procesos largos	posible
SRT	$\max[s-e]$	expulsiva	alto	proporciona buen tiempo de respuesta	puede ser alta	penaliza procesos largos	posible
HRRN	$\max[(w+s)/s]$	no expulsiva	alto	proporciona buen tiempo de respuesta	puede ser alta	buen equilibrio	no
Realimentación	(ver texto)	expulsiva (por rodajas de tiempo)	no especificado	no especificado	puede ser alta	puede favorecer procesos con mucha e/s	posible

# Ejemplo de planificación de procesos

Proceso	Tiempo de llegada	Tiempo de servicio
<b>A</b>	<b>0</b>	<b>3</b>
<b>B</b>	<b>2</b>	<b>6</b>
<b>C</b>	<b>4</b>	<b>4</b>
<b>D</b>	<b>6</b>	<b>5</b>
<b>E</b>	<b>8</b>	<b>2</b>

# Primero en llegar, primero en ser servido (FCFS)

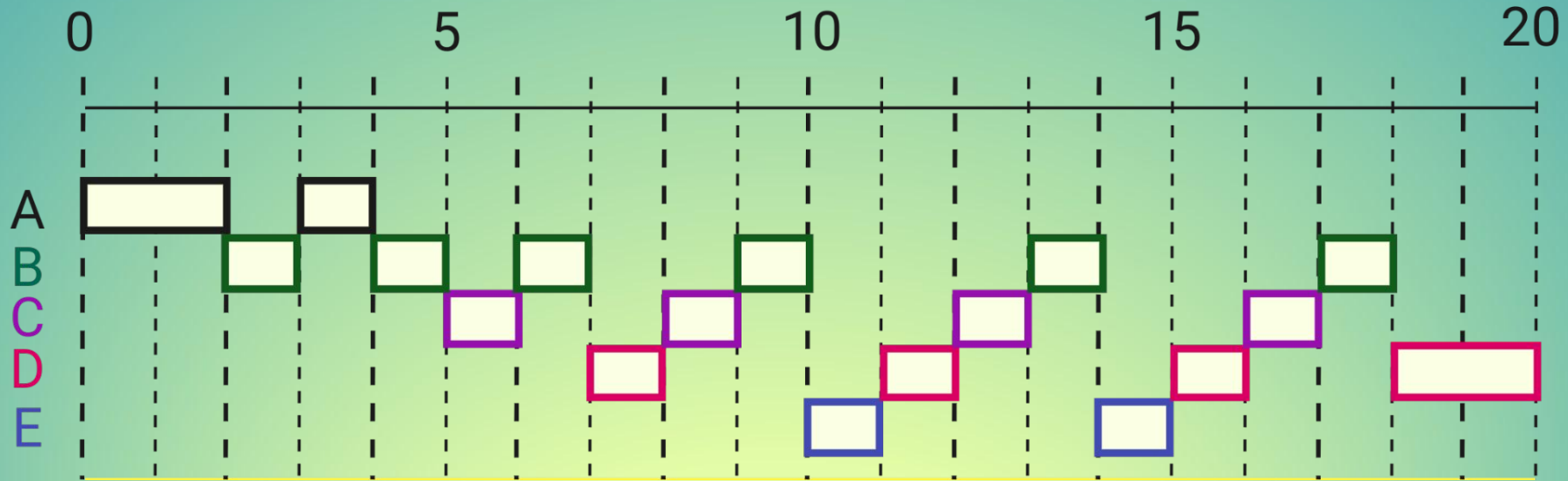


- Cada proceso se une a la cola de listos.
- Cuando el proceso actual termina de ejecutarse, se elige el proceso más antiguo de la cola de listos.

# Primero en llegar, primero en ser servido (FCFS)

- Un proceso corto puede tener que esperar mucho tiempo antes de ser ejecutado.
- Favorece a los procesos con carga de CPU.
  - Procesos de E/S deben esperar hasta que los de carga de CPU se terminen

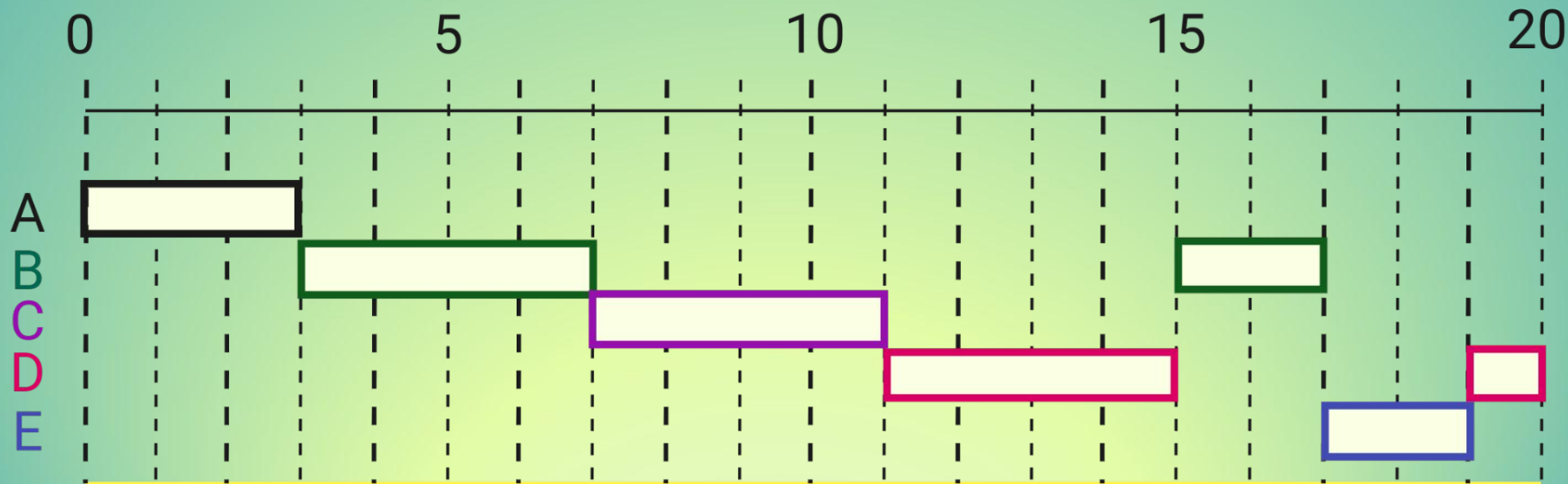
# Round-Robin (turno rotatorio)



- Usa apropiación basada en reloj
- Se determina la cantidad de tiempo fija que cada proceso usa el procesador. (ej:  $q_{\text{tiempo}}=1$ )

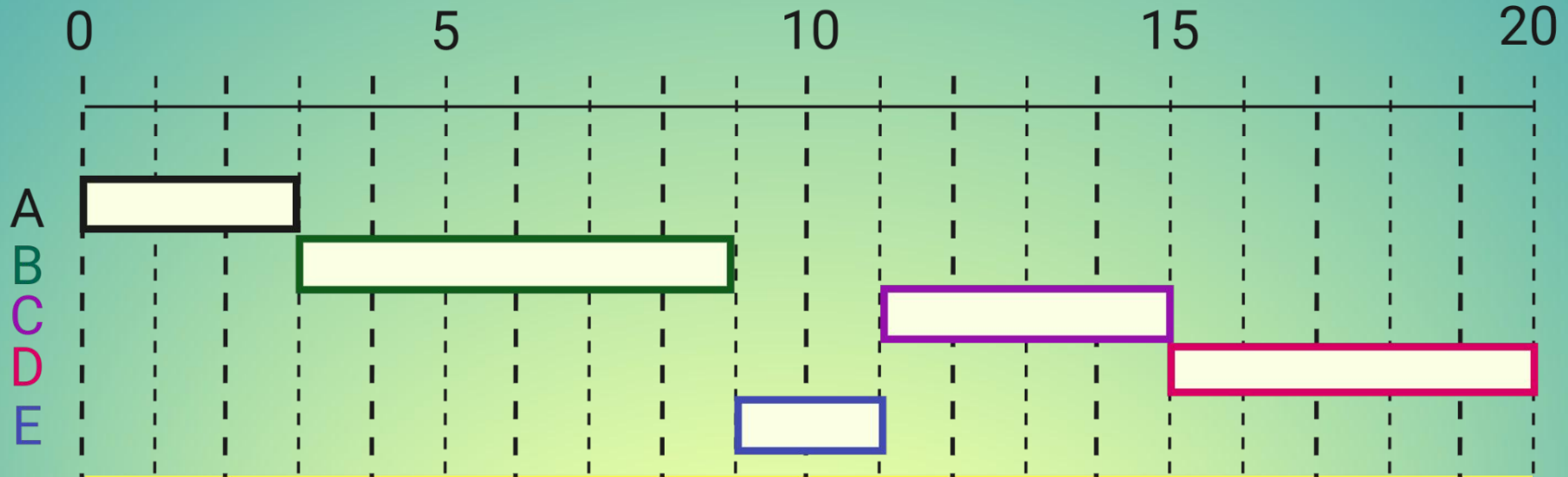
# Round-Robin (turno rotatorio)

