

Sistemas Operativos

Planificación para monoprocesadores



U. ZARAGOZA



Planificación para monoprocesadores

- Tipos de Planificador
- Tipos de Sistema Operativo
- Criterios para el Planificador de corto
- Planificación por Prioridades
- Otras políticas de planificación
 - FCFS, RR, SPN, SRT, HRRN, Colas con Realimentación
- Planificación en UNIX

[Sta05]: capítulo 9

[SGG05]: capítulo 5



Tipos de Planificador

- Largo plazo
 - Decide sobre la creación de nuevos procesos, determina qué programas son admitidos por el sistema para ser procesados
 - Controla el grado de multiprogramación
 - Mayor número de procesos en el sistema implica menor porcentaje de tiempo dedicado a cada proceso
- Medio plazo
 - Swapping: intercambio de procesos entre memoria y disco
- Corto plazo
 - Se invoca cada vez que ocurre un evento
 - Interrupción de reloj, Interrupción de E/S, Llamada al sistema operativo, señales
 - Decide qué proceso de los preparados será el siguiente en ejecutarse



Planificador y estados de un proceso

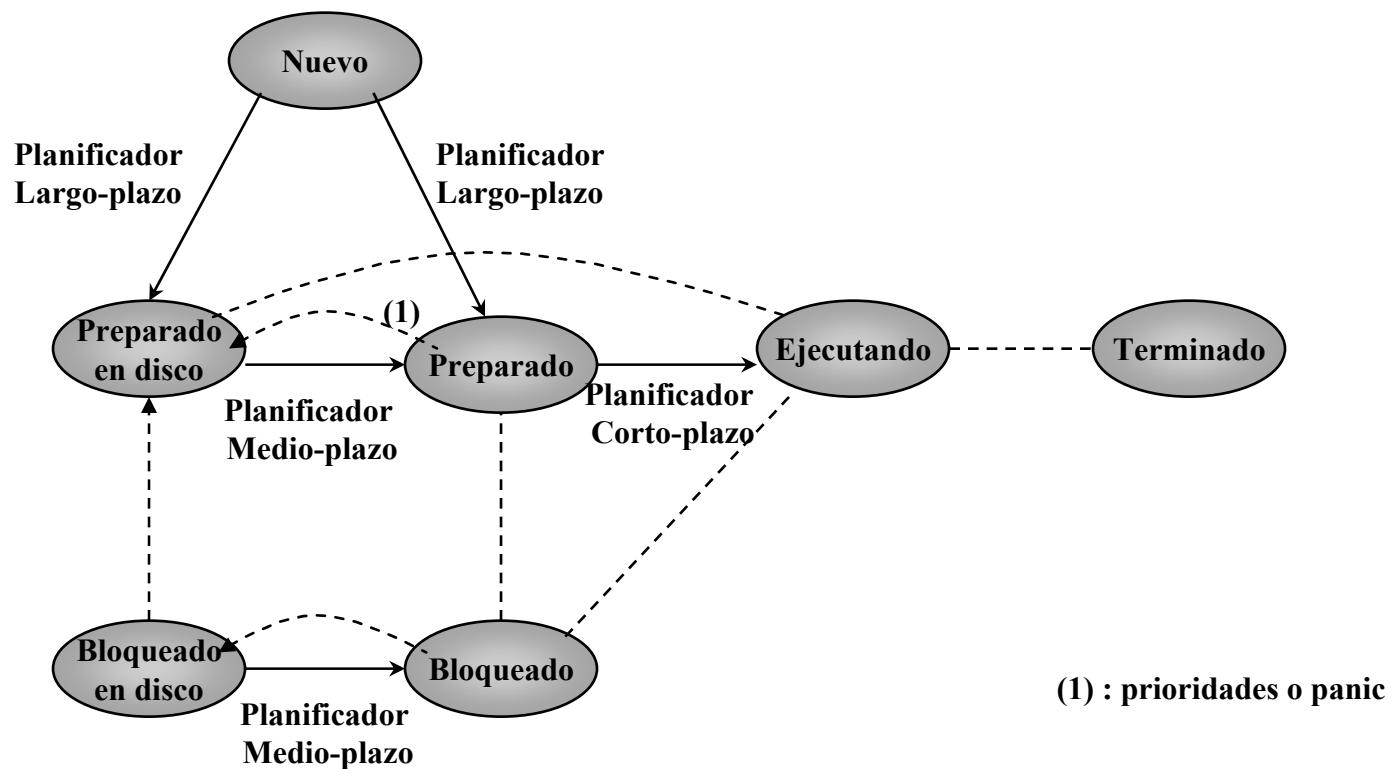
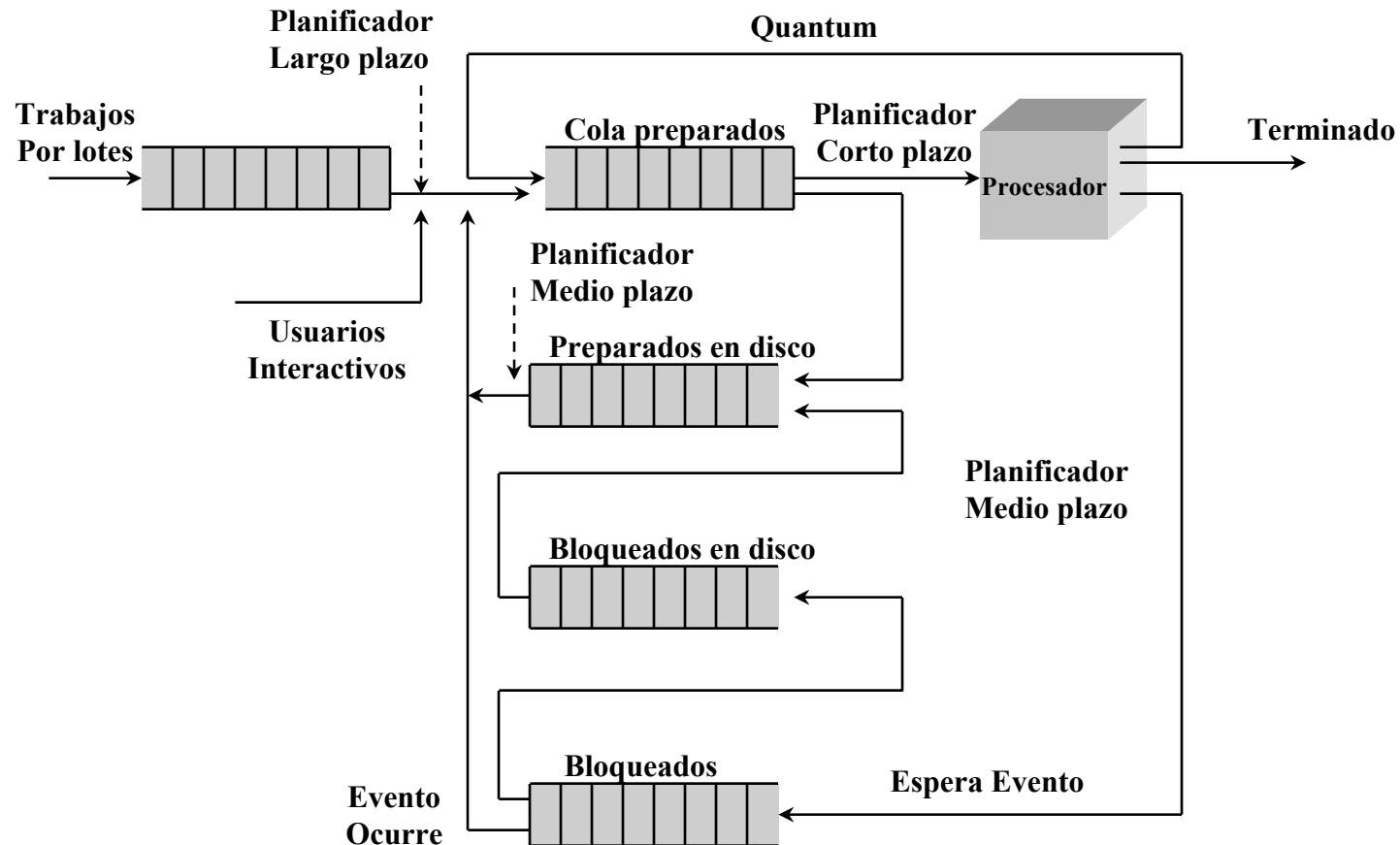


