

# Tema IV

## Planificación de Procesos





# Planificador de Procesos

Un planificador de procesos define qué procesos tendrán derecho al CPU, cuándo y cuánto tiempo.

La ejecución de un proceso consiste en una alternancia entre:

- **Ráfagas de CPU.** El proceso ejecuta instrucciones.
- **Ráfagas de E/S.** El proceso utiliza o espera por e/s.



# Planificador de Procesos





# Tipos de planificación

- **Planificación apropiativa**

Interrumpe el procesamiento de un proceso y transfiere el CPU a otro.

- **Planificación no apropiativa**

Una vez que un proceso captura el procesador e inicia la ejecución, se mantiene en el estado de ejecución hasta que emite una solicitud de e/s o hasta que termina.



# Criterios de planificación

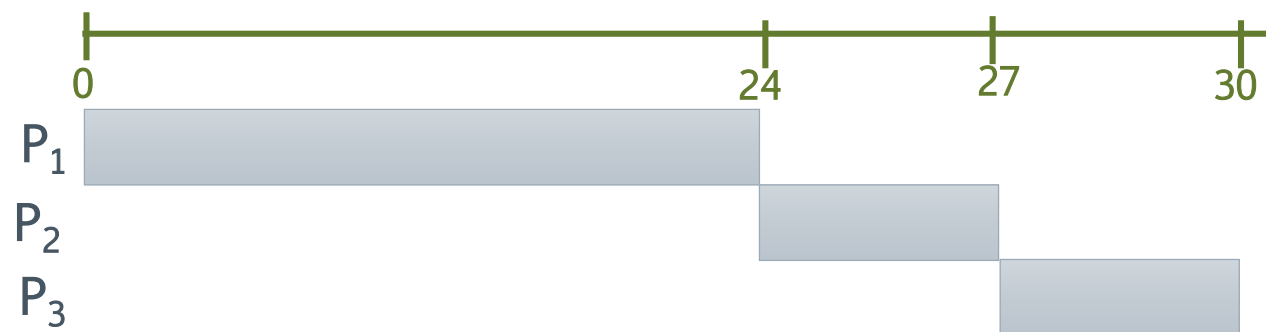
- **Tiempo de retorno:** Conocido también como *tiempo de entrega*. Es el tiempo transcurrido desde que se presenta el proceso hasta que se completa.
- **Tiempo de espera:** Tiempo que un proceso pasa en la cola de procesos listos esperando tiempo de CPU.
- **Tiempo de respuesta:** Tiempo en que se presenta la primera respuesta desde la solicitud del proceso.



# Algoritmo First-Come, First-Served (FCFS)

Procesos	Ráfaga de CPU [ms]
P1	24
P2	3
P3	3

Los procesos llegan en el orden: P1 , P2 , P3 . La planificación es:



Tiempo de espera para P1 = 0; P2 = 24; P3 = 27

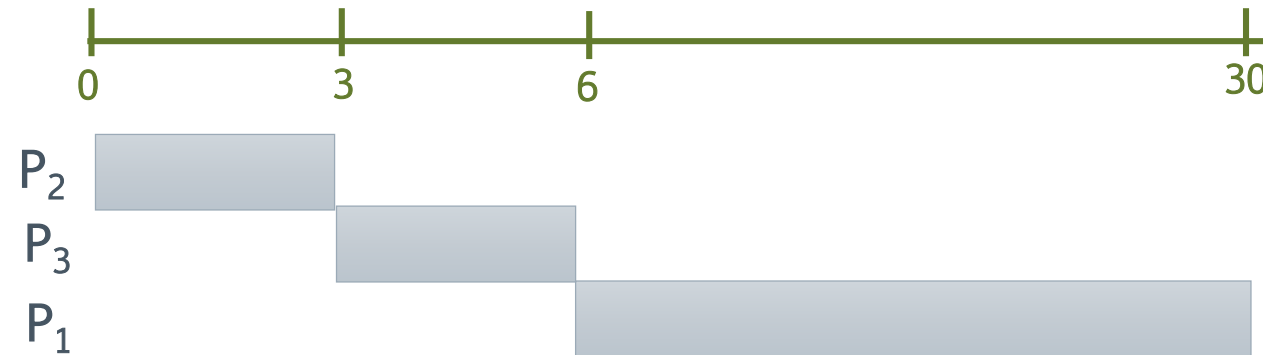
Tiempo de espera medio:  $(0 + 24 + 27)/3 = 17$