(ej:  $q_{\text{tiempo}}=4$ )





# Primero el más corto (SPN)

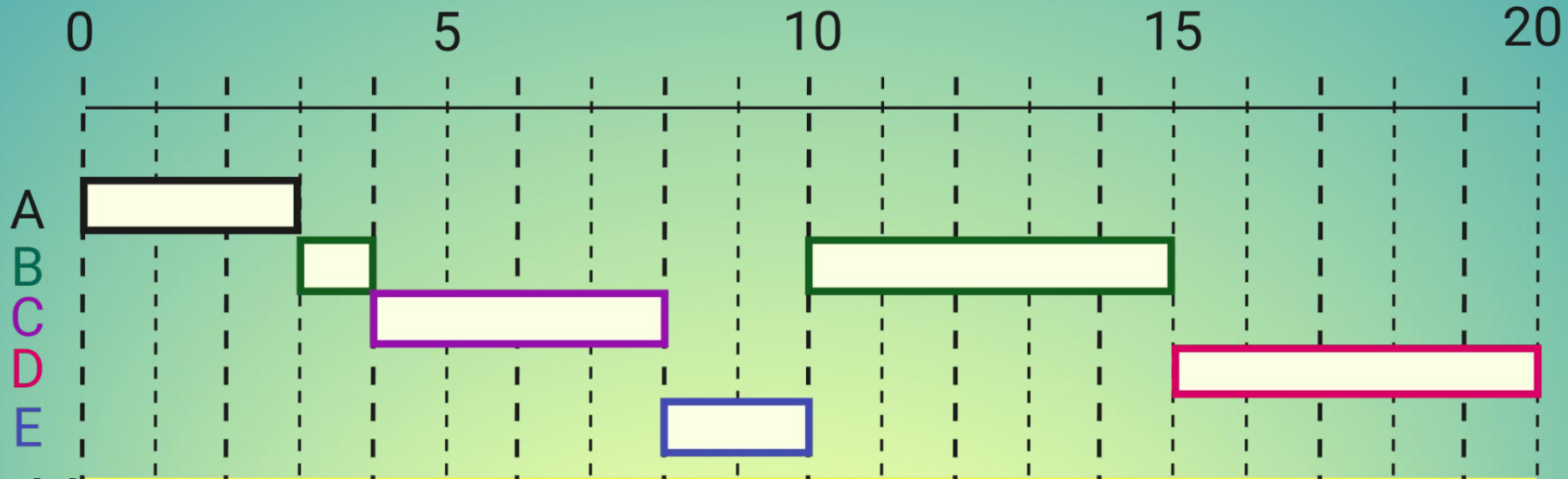


- Política no apropiativa (sin expulsión)
- Se elige el proceso con el menor tiempo de procesamiento
- Los procesos cortos sobrepasan a los procesos más largos

# Primero el más corto (SPN)

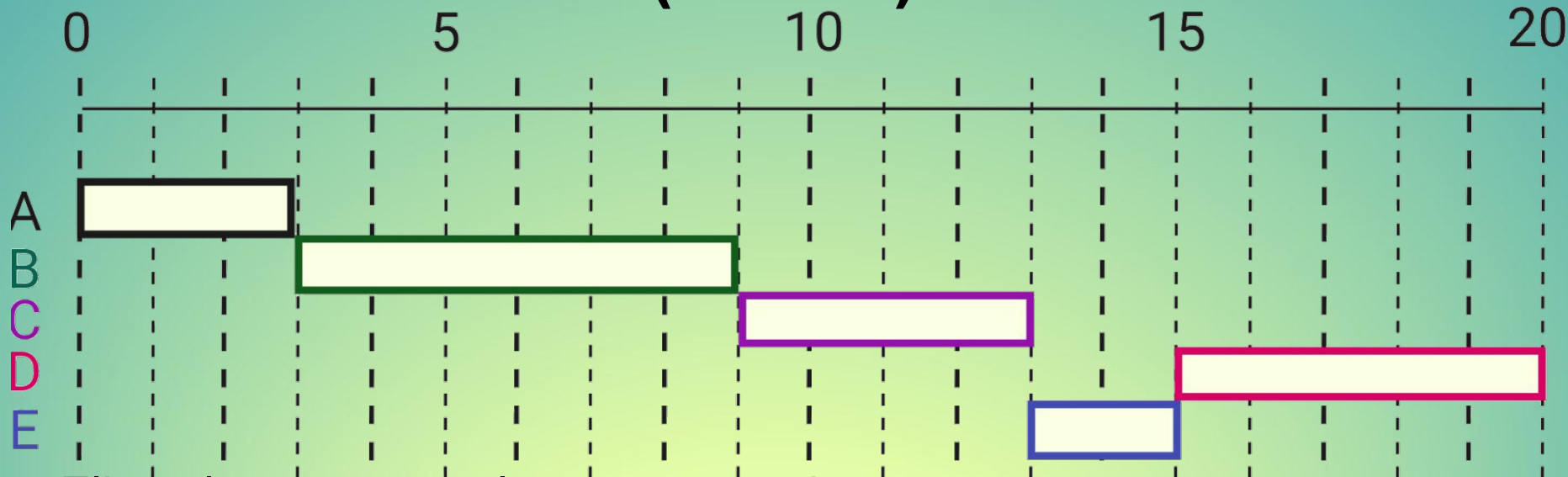
- Se reduce la previsibilidad de los procesos más largos
- Si el tiempo estimado para el proceso no es correcto, el sistema operativo puede abandonarlo
- Posibilidad de inanición para procesos más largos

# Menor tiempo restante (SRT)



- Versión apropiativa de la política de primero el más corto
- Debe estimarse el tiempo de procesamiento

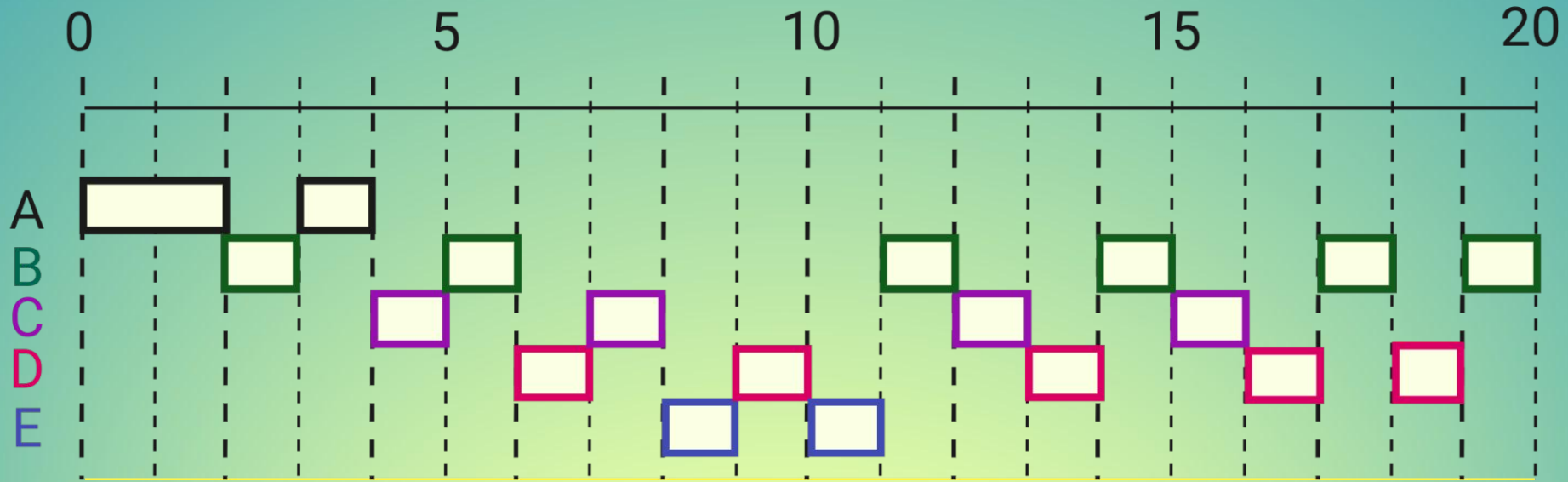
# Primero el de mayor tasa de respuesta (HRRN)