Diagrama de colas para la planificación



Tipos de Sistema Operativo

- SO por Lotes
 - Se agrupan programas, datos y órdenes de control para formar “trabajos” a ejecutar por el ordenador
 - No permiten interacción hombre - máquina
 - Buscan la máxima eficiencia del sistema
- SO de Tiempo Compartido
 - Pensados para entornos muy interactivos
 - El sistema pretende dar la imagen a cada usuario de que tiene toda la máquina para él
- SO de Tiempo Real
 - Entornos en los que han de ser aceptados y procesados un gran número de sucesos externos con mayor o menor urgencia
 - Control industrial, comunicaciones, control de vuelo, ...
- SO de Plazo Fijo
 - Ciertos trabajos tienen un tiempo específico para ser terminados. Se han de terminar antes de ese momento



Criterios para el Planificador de corto plazo

- Orientados al usuario
 - Tiempo de respuesta
 - Tiempo desde que se emite una petición hasta que se recibe la primera respuesta del sistema
 - Tiempo de retorno
 - Tiempo desde que se emite una petición hasta que se completa su servicio
 - Previsibilidad
 - El tiempo en que se ejecuta un trabajo debe ser independiente de la carga del sistema



Criterios para el Planificador de corto plazo

- Orientados al sistema
 - Productividad (throughput)
 - Maximizar el número de trabajos terminados por unidad de tiempo
 - Utilización del procesador
 - Maximizar el porcentaje de tiempo que el procesador está ocupado
 - Recursos equilibrados
 - Mantener ocupados todos los recursos del sistema
 - Favorecer los procesos que no usen recursos sobrecargados



Criterios para el Planificador de corto plazo

- Otros criterios
 - Plazos
 - Cada tarea tiene asociado un plazo de terminación
 - Prioridades
 - Cada tarea tiene asignada una prioridad
 - Equidad
 - En ausencia de otras directrices, todos los procesos deben tratarse de forma equitativa