# Algoritmo First-Come, First-Served (FCFS)

Procesos	Ráfaga de CPU [ms]
P2	3
P3	3
P1	24

Si cambiamos el orden de llegada de los procesos: P2 , P3, P1. La planificación es:



Tiempo de espera para  $P_1 = 6$ ;  $P_2 = 0$ ;  $P_3 = 3$  Tiempo medio de espera:  $(6 + 0 + 3)/3 = 3$

Con este algoritmo se puede producir un **efecto convoy**: varios procesos de ráfaga de CPU corta tienen que esperar a un proceso de ráfaga larga.



# Algoritmo

## *Shortest Job Next* (SJN)

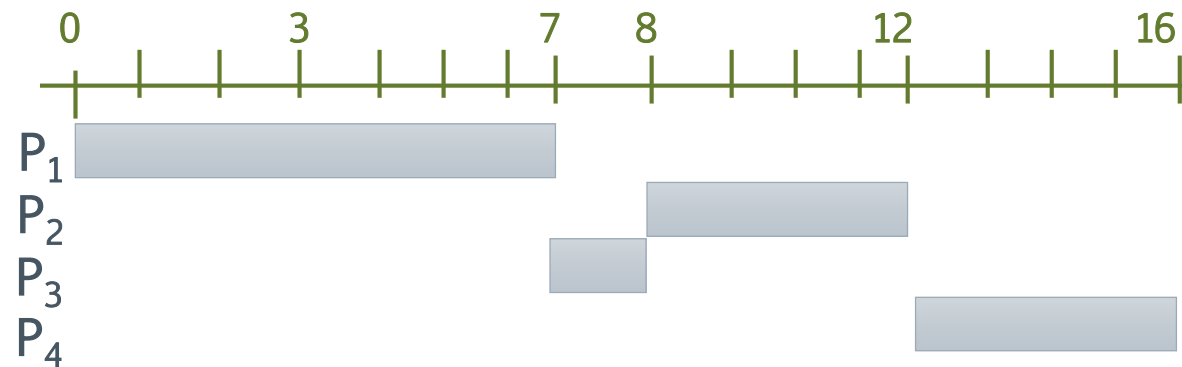
- Conocido también como **SJF** (*Shortest Job First*).
- Asigna tiempo de CPU al proceso cuya siguiente ráfaga de CPU es más corta. Si dos procesos empatan se resuelve por FCFS.
- No apropiativo.





# Algoritmo *Shortest Job Next* (SJN)

Procesos	Tiempo de llegada [ms]	Ráfaga de CPU [ms]
P1	0	7
P2	2	4
P3	4	1
P4	5	4



$$\text{Tiempo de espera medio} = (0 + 6 + 3 + 7)/4 = 4$$



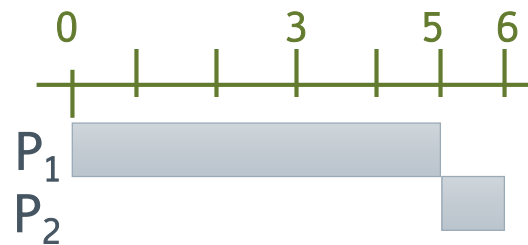
# Algoritmo Planificación con Prioridad

- Se asocia con cada proceso una prioridad (número entero).
- El CPU se asigna al proceso con la prioridad más alta (consideramos número pequeño  $\equiv$  prioridad alta).
- No apropiativo y apropiativo.
- **Problema:** Inanición (*starvation*): Los procesos de más baja prioridad podrían no ejecutarse nunca.
- **Solución:** Envejecimiento (*aging*): Conforme el tiempo pasa, se aumenta la prioridad de los procesos que esperan mucho en el sistema.