- Elige el proceso con la mayor tasa de respuesta:

$$R = \frac{\text{Tiempo consumido esperando } W + \text{Tiempo de servicio esperado } S}{\text{Tiempo de servicio esperado } S}$$

# Realimentación



- Penaliza los trabajos que han estado ejecutándose por más tiempo
- No se conoce el tiempo remanente que el proceso necesita

# Volvemos a :Características de políticas de planificación

	Función de selección	Modo de Decisión	Rendimiento	Tiempo de respuesta	Sobre-carga	Efecto sobre los procesos	Inanición
FCFS	$\max[w]$	no expulsiva	no especificado	puede ser alto especialmente si hay mucha diferencia entre los tiempos de ejecución de los procesos	mínima	penaliza procesos cortos; penaliza procesos con mucha e/s	no
Turno rotatorio(RR)	constante	expulsiva (por rodajas de tiempo)	puede ser bajo si la rodaja es demasiado pequeña	proporciona buen tiempo de respuesta para procesos cortos	mínima	tratamiento justo	no
SPN	$\min[s]$	no expulsiva	alto	proporciona buen tiempo de respuesta para procesos cortos	puede ser alta	penaliza procesos largos	posible
SRT	$\max[s-e]$	expulsiva	alto	proporciona buen tiempo de respuesta	puede ser alta	penaliza procesos largos	posible
HRRN	$\max[(w+s)/s]$	no expulsiva	alto	proporciona buen tiempo de respuesta	puede ser alta	buen equilibrio	no
Realimentación	(ver texto)	expulsiva (por rodajas de tiempo)	no especificado	no especificado	puede ser alta	puede favorecer procesos con mucha e/s	posible

# Planificación por contribución justa

- La aplicación del usuario se ejecuta como un conjunto de procesos (hilos)
- El usuario está interesado en el rendimiento de la aplicación
- Necesita tomar decisiones de planificación basadas en grupos de procesos

# Planificación por contribución justa

Tiempo	Proceso A			Proceso B			Proceso C		
	Prioridad	Contador CPU proceso	Contador CPU grupo	Prioridad	Contador CPU proceso	Contador CPU grupo	Prioridad	Contador CPU proceso	Contador CPU grupo
0	60	0 1 2 • • 60	0 1 2 • • 60	60	0	0	60	0	0
1	90	30	30	60	0 1 2 • • 60	0 1 2 • • 60	60	0	0 1 2 • • 60
2	74	15 16 17 • • 75	15 16 17 • • 75	90	30	30	75	0	30
3	96	37	37	74	15 16 17 • • 75	15 16 17 • • 75	67	0 1 2 • • 60	15 16 17 • • 75
4	78	18 19 20 • • 78	18 19 20 • • 78	81	7	37	93	30	37
5	98	39	39	70	3	18	76	15	18

Grupo 1
Grupo 2



# Planificación por contribución justa

## (1 de 3)

	Proceso A			Proceso B			Proceso C		
Tiempo	Prioridad	Contador CPU Proceso	Contador CPU Grupo	Prioridad	Contador CPU Proceso	Contador CPU Grupo	Prioridad	Contador CPU Proceso	Contador CPU Grupo
0	60	0 1 2 . . 60	0 1 2 . . 60	60	0 1 2 . . 60	0 1 2 . . 60	60	0 1 2 . . 60	0 1 2 . . 60
1	90	30 . . . . 60	30 . . . . 60	60	0 1 2 . . 60	0 1 2 . . 60	60	0 1 2 . . 60	0 1 2 . . 60

# Planificación por contribución justa

## (2 de 3)

	Proceso A			Proceso B			Proceso C		
Tiempo	Prioridad	Contador CPU Proceso	Contador CPU Grupo	Prioridad	Contador CPU Proceso	Contador CPU Grupo	Prioridad	Contador CPU Proceso	Contador CPU Grupo
2	74	15 16 17 . . 75	15 16 17 . . 75	90	30	30	75	0	30
3	96	37	37	74	15	15 16 17 . . 75	67	0 1 2 . . 60	15 16 17 . . 75

# Planificación por contribución justa

## (3 de 3)

	Proceso A			Proceso B			Proceso C		
		Contador CPU Proceso	Contador CPU Grupo		Contador CPU Proceso	Contador CPU Grupo		Contador CPU Proceso	Contador CPU Grupo
Tiempo	Prioridad			Prioridad			Prioridad		
4	78	18	18	81	7	37	93	30	37
		19	19						
		20	20						
		.	.						
		.	.						
		78	78						
5	98	39	39	70	3	18	76	15	18
	Grupo 1			Grupo 2					

# Planificación UNIX tradicional

- Realimentación multinivel usando round robin en cada cola de prioridad
- Las prioridades se calculan una vez por segundo
- La prioridad base divide a todos los procesos en bandas fijas de niveles de prioridad
- Se usa un factor de ajuste para mantener cada proceso en su banda asignada

# Planificación UNIX tradicional

## Bandas

- **Orden decreciente de prioridad**
  - Swapper
  - Control dispositivos E/S al bloque
  - Manejo de archivos
  - Control dispositivos E/S al caracter
  - Procesos usuario

# Planificación UNIX tradicional

Tiempo	Proceso A		Proceso B		Proceso C	
	Prioridad	Contador CPU	Prioridad	Contador CPU	Prioridad	Contador CPU
0	60	0 1 2 • • 60	60	0	60	0
1	75	30	60	0 1 2 • • 60	60	0
2	67	15	75	30	60	0 1 2 • • 60
3	63	7 8 9 • • 67	67	15	75	30
4	76	33	63	7 8 9 • • 67	67	15
5	68	16	76	33	63	7

El rectángulo coloreado representa el proceso en ejecución

# Planificación UNIX tradicional (1 de 2)

	Proceso A		Proceso B		Proceso C	
Tiempo	Prioridad	Contador CPU	Prioridad	Contador CPU	Prioridad	Contador CPU
0	60	0 1 2 . . 60	60	0	60	0
1	75	30	60	0 1 2 . . 60	60	0
2	67	15	75	30	60	0 1 2 . . 60

# Planificación UNIX tradicional (2 de 2)

	Proceso A		Proceso B		Proceso C	
Tiempo	Prioridad	Contador CPU	Prioridad	Contador CPU	Prioridad	Contador CPU
3	63	7 8 9 . . 67	67	15	75	30
4	76	33	63	7 8 9 . . 67	67	15
5	68	16	76	33	63	7



# **Planificación multiprocesadores**

# Clasificación de multiprocesadores

- Multiprocesadores débilmente acoplados
  - cada procesador tiene su propia memoria y canales de E/S
- Procesadores funcionalmente especializados
  - como procesador de E/S
  - controlado por un procesador maestro
- Multiprocesadores fuertemente acoplados
  - procesadores que comparten memoria ppal
  - controlado por el sistema operativo

# Granularidad

- Paralelismo independiente Separa procesos en ejecución
  - No hay sincronización
  - Más de un procesador disponible
    - Menor tiempo de respuesta promedio para usuarios

# Granularidad

- Paralelismo grano muy grueso o grueso
  - Sincronización burda
  - Similar a ejecutar muchos procesos en un procesador excepto que se reparte en más procesadores

# Granularidad

- Paralelismo de grano medio
  - Procesamiento paralelo o multitarea dentro de una aplicación única
  - Una aplicación única es una colección de hilos
  - Los hilos interactúan de forma muy frecuente
  - Las decisiones de planificación pueden afectar al rendimiento de la aplicación

# Granularidad

- Paralelismo de grano fino
  - Aplicaciones muy paralelas
  - Campo muy especializado y fragmentado

# Elementos de diseño

- Asignación de procesos a procesadores
- Uso de multiprogramación en los procesadores individuales
- Despacho real de los procesos

# Asignación de procesos a procesadores

- Tratar a los procesadores como un recurso reservado y asignar por demanda
- Asignación permanente de un proceso a un procesador
  - Cola corto plazo dedicada para cada procesador
  - Menos sobrecarga
  - Puede haber procesadores desocupados
- Cola global
  - Ejecución en cualquier procesador disponible



# Asignación de procesos a procesadores

- Arquitectura master/slave
  - Funciones clave del kernel siempre en un procesador particular
  - El master es el responsable de la planificación
  - Los esclavos deben enviar pedido de servicio
  - Desventajas:
    - Falla del master hace caer el sistema
    - Master: cuello de botella

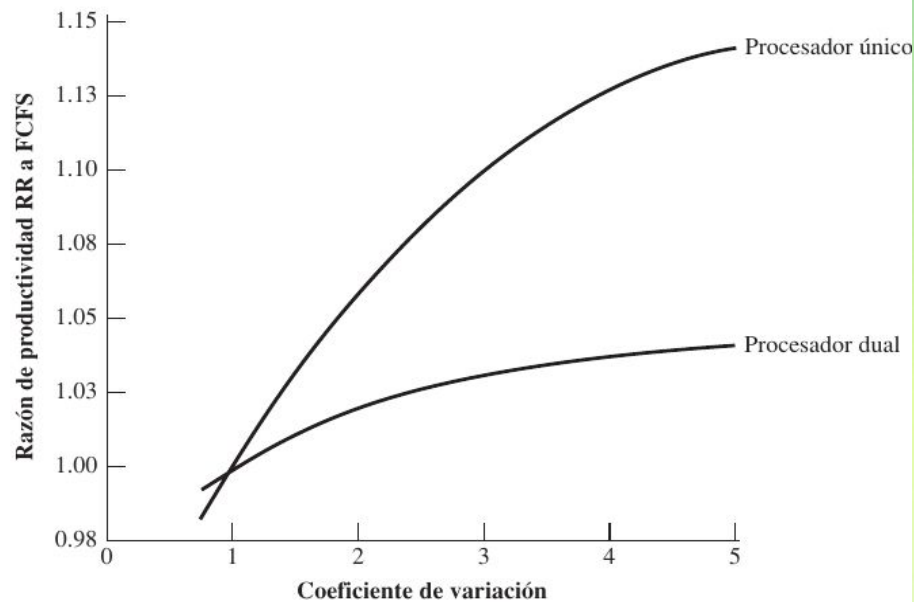
# Asignación de procesos a procesadores

- Arquitectura peer (camaradas)
  - El sistema operativo se puede ejecutar en cualquier procesador
  - Cada procesador puede hacer autoplanificación
  - Sistema operativo más complejo
    - Cuidado de que dos procesadores no elijan el mismo proceso

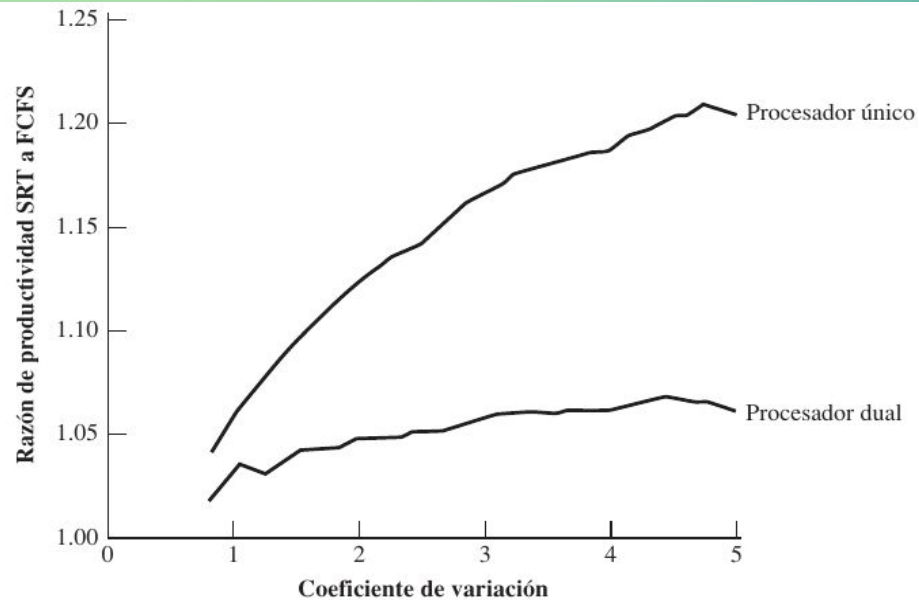
# Planificación de procesos

- Una sola cola para todos los procesos
- Múltiples colas usadas para prioridades
- Todas las colas alimentan una reserva común de procesadores
- La disciplina de planificación específica es menos importante con más de un procesador

# Comparación de rendimiento para uno y dos procesadores



(a)



(b)

# Planificación de hilos

- La ejecución se separa del resto de la definición del proceso
- Una aplicación puede ser un conjunto de hilos que cooperan y ejecutan concurrentemente en el mismo espacio de direcciones
- Los hilos ejecutándose en procesadores separados permiten una gran ganancia en rendimiento

# Propuestas de planificación de hilos en multiprocesadores

- Compartición de carga
  - Los procesos no están asignados a un procesador particular
- Planificación en pandilla
  - Planificación para ejecutar un conjunto de hilos relacionados en un conjunto de procesadores al mismo tiempo
- Asignación de procesador dedicado
  - Los hilos se asignan a un procesador específico durante toda su ejecución (procesador dedicado)
- Planificación dinámica
  - Cambio dinámico del número de hilos durante la ejecución

# Compartición de carga

- Se distribuye equitativamente la carga entre los procesadores
- Ningún procesador está ocioso mientras hay trabajo disponible
- No se requiere planificador centralizado
- Uso de colas globales

# Desventajas de la compartición de carga

- La cola central necesita exclusión mutua
  - Se puede convertir en un cuello de botella cuando más de un procesador busca trabajo al mismo tiempo
- Es improbable que los hilos expulsados reanuden su ejecución en el mismo procesador
  - Uso de cache menos eficiente
- Si todos los hilos de un programa están en la cola central, no todos accederán a los procesadores al mismo tiempo



# Planificación en pandilla

- Planificación simultánea de hilos que forman un solo proceso
- Útil para aplicaciones donde el rendimiento se degrada mucho más cuando alguna parte no se ejecuta
- Los hilos necesitan sincronizarse entre ellos

# Planificación en pandilla con cuatro y un hilos

División uniforme		
	Grupo 1	Grupo 2
PE1		
PE2		Ocioso
PE3		Ocioso
PE4		Ocioso
Tiempo	1/2	1/2
37,5% Desperdicio		

División por pesos		
	Grupo 1	Grupo 2
PE1		
PE2		Ocioso
PE3		Ocioso
PE4		Ocioso
Tiempo	4/5	1/5
15% Desperdicio		

**Figura 10.2.** Ejemplo de planificación de grupos con cuatro y un hilos [FEIT90b].

# Asignación de procesador dedicado

- Cuando se planifica la aplicación, sus hilos se asignan a un procesador dedicado hasta que termine la aplicación
- Algunos procesadores pueden estar desocupados
- Evita intercambio de procesos, mayor velocidad del programa

# Planificación dinámica

- El sistema operativo ajusta la carga para mejorar el uso
  - Asignación de procesadores desocupados
  - Asignación de un procesador a un recién llegado, quitándoselo a algún proceso que esté usando más de un procesador
  - Pedidos pendientes hasta que haya un procesador disponible, si no se pudo satisfacer alguna parte de la petición
  - Mayor prioridad recién llegados que aplicaciones en ejecución

# **Planificación sistemas de tiempo real**

# Sistemas de tiempo real

- Control de experimentos de laboratorio
- Procesos de control de edificios
- Robotica
- Control de tráfico aéreo
- Telecomunicaciones
- Sistemas de control y mando militar

# Sistemas de tiempo real

- La exactitud del sistema no solo depende del resultado lógico de un cálculo sino también del tiempo en el que se producen los resultados
- Las tareas o procesos intentan controlar o reaccionar ante eventos del mundo exterior
- Estos sucesos ocurren en “tiempo real” y los procesos deben poder responder adecuadamente
  - Normalmente es posible asociar un plazo (comienzo y final) a una tarea específica

# Tareas de tiempo real. Tipos

- Tarea rígida de tiempo real
  - Debe cumplir el plazo
- Tarea flexible de tiempo real
  - Es conveniente cumplir el plazo
- Tarea aperiódica
  - Comienzo y final
- Tarea periódica
  - Una vez por cada periodo  $T$ , por ej.