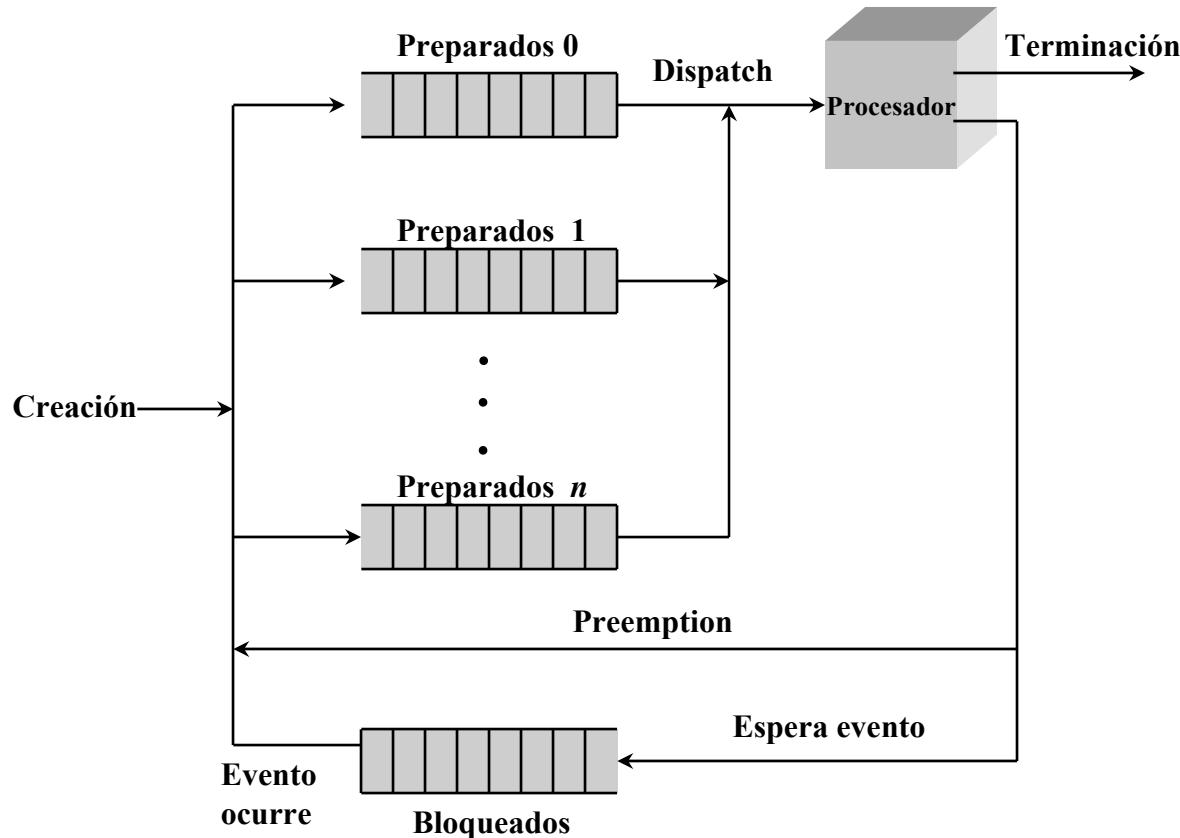


Planificación por Prioridades

- Cada proceso tiene asignada una prioridad
- El Planificador elige siempre el proceso con mayor prioridad entre los preparados
 - Implementación típica: una cola de preparados para cada prioridad
- Los procesos de baja prioridad pueden sufrir inanición
 - A veces los procesos cambian su prioridad en función de su historia en el sistema



Colas de prioridad



Asignación de prioridades

- Interna / Externa
 - Definida a partir de parámetros obtenidos por el propio SO
 - Definida por el administrador en base a tipo de usuario, importancia del trabajo, cuota que se paga, ...
- Estática / Dinámica
 - Estable durante toda la vida del proceso
 - Varía durante la ejecución en función del patrón de uso de recursos, de la ocupación del sistema, ...
- Se gana o se compra
 - El proceso gana o pierde prioridad dependiendo de su comportamiento
 - El proceso adquiere un cierto nivel de prioridad pagando por ello al propietario de la máquina



Modo de Decisión

- No preemption
 - Cuando un proceso consigue el procesador, no lo abandona hasta que termina o se bloquea en espera de una operación de E/S
- Preemption
 - Un proceso en estado de ejecución puede ser interrumpido por el Sistema Operativo, pasando a estado preparado
 - Evita que un proceso pueda monopolizar el procesador durante demasiado tiempo