# Algoritmo Prioridad no apropiativo

Procesos	Tiempo de llegada [ms]	Ráfaga de CPU [ms]	Prioridad
P1	0	5	4
P2	3	1	1

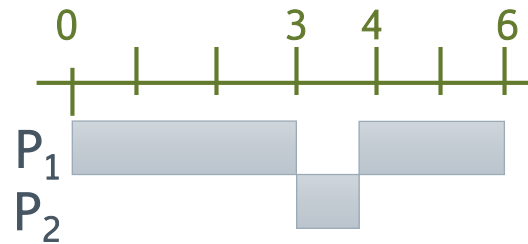


Procesos	Tiempo de espera [ms]	Tiempo de retorno [ms]
P1	0	5
P2	2	3



# Algoritmo Prioridad apropiativo

Procesos	Tiempo de llegada [ms]	Ráfaga de CPU [ms]	Prioridad
P1	0	5	4
P2	3	1	1



Procesos	Tiempo de espera [ms]	Tiempo de retorno [ms]
P1	1	6
P2	0	1

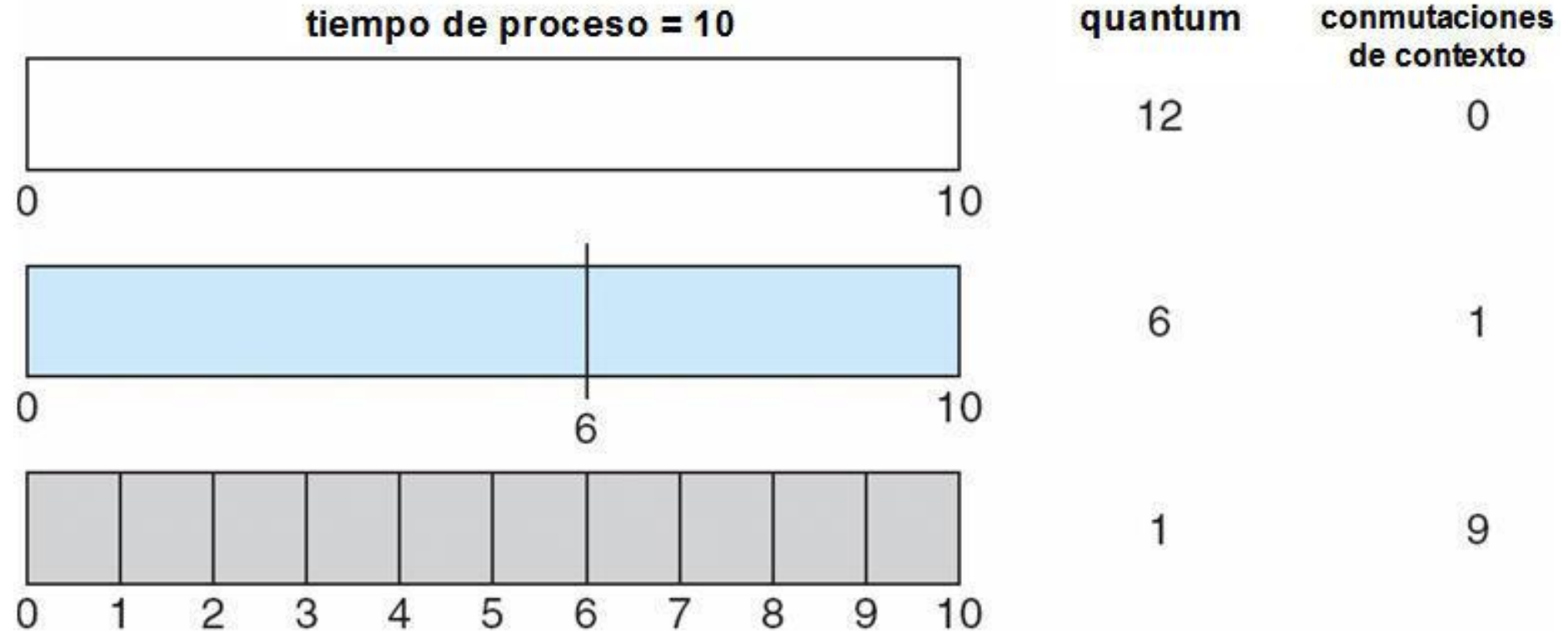


# Algoritmo Round Robin

- Cada proceso obtiene tiempo de CPU durante un **quantum de tiempo** (normalmente