# Requisitos de los sistemas operativos en tiempo real

- **Determinismo**
  - Las operaciones se realizan en instantes fijos y predeterminados o en intervalos de tiempo predeterminado
  - Tiene que ver con el retardo máximo desde la llegada de una interrupción hasta que esta sea reconocida (pocos microsegundos)

# Requisitos de los sistemas operativos en tiempo real

- Sensibilidad

- Cuánto tiempo consume un sistema operativo en dar servicio a la interrupción después de reconocerla
- Incluye la cantidad de tiempo para comenzar la ejecución de su rutina de tratamiento

# Requisitos de los sistemas operativos en tiempo real

- Control del usuario
  - Se debe permitir que el usuario especifique
    - Prioridad
    - Uso de paginación
    - Intercambio de procesos
    - Procesos que deben estar residentes en memoria
    - Algoritmos de transferencia de disco
    - Tareas rígidas y flexibles (plazos fijos obligatorios o solo convenientes)

# Requisitos de los sistemas operativos en tiempo real

- **Confiabilidad**
  - Degradación del rendimiento puede tener consecuencias catastróficas
- **Tolerancia a fallos**
  - Intenta corregir el problema o minimizar sus efectos mientras se sigue ejecutando
  - Ejecuta tareas más críticas, de alta prioridad

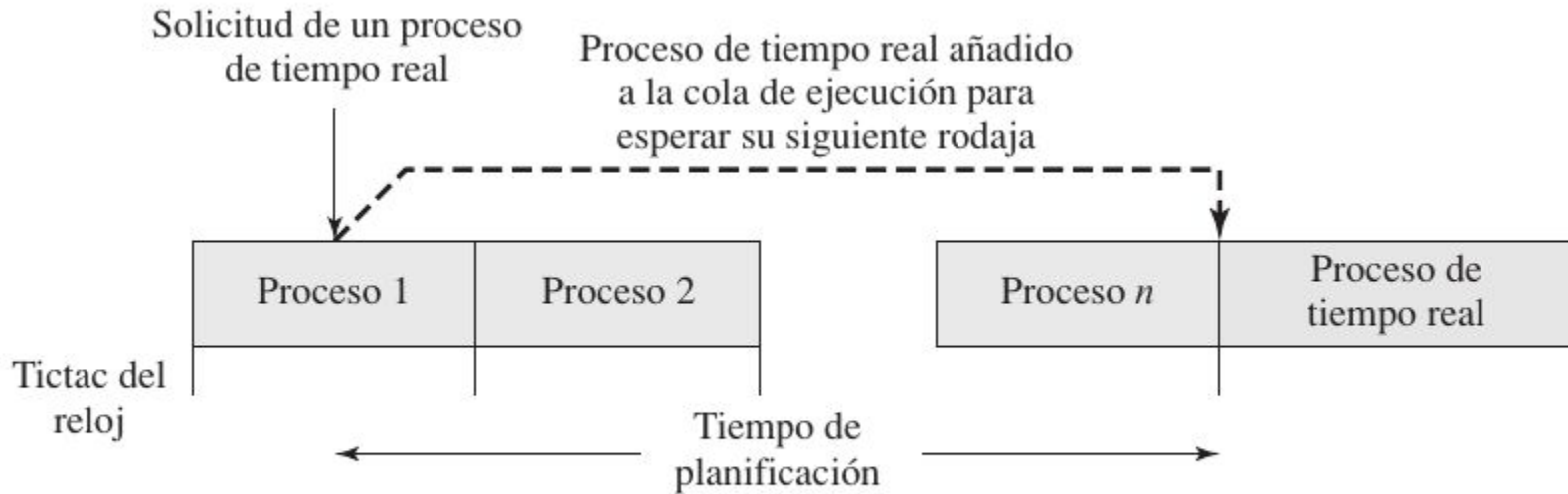
# Características de los sistemas operativos en tiempo real

- Cambios rápidos de contexto
- Pequeño tamaño
- Capacidad de responder rápidamente a interrupciones externas
- Multitarea con herramientas de comunicación tales como semáforos, señales y eventos
- Uso de archivos secuenciales que acumulen datos a alta velocidad

# Características de los sistemas operativos en tiempo real

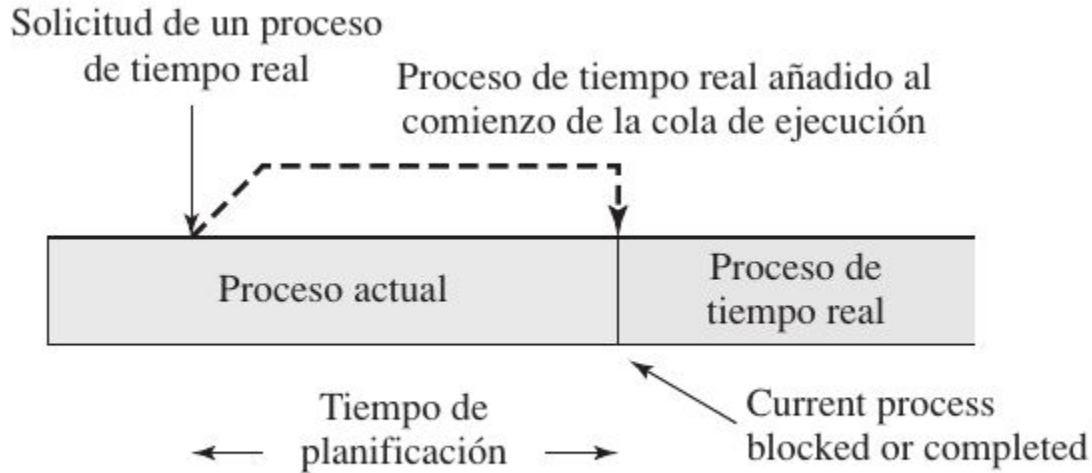
- Planificación apropiativa basada en prioridades
- Minimización de intervalos durante los cuales se inhabiliten las interrupciones
- Primitivas para demorar tareas durante un tiempo fijo
- Alarmas especiales y temporizadores

# Planificación de un proceso de tiempo real



(a) Planificador expulsivo de turno circular

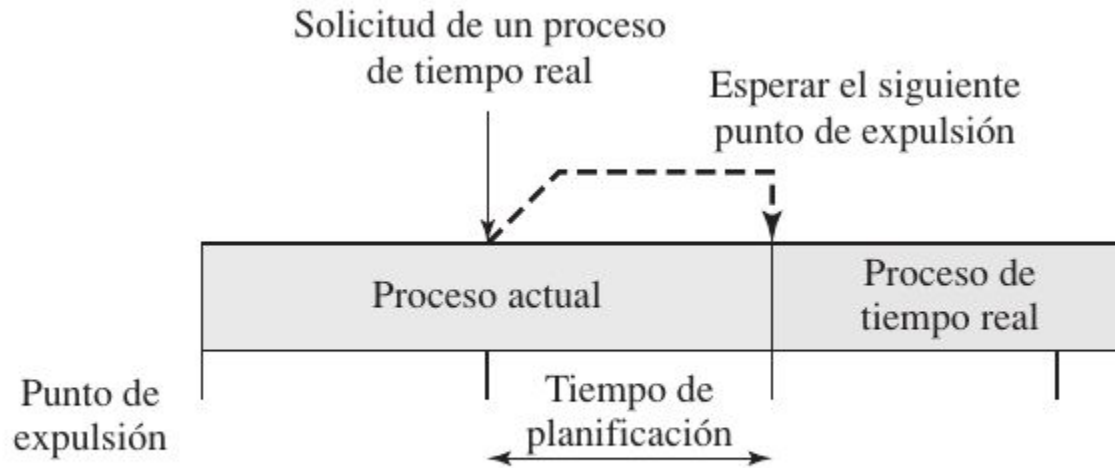
# Planificación de un proceso de tiempo real