Otras políticas de planificación

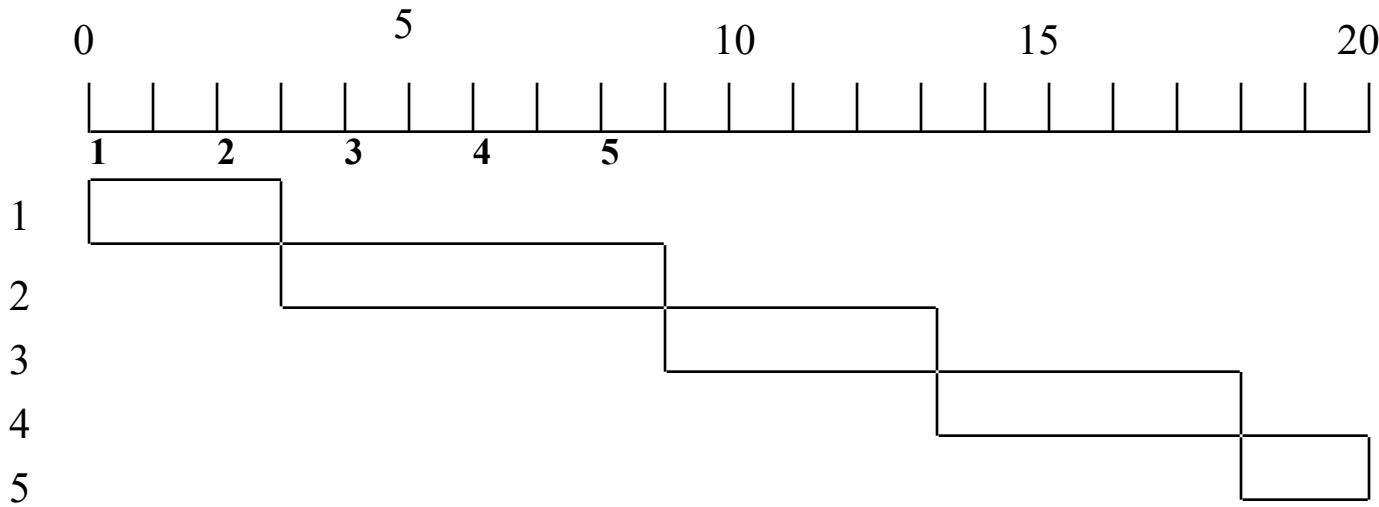
- First Come First Served (FCFS)
- Round-Robin (RR)
- Shortest Process Next (SPN)
- Shortest Remaining Time (SRT)
- Highest Response Ratio Next (HRRN)
- Colas con Realimentación

Ejemplo

Proceso	Tiempo de llegada	Tiempo de servicio
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2



First-Come-First-Served (FCFS)



- Cuando un proceso abandona la CPU, el proceso más viejo de la cola de Preparados pasa a ejecutarse

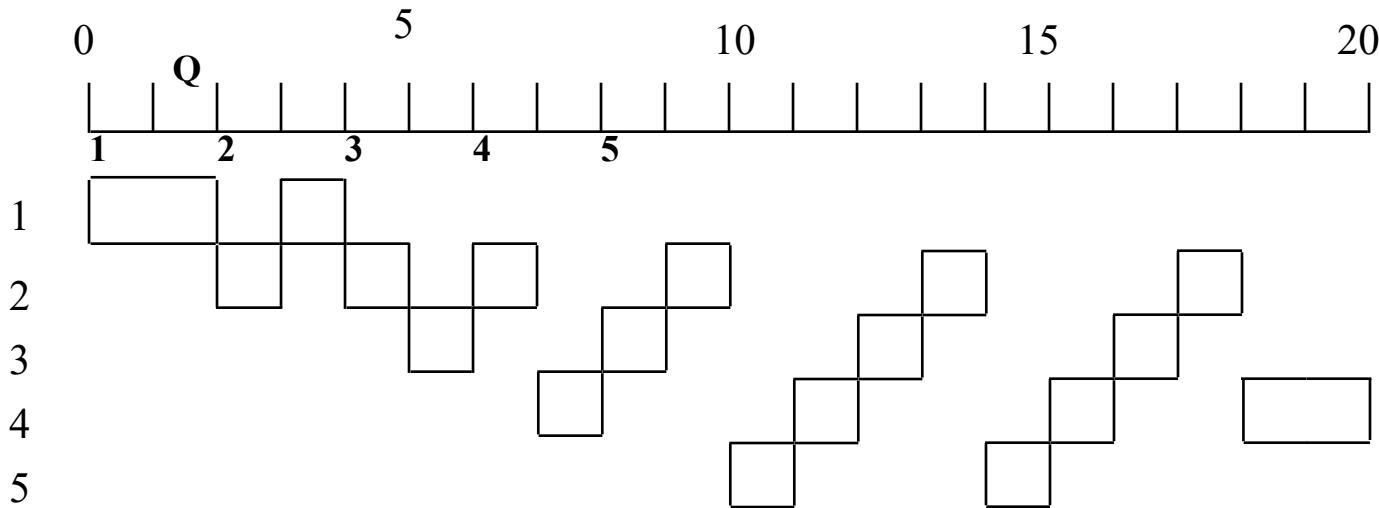


First-Come-First-Served (FCFS)

- Un proceso corto puede ser obligado a esperar durante mucho tiempo antes de entrar a ejecutarse
- Favorece a los procesos CPU-bound
 - Los procesos con mucha E/S deben esperar hasta que los CPU-bound abandonan el procesador
- Minimiza el número de cambios de contexto



Round-Robin



- Usa preemption basada en un reloj
- Quantum (Q): tiempo máximo de permanencia en ejecución que se da a un proceso