de 10 a 100 milisegundos). Cuando el quantum termina, el proceso es expulsado e insertado al final de la cola de listos.
- Algoritmo **apropiativo**.
- Requiere seleccionar un buen **quantum de tiempo**.



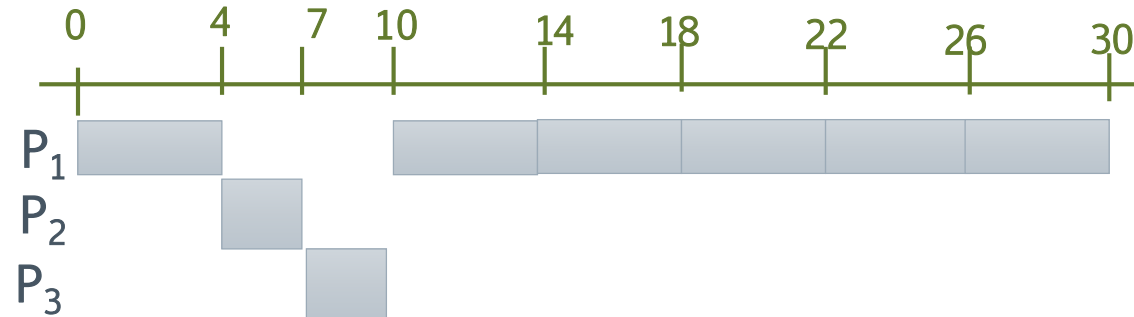
# Quantum y conmutación de contexto





# Algoritmo Round Robin con $q = 4$

Procesos	Ráfaga de CPU [ms]	Tiempo de llegada [ms]
P1	24	0
P2	3	0
P3	3	0

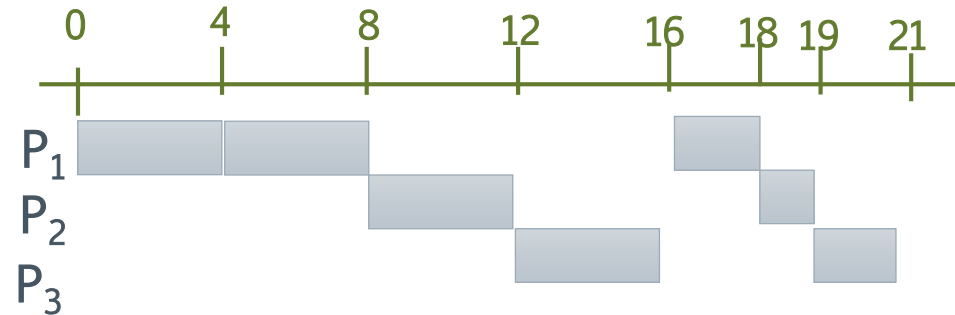


Procesos	Tiempo de espera [ms]	Tiempo de respuesta [ms]	Tiempo de retorno [ms]
P1	6	0	30
P2	4	4	7
P3	7	7	10



# Algoritmo Round Robin con $q = 4$

Procesos	Ráfaga de CPU [ms]	Tiempo de llegada [ms]
P1	10	0
P2	5	5
P3	6	5



Procesos	Tiempo de espera [ms]	Tiempo de respuesta [ms]	Tiempo de retorno [ms]
P1	8	0	18
P2	9	8	14
P3	10	12	16



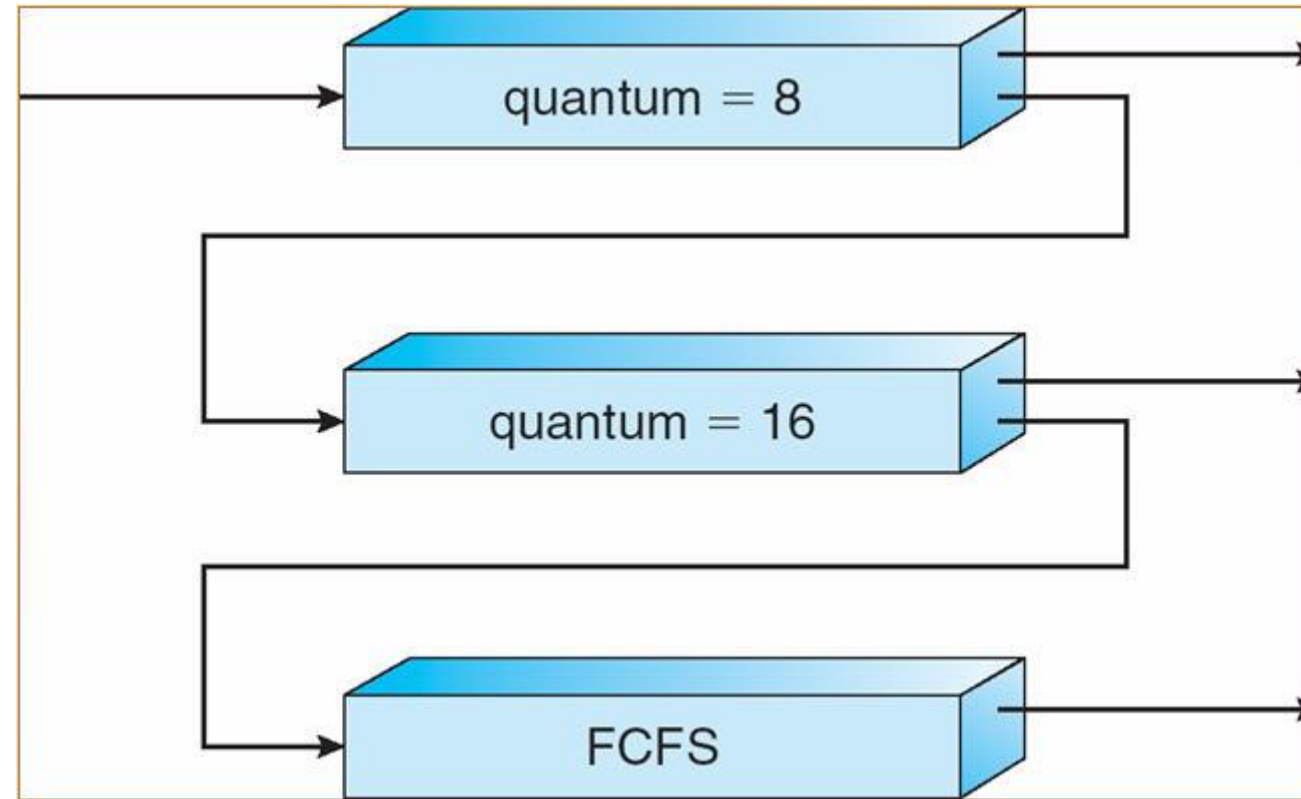


# Colas Multinivel con Realimentación

- Esquema flexible.
- Sobrecarga incurrida por revisión constante de colas.
- Se opone al aplazamiento indefinido con el envejecimiento u otro movimiento en cola.
- Da un tratamiento adecuado a los procesos asignados al CPU por incremento de quantum sobre colas de baja prioridad u otro tipo de colas.



# Colas Multinivel con Realimentación





# SRT (*Shortest Remaining Time*)

- El proceso con el tiempo estimado de ejecución menor para llegar a su terminación, es el siguiente en ser ejecutado, incluyendo las nuevas llegadas.
- Apropiativo.
- Útil en tiempo compartido.

[illegible]



# HRRN (*Highest Response Ratio Next*)

- Brinch Hansen desarrolló esta estrategia: Primero el de mayor tasa de respuesta.
- Corrige las debilidades del SJN.
- Exceso de prejuicio hacia los procesos largos y exceso de favoritismo hacia los nuevos procesos cortos.
- Los procesos más cortos recibirán preferencia, y a su vez los procesos más largos que han estado esperando recibirán también un tratamiento favorable.
- Es no apropiativo.
- La prioridad de cada proceso esta en función del tiempo de servicio del proceso y de la cantidad de tiempo que el proceso ha estado esperando ser servido.



# HRRN (*Highest Response Ratio Next*)

- La tasa de respuesta se calcula con la formula:

$$R = (w + s) / s$$

R = tasa de respuesta

w = tiempo consumido esperando al procesador

s = tiempo de servicio esperado

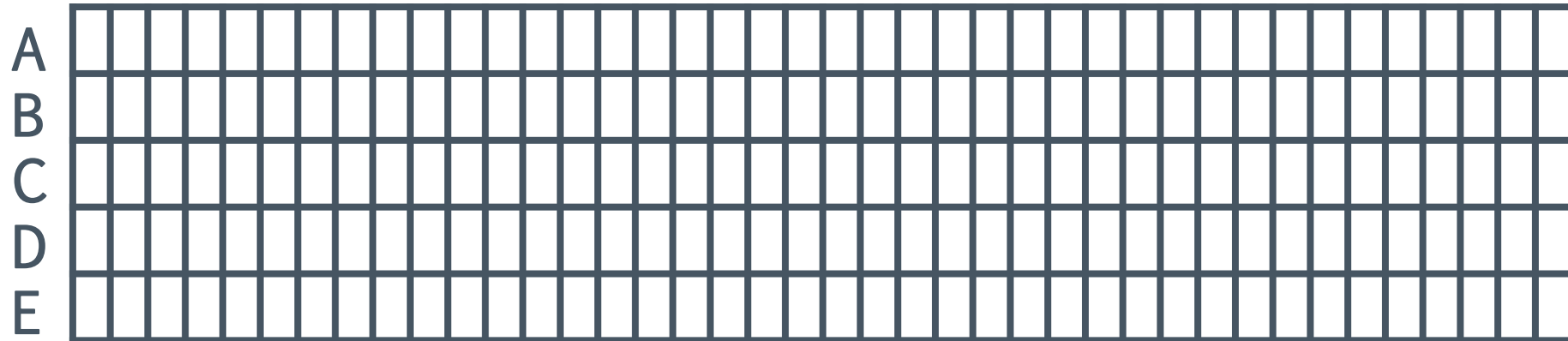
- Cabe señalar que la suma del Tiempo de respuesta + Tiempo de servicio es el Tiempo de respuesta del sistema para el proceso, si éste se inicia de inmediato.

[illegible]



# Ejemplo con SJN

Proceso	Instante de llegada [ms]	Ráfaga de CPU [ms]	E/S	Ráfaga de CPU [ms]
A	0	3	2	2
B	2	6	-	-
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2



En ejecución



Listo



E/S (Bloqueado)





Ing. Yesenia Carrera Fournier  
sofiunam at gmail dot com