

FCFS (First Come First Served)

(FIFO)

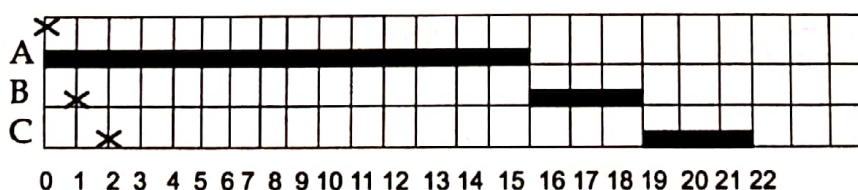
- Los procesos son servidos según el orden de llegada a la cola de ejecutables. / preparados.
- Es **no apropiativo**, cada proceso se ejecutará hasta que finalice o se bloquee.
- Fácil de implementar pero pobre en cuanto a prestaciones.
(Porque el tiempo de respuesta de los procesos va a depender de los procesos que haya delante de él en la cola de preparados. Si p.ej. delante hay un proceso largo, tendrá que esperar mucho tiempo).
- Todos los procesos pierden la misma cantidad de tiempo esperando en la cola de ejecutables/preparados independientemente de sus necesidades.
(Es decir, el tiempo que espera ahí no depende del mayor o menor tiempo de CPU que éste requiera, sino de las necesidades de CPU que requieren los procesos que están delante de él).
- Procesos cortos muy penalizados. (Porque en relación al pequeño tiempo de CPU que ellos necesitan, normalmente van a tener un tiempo de respuesta alto)
- Procesos largos poco penalizados.

Procesos	Tº llegada	Ráfaga	T(tº respuesta)	M(tº espera)
A	0	15	$15-0=15$	0
B	1	3	$18-1=17$	14
C	2	3	$21-2=19$	16

*cada fila un proceso

*cada columna un instante de tiempo de ejecución

Diagrama de ocupación de la CPU

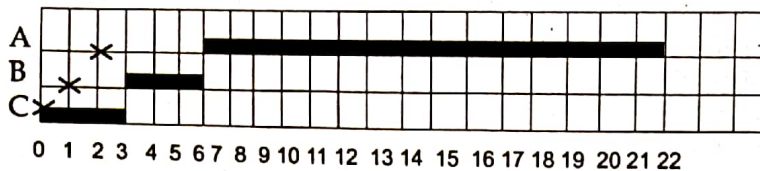


Tº DE ESPERA: tiempo que están en la cola de preparados antes de que se le asigne la CPU

- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU durante 15 ms (tiempo de ráfaga).
- 3) En el instante 15 se vuelve a ejecutar el planificador a corto plazo, ¿quién hay en la cola de preparados? El proceso B y el C.
¿A cuál le asigno la CPU? Al B, porque llegó primero a la cola de preparados.
¿Hasta cuándo? Hasta que el proceso voluntariamente lo libere (*no apropiativo*).
- 4) En el instante 18 volvemos a activar el planificador a corto plazo, ¿quién hay en la cola? El proceso C. Le asigno la CPU durante 3 ms (tiempo de ráfaga)
- 5) Podemos decir que en relación al tiempo de sus ráfagas de CPU:
A es un proceso largo, y B y C son cortos.

Procesos	Tº llegada	Ráfaga	T (tº respuesta)	M (tº espera)
A	2	15	$21-2 = 19$	4
B	1	3	$6-1 = 5$	2
C	0	3	$3-0 = 3$	0

Diagrama de ocupación de la CPU



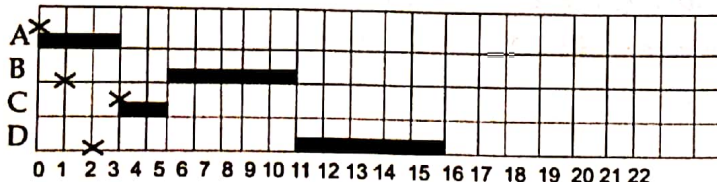
- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso C. El planificador a corto plazo le da la CPU durante 3 ms (tiempo de ráfaga).
- 3) En el instante 3 se vuelve a ejecutar el planificador a corto plazo, ¿quién hay en la cola de preparados? El proceso B y el A.
¿A cuál le asigno la CPU? Al B, porque llegó primero a la cola de preparados.
¿Hasta cuándo? Hasta que el proceso voluntariamente lo libere (*no apropiativo*).
- 4) En el instante 6 volvemos a activar el planificador a corto plazo, ¿quién hay en la cola? El proceso A. Le asigno la CPU durante 15 ms (tiempo de ráfaga)
- 5) Podemos ver cómo han mejorado ahora los tº de respuesta de los procesos cortos.
- 6) La penalización de los procesos va a depender no de sus ráfagas, sino de su posición en la cola de preparados.