(b) Planificador no expulsivo dirigido por prioridad

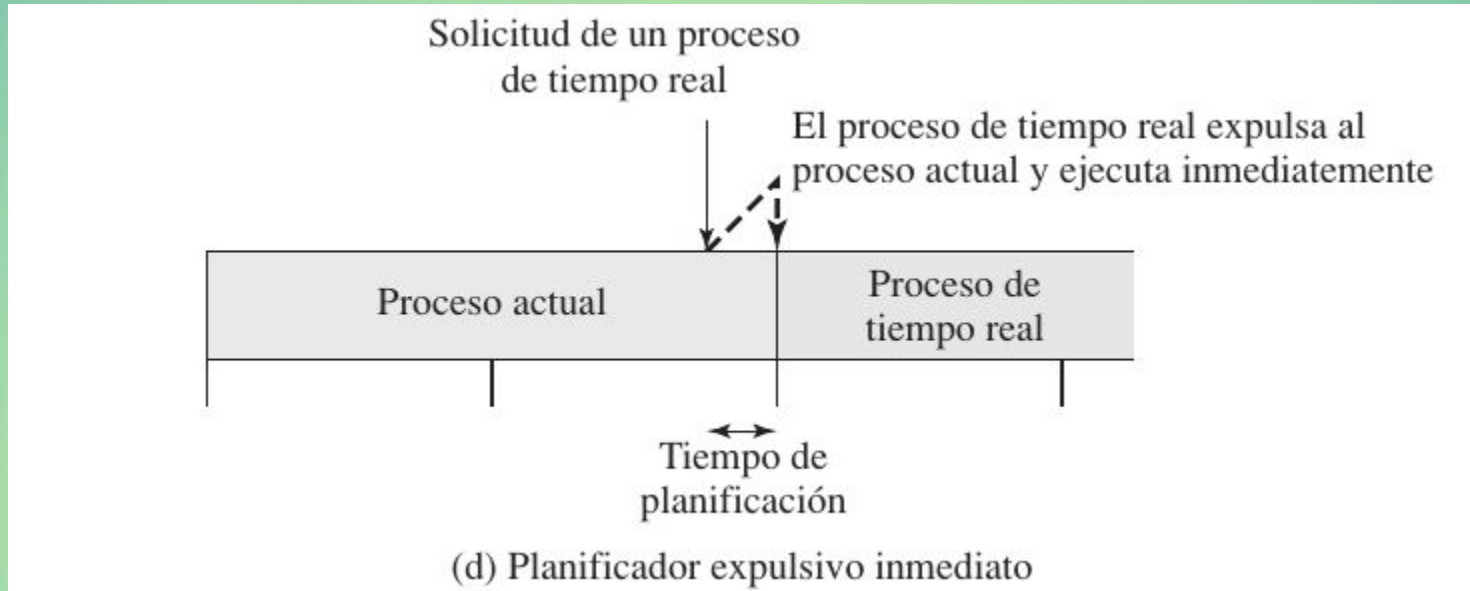


# Planificación de un proceso de tiempo real



(c) Planificador expulsivo dirigido por prioridad en puntos de expulsión

# Planificación de un proceso de tiempo real



# Planificación de tiempo real

- Planificación con tablas estáticas
  - Determina en tiempo de ejecución cuándo una tarea debe comenzar a ejecutarse
- Planificación apropiativa con prioridades estáticas
  - Se usa el planificador tradicional de prioridades
- Planificación dinámica basada en un plan
  - Una tarea será aceptada sólo si es posible satisfacer sus restricciones de tiempo. Plan.
- Planificación dinámica del mejor resultado
  - Sin análisis de factibilidad. El sistema intenta cumplir todos los plazos y aborta los procesos cuyo plazo haya fallado

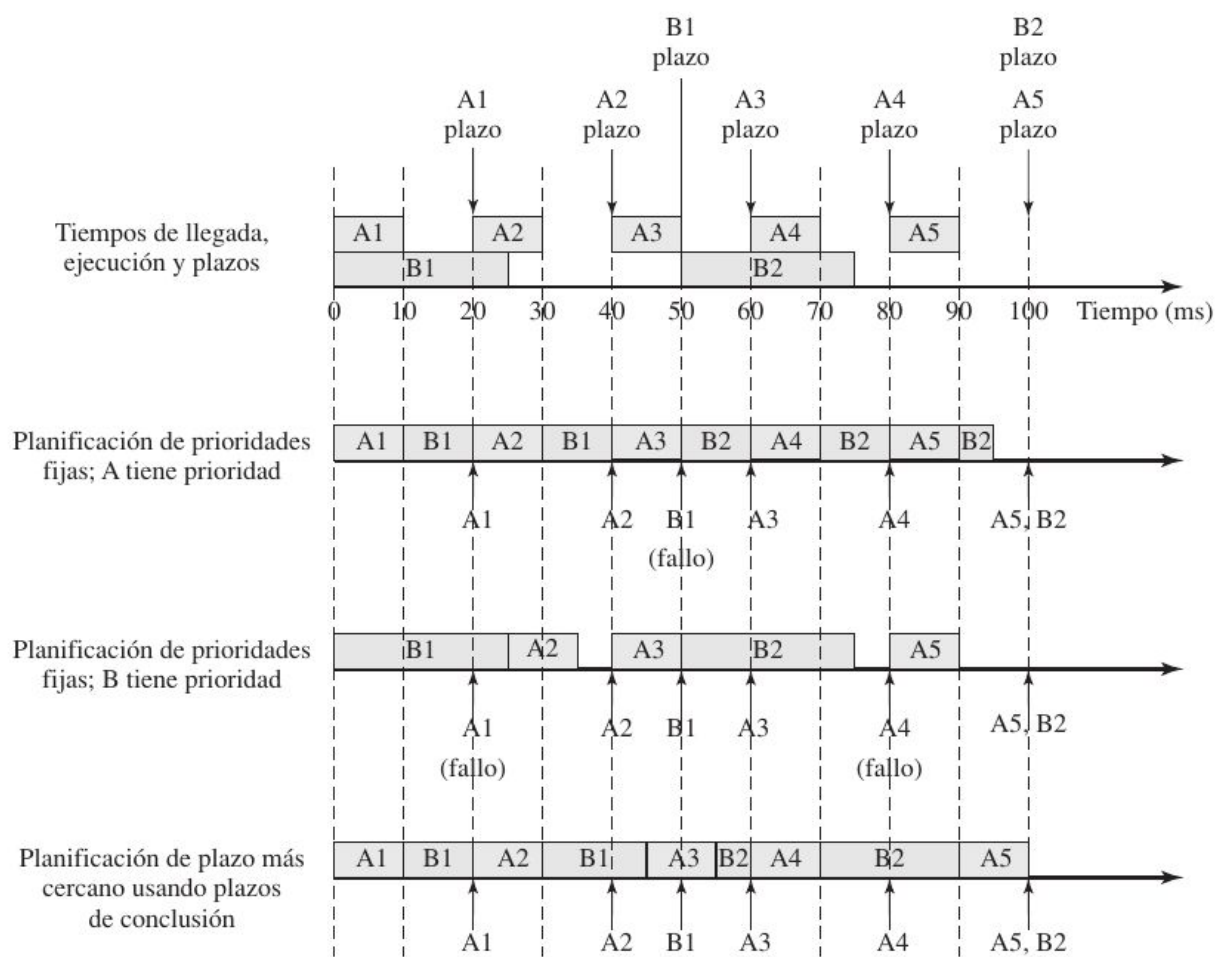
# Planificación por plazos

- En las aplicaciones de tiempo real no es importante la velocidad sino completar las tareas
- Información usada
  - Instante en que está lista
  - Plazo de comienzo
  - Plazo de terminación
  - Tiempo de procesamiento
  - Requisitos de recursos
  - Prioridad
  - Planificador de subtareas

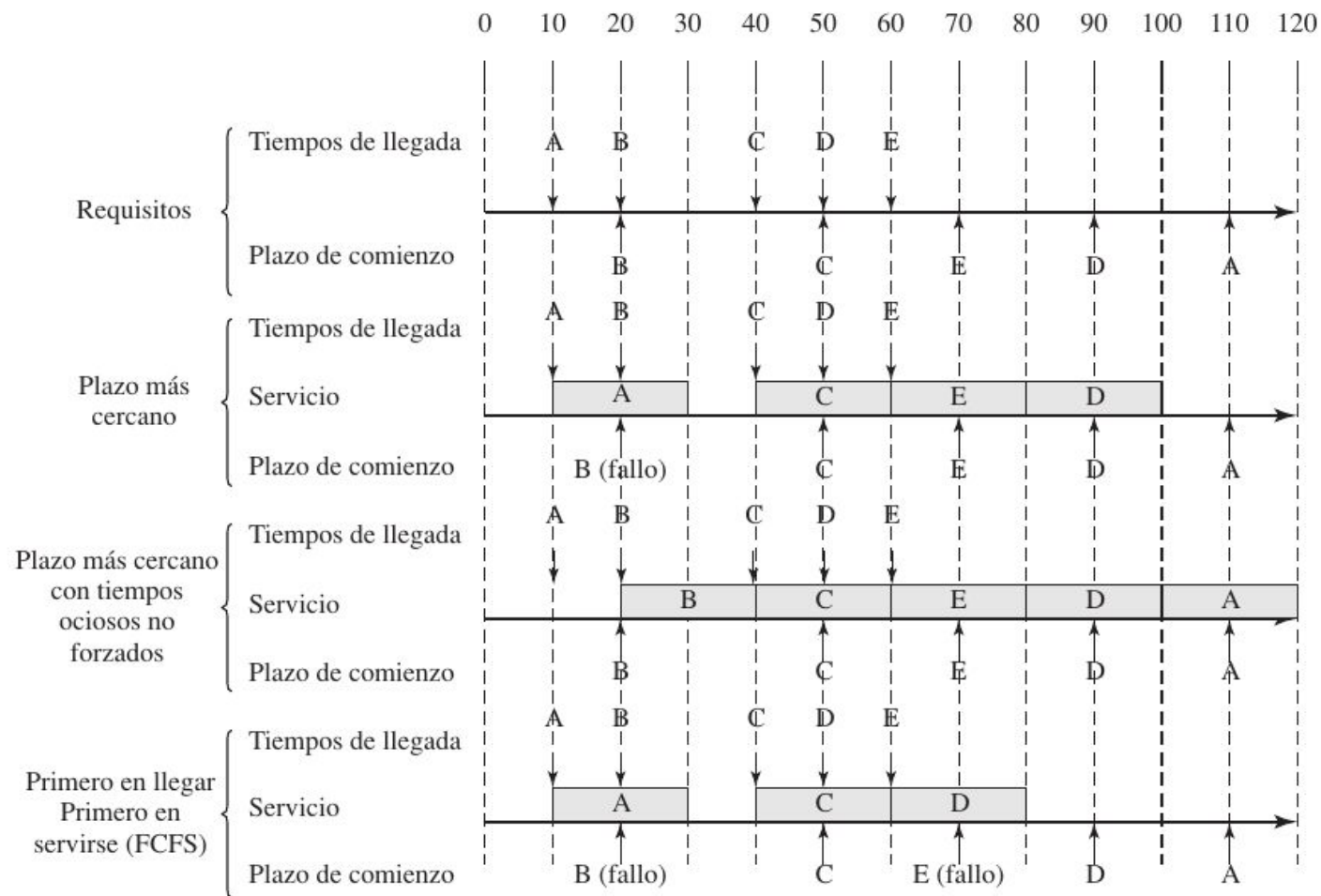
# Dos tareas

**Tabla 10.2.** Perfil de ejecución de dos tareas periódicas.

Proceso	Tiempo de llegada	Tiempo de ejecución	Plazo de conclusión
A(1)	0	10	20
A(2)	20	10	40
A(3)	40	10	60
A(4)	60	10	80
A(5)	80	10	100
•	•	•	•
•	•	•	•
•	•	•	•
B(1)	0	25	50
B(2)	50	25	100
•	•	•	•
•	•	•	•
•	•	•	•



**Figura 10.5.** Planificación de tareas periódicas de tiempo real con plazos de conclusión (basado en la Tabla 10.2).



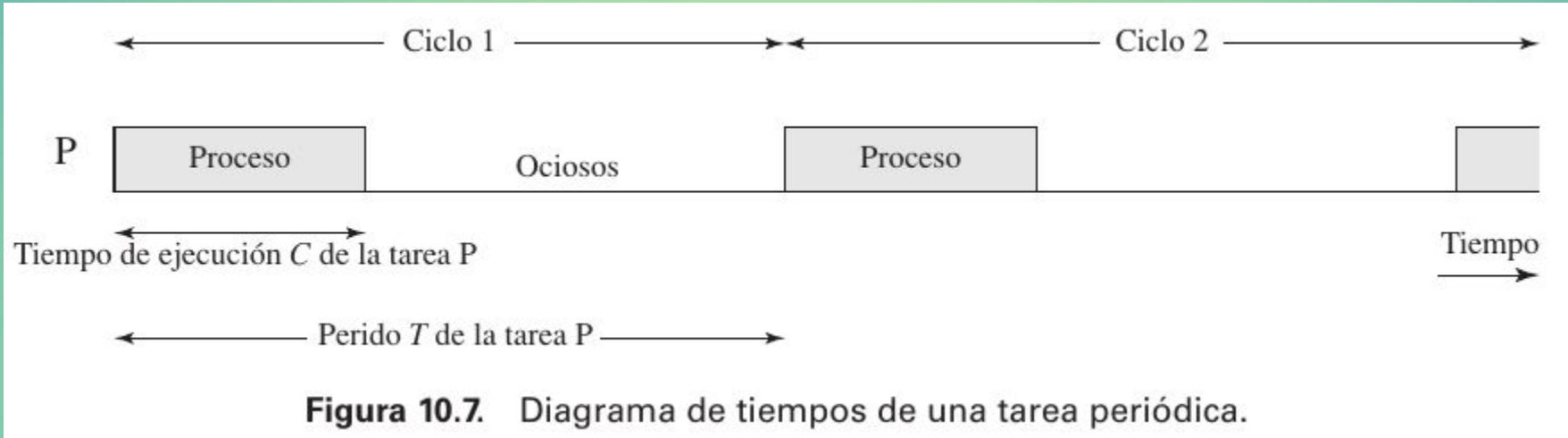
**Figura 10.6.** Planificación de tareas aperiódicas de tiempo real con plazos de comienzo.

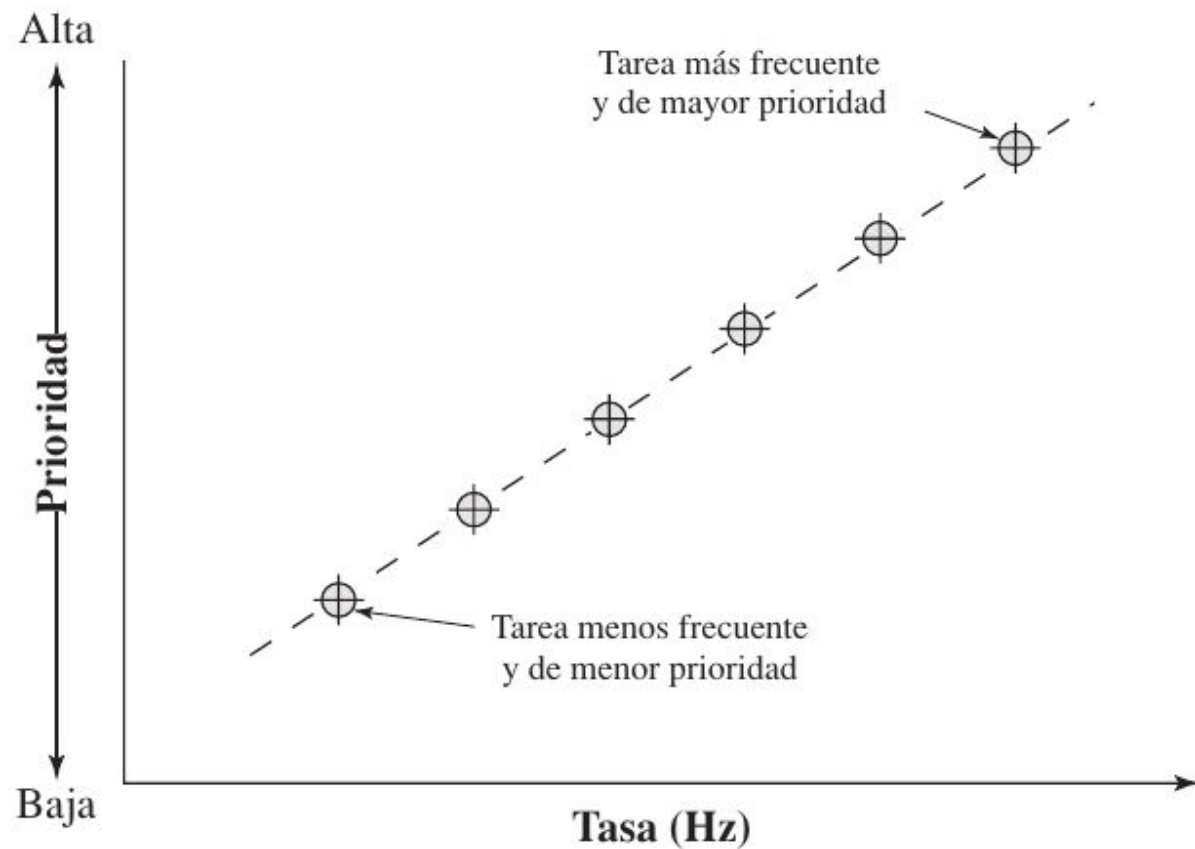
# Planificación monótona de frecuencia

- Asigna prioridades a tareas en base a sus periodos
- La tarea de mayor prioridad es la que tiene el periodo más corto



# Diagrama de tiempos de tareas periódicas





**Figura 10.8.** Un conjunto de tareas con RMS [WARR91].

# **Planificación sistemas ejemplo**

# Planificación Linux

- Clases planificación
  - Sched\_FIFO: Hilos tiempo real primero que entra primero que sale
  - SCHED\_RR: Hilos tiempo real round robin
  - SCHED\_OTHER: Otros, hilos tiempo no real
- Dentro de cada clase, se pueden usar múltiples prioridades

# Planificación Linux

A	Mínima
B	Media
C	Media
D	Máxima

D → B → C → A →

(a) Prioridades relativas de los hilos

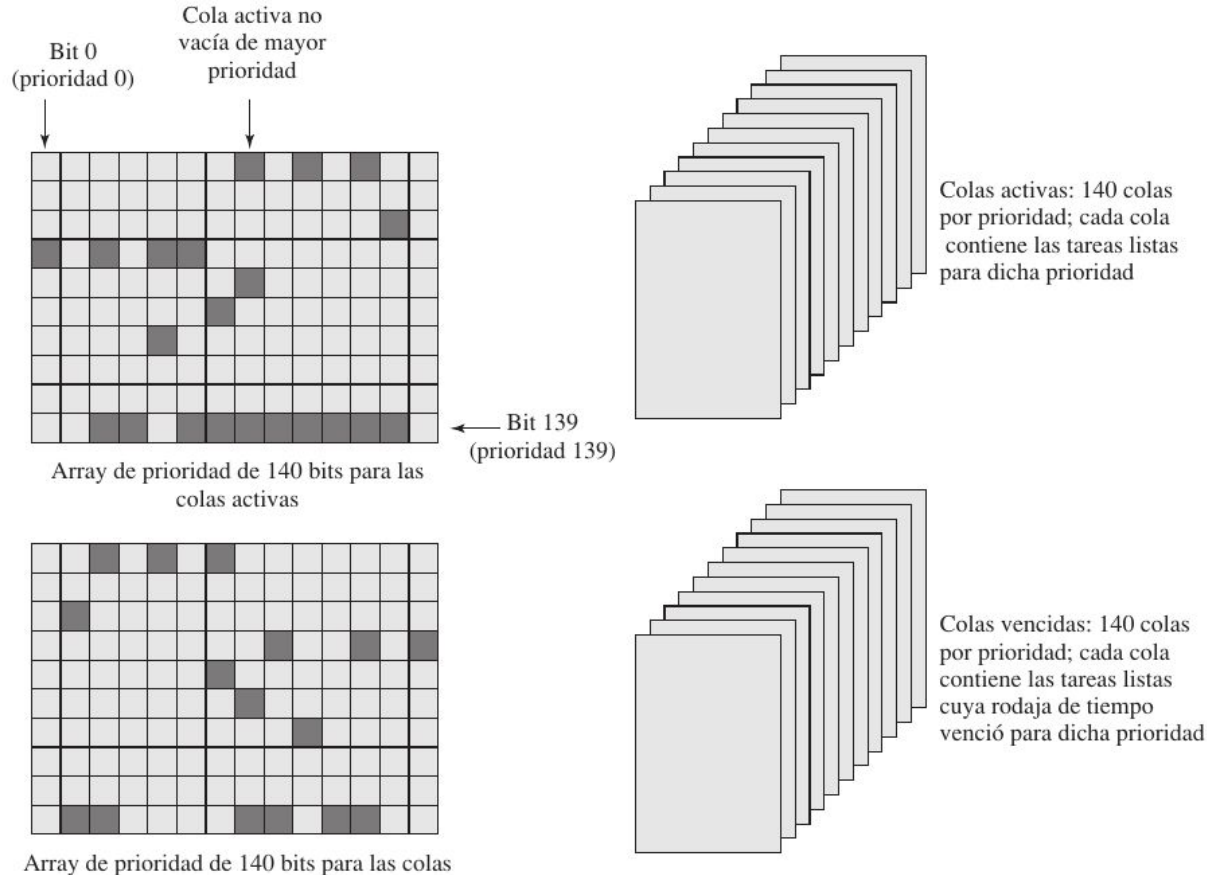
(b) Secuencia con planificación FIFO

D → B → C → B → C → A →

(c) Secuencia con planificación RR

**Figura 10.10.** Ejemplo de la planificación de tiempo real de Linux.

# Planificación Linux



**Figura 10.11.** Estructuras de datos para la planificación en Linux por cada procesador.

# Planificación UNIX SVR4

- **Orden decreciente de preferencias**
  - Procesos de tiempo real
  - Procesos modo kernel
  - Procesos modo usuario
- Colas de despacho



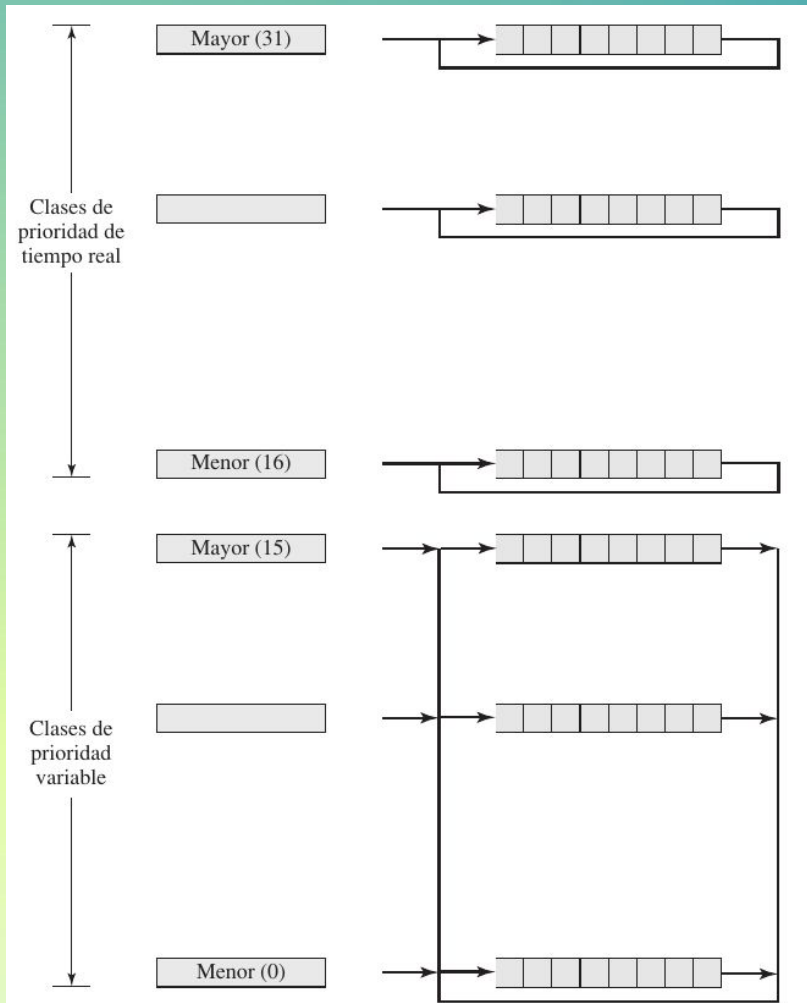
**Figura 10.13.** Colas de activación de SVR4.

# Planificación Windows

- Prioridades organizadas en dos bandas o clases
  - Tiempo real
  - Variable
- Planificador apropiativo manejado por prioridades

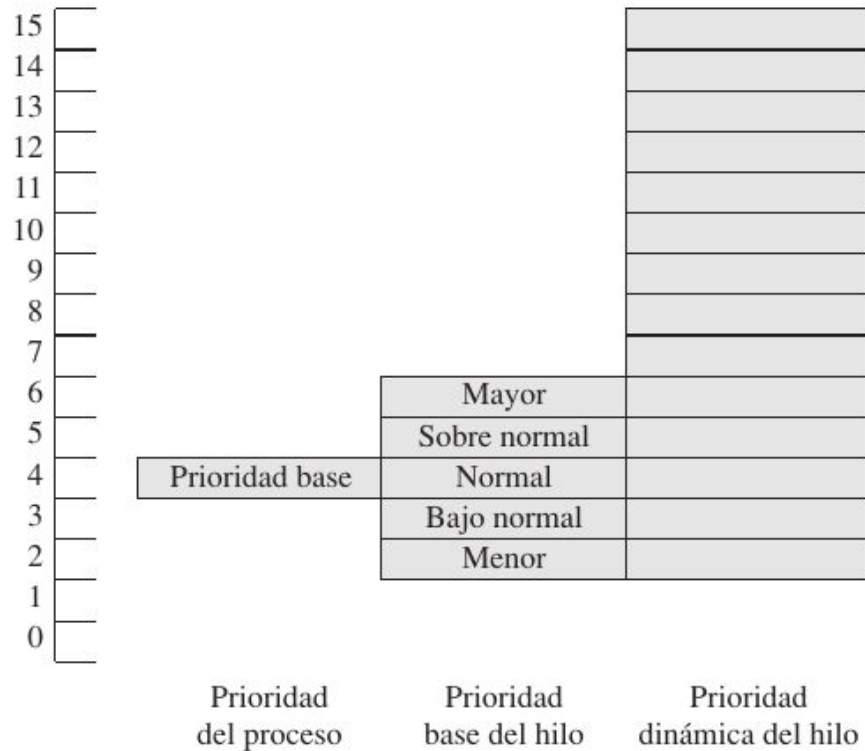


# Prioridades despacho hilos Windows



**Figura 10.14.** Prioridades de activación de hilos en Windows.

# Relación prioridades en Windows



**Figura 10.15.** Ejemplo de relación entre las prioridades de Windows.