Round Robin: tamaño del Quantum

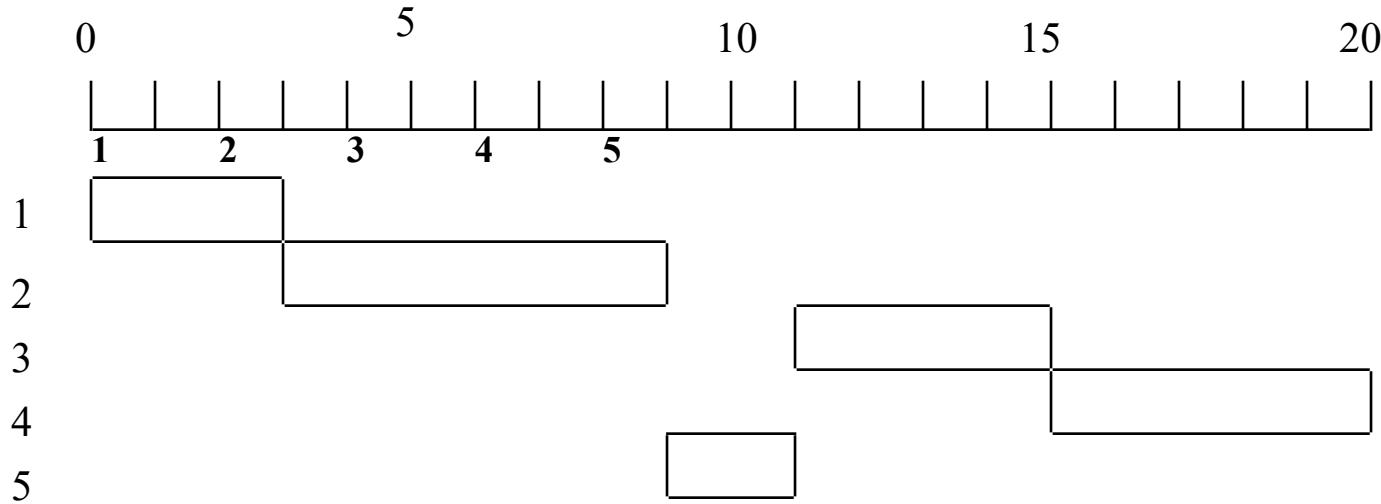
- Mayor quantum, mejor throughput (pensar por qué)
 - Caso extremo: quantum infinito. Óptimo en supercomputación...
- Menor quantum, mejor tiempo de respuesta (pensar por qué)
- En general, más cambios de contexto = menor rendimiento = mayor satisfacción de los usuarios en tiempo compartido
- Porcentaje de tiempo perdido en Cambio de Contexto

$$\%Tcc = 100 * \frac{Tcc}{Tcc + Q} \quad (\text{Mínimo})$$

- Tcc disminuye cuando aumentan prestaciones
- Si Q constante => %Tcc disminuye, tiende a cero
- Se puede disminuir Q para mejorar el servicio sin aumentar la pérdida de prestaciones



Shortest Process Next



- Política no preemption
- Se selecciona al proceso con menor tiempo en ejecución previsto
- Los procesos cortos se adelantan a los largos

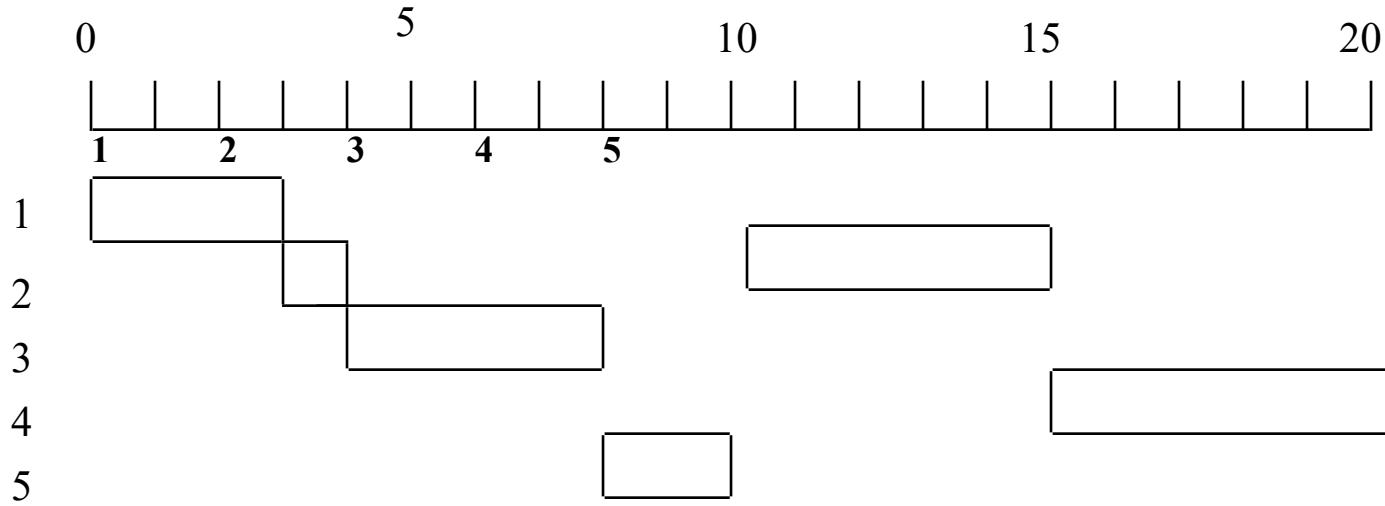


Shortest Process Next

- Minimiza el tiempo medio de espera (óptimo)
- El comportamiento de los procesos largos es menos previsible
- Los procesos largos pueden sufrir inanición
- Debe estimar el tiempo de proceso
- Si el tiempo estimado para un proceso no es el correcto, el SO puede abortarlo



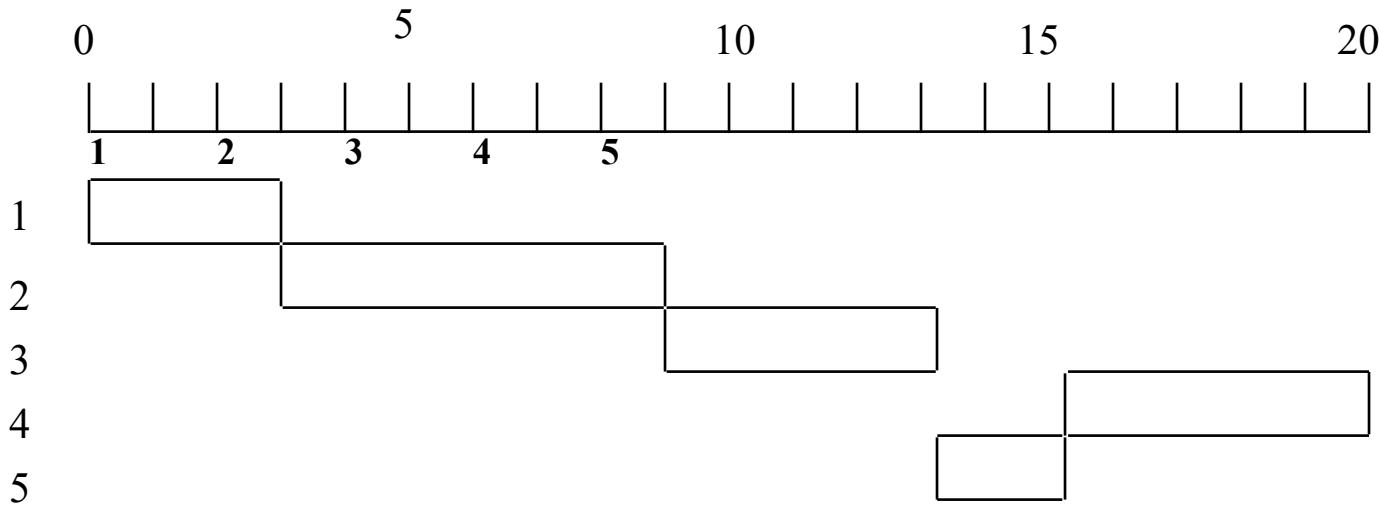
Shortest Remaining Time



- Versión preemptive de la política shortest process next
- Debe estimar el tiempo de proceso
- Posible inanición de procesos largos



Highest Response Ratio Next (HRRN)



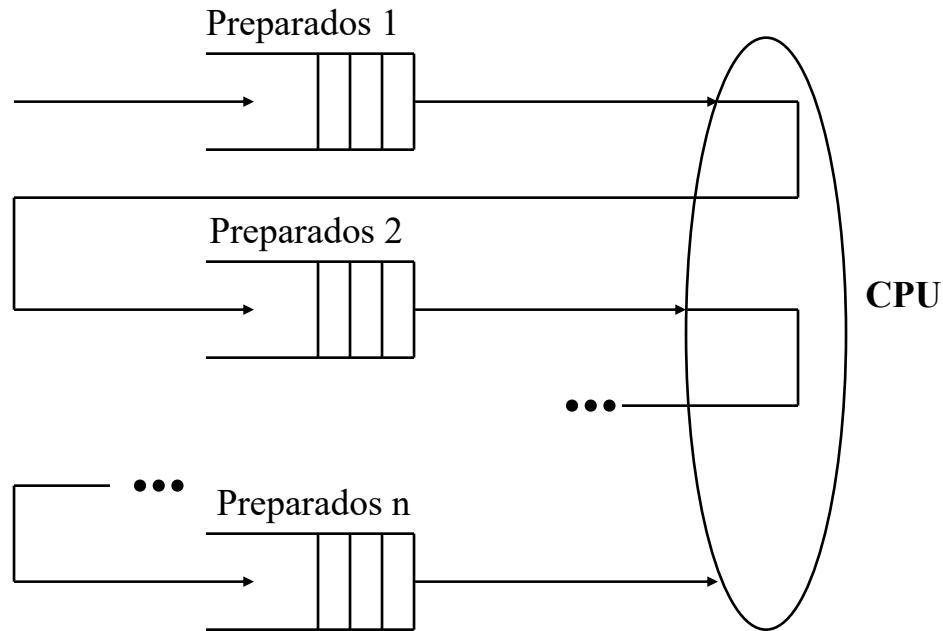
- Elige el siguiente proceso según mayor ratio

$$\frac{\text{Tiempo de espera en las colas} + \text{tiempo de servicio}}{\text{tiempo de servicio}}$$

- No preemptive
- Resuelve problemas inanición



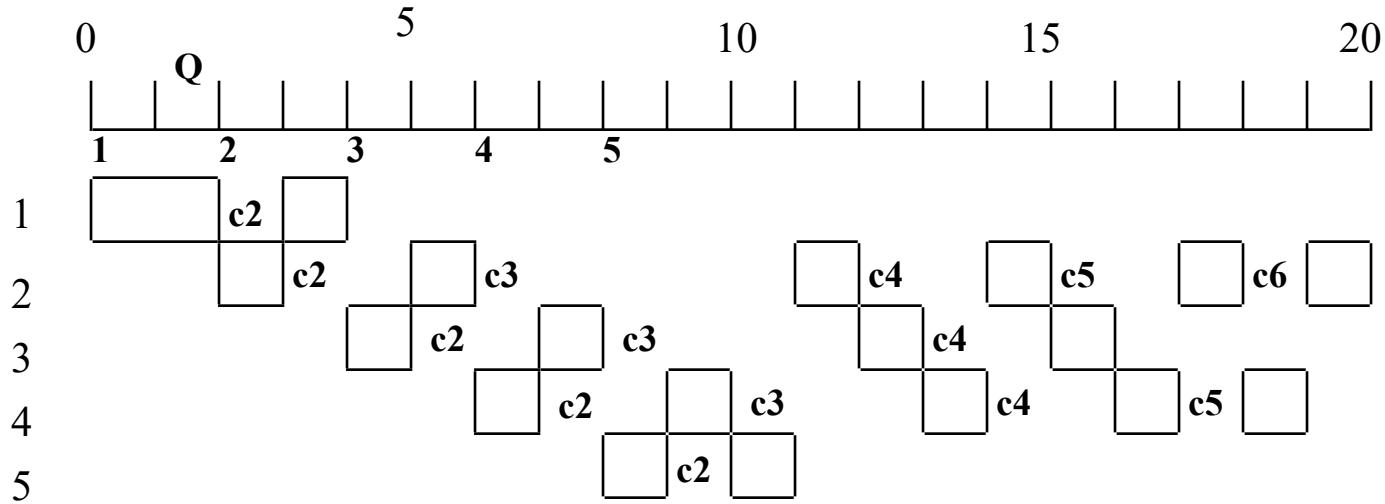
Colas con realimentación



- Varias colas con distinta prioridad
 - Los procesos pasan de unas a otras en función de su comportamiento temporal



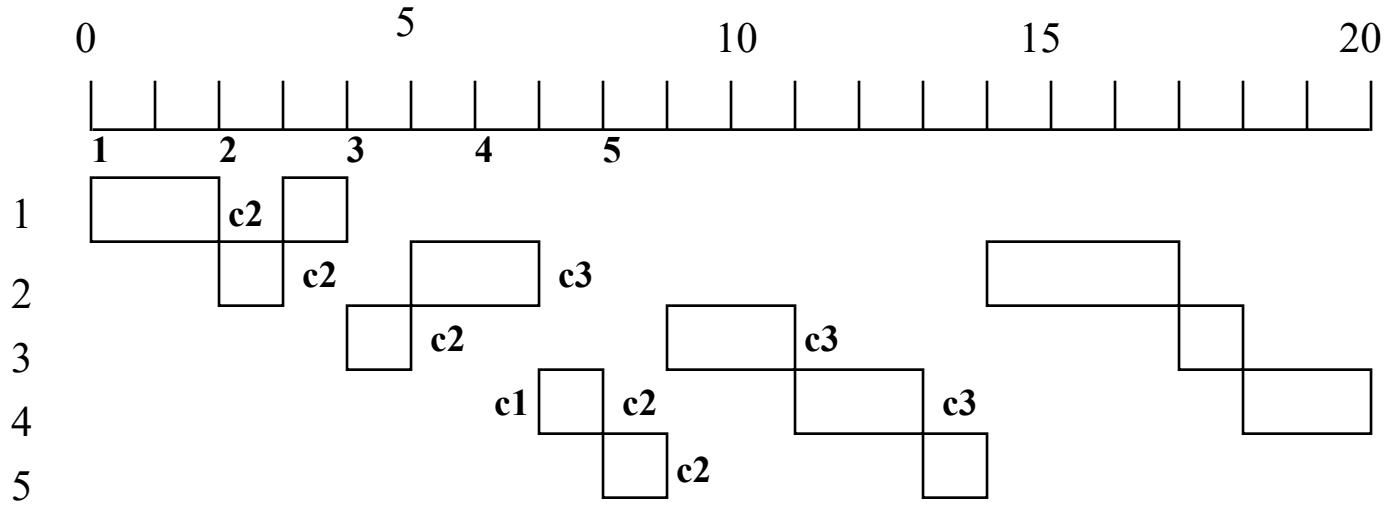
Colas con realimentación



- Penaliza a los procesos que más tiempo llevan en ejecución



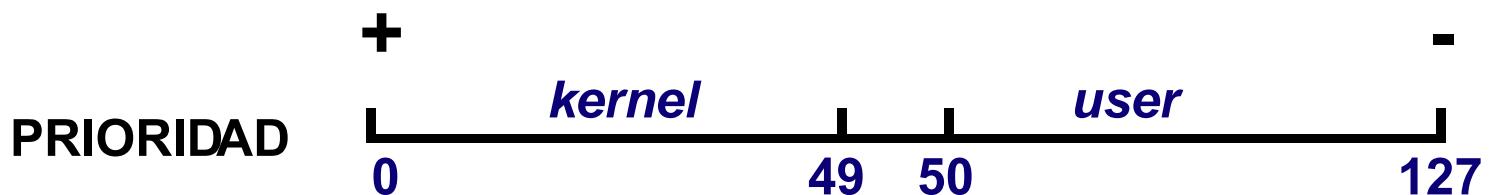
Colas con realimentación



- Quantum variable según cola
 - $Q(c1)=1, Q(c2)=2, \dots$

Planificación en UNIX

- Prioridad variable + Round Robin
- Cálculo de prioridad
 - Valor numérico: A mayor valor menor prioridad



- Prioridad base + NICE + prioridad variable
- *Prioridad base* de usuario: 50
- *NICE*: valor positivo que el usuario puede añadir a sus programas para “molestar menos”
- *Prioridad variable*: función de la carga del sistema y del uso de CPU que el proceso ha hecho recientemente



Planificación en Solaris (hendrix)

- Planificación basada en prioridades
- Seis clases de procesos (prioridad de + a -)
 - Tiempo real
 - Sistema
 - Parte justa (fair share)
 - Prioridad fija
 - Tiempo compartido (por defecto)
 - Interactivo
- Cada proceso está en una única clase en un momento dado
- Cada clase tiene su propio algoritmo de planificación
- Tiempo compartido es una cola multinivel con realimentacion (configurable por el administrador)

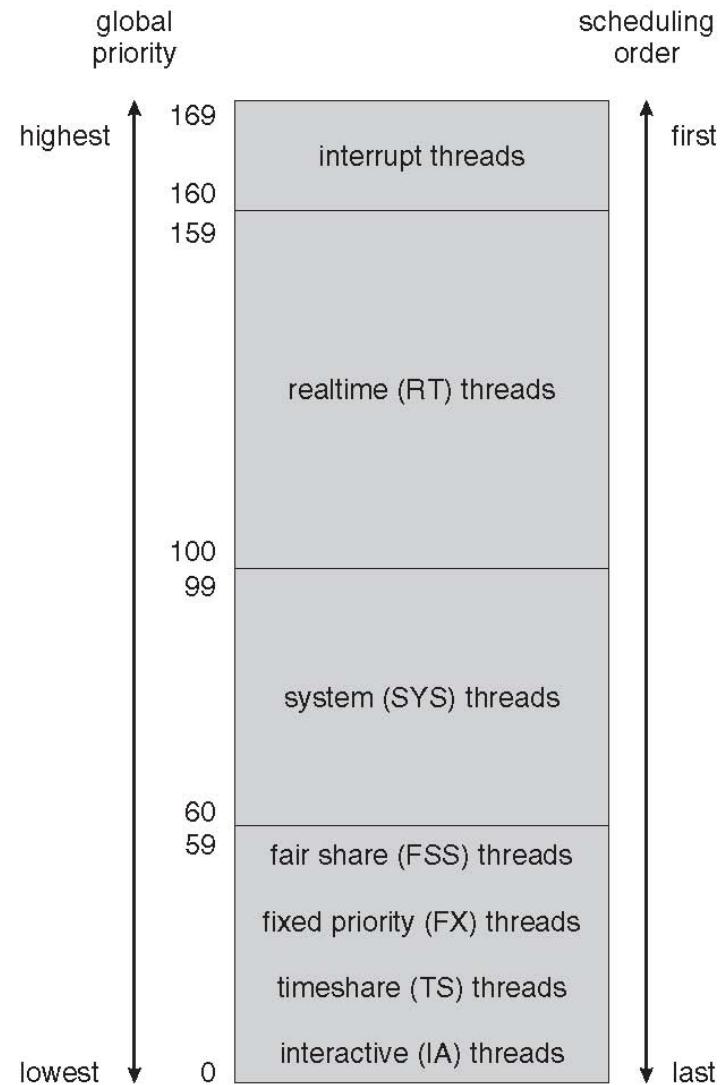


Solaris Dispatch Table

priority	time quantum	time quantum expired	return from sleep
0	200	0	50
5	200	0	50
10	160	0	51
15	160	5	51
20	120	10	52
25	120	15	52
30	80	20	53
35	80	25	54
40	40	30	55
45	40	35	56
50	40	40	58
55	40	45	58
59	20	49	59



Planificación en Solaris



Planificación en Solaris (Cont.)

- El planificador convierte las prioridades particulares de cada clase en una prioridad global por proceso
 - El proceso con más prioridad es el siguiente en ejecutarse
 - Se ejecuta hasta (1) bloqueo, (2) quantum, (3) preempted por un proceso más prioritario
 - Si hay varios procesos de la misma prioridad, se selecciona por RR