El más corto primero (SJF) (Shortest Job First)

- Es no apropiativo.
- Cuando el procesador queda libre, selecciona el proceso que requiera un tiempo de servicio menor.
- Si existen dos o más procesos en igualdad de condiciones, se sigue FCFS.
- Necesita conocer explícitamente el tiempo estimado de ejecución (tº servicio) ¿Cómo?.
- Disminuye el tiempo de respuesta para los procesos cortos y discrimina a los largos. (Puesto que da más prioridad a los procesos con un menor tiempo de servicio)
- Tiempo medio de espera bajo.

Procesos	Tº llegada	Ráfaga real	Ráfaga estimada
A	0	3	3
B	1	5	5
C	3	2	2
D	2	5	5

Diagrama de ocupación de la CPU

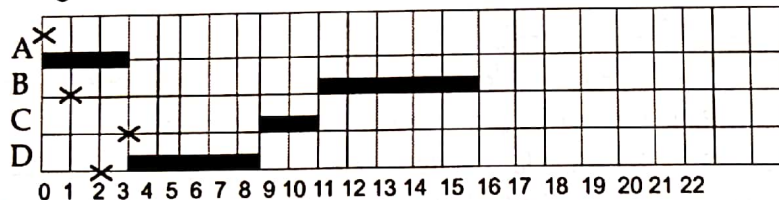


- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU durante 3 ms (tiempo de ráfaga).
- 3) En el instante 3 se vuelve a ejecutar el planificador a corto plazo, ¿quién hay en la cola de preparados? Los procesos B, C, D.
¿A cuál le asigno la CPU? Al C, porque es el que requiere menor tiempo de CPU.
- 4) En el instante 5 volvemos a activar el planificador a corto plazo, ¿quién hay en la cola? El proceso B y D. Tienen = tiempo de ráfaga, por lo que aplicamos FCFS. Le asigno la CPU al B.
- 5) Por último le asignamos la CPU a D.

- Ejemplo del más corto primero, suponiendo que se ha hecho una mala estimación.

Procesos	Tº llegada	Ráfaga real	Ráfaga estimada
A	0	3	3
B	1	5	6
C	3	2	5
D	2	5	5

Diagrama de ocupación de la CPU



Nos fijamos en la ráfaga estimada, en lugar de en la ráfaga real

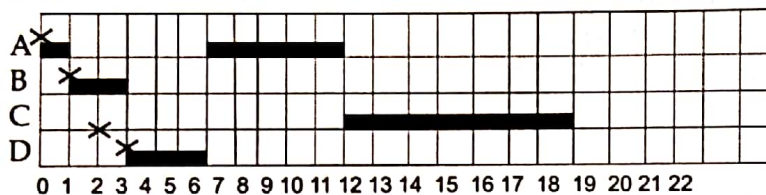
- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU durante 3 ms (tiempo de ráfaga).
- 3) En el instante 3 se vuelve a ejecutar el planificador a corto plazo, ¿quién hay en la cola de preparados? Los procesos B, C, D.
¿A cuál le asigno la CPU? Al D, porque C y D tienen = tiempo de ráfaga, por lo que aplicamos FCFS. Le asigno la CPU al D (que ha llegado antes).
- 4) Ahora se le da al C. Y lógicamente no se consumen 5, se consumen 2 porque cuando el proceso ejecuta su tiempo de ráfaga real finaliza o se bloquea, y entonces se activa el planificador a corto plazo.
- 5) En el instante 11 le asignamos la CPU a B, que también consume su tiempo de CPU real.

El más corto primero apropiativo (SRTF)

- Cada vez que entra un proceso a la cola de preparados se comprueba si su tiempo de servicio es menor que el tiempo de servicio que le queda al proceso que está ejecutándose. (Se activa el planificador a corto plazo cada vez). Casos:
 - **Sí es menor:** se realiza un cambio de contexto y el proceso con menor tiempo de servicio es el que se ejecuta.
 - **No es menor:** continúa el proceso que estaba ejecutándose.
- El tiempo de respuesta es menor excepto para procesos muy largos.
- Se obtiene la menor penalización en promedio (mantiene la cola de ejecutables con la mínima ocupación posible).

Procesos	Tº llegada	Ráfaga real	Ráfaga estimada
A	0	6	6
B	1	2	2
C	2	7	7
D	3	3	3

Diagrama de ocupación de la CPU



- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU hasta que entre un nuevo proceso.
- 3) En el instante 1 B ha llegado a la cola de preparados, ahora se comprueba ¿cuánto tiempo le queda a A? $6(\text{ráfaga real}) - 1(\text{que ha consumido}) = 5$. ¿Cuánto necesita B? 2(tiempo de ráfaga).
¿A cuál le asigno la CPU? Al B, porque es el que tiene menor tiempo de ráfaga.
- 4) En el instante 2 llega un nuevo proceso (C). Se activa el planificador a corto plazo, pero el proceso C requiere 7 y a B solo le queda 1 por consumir. Por lo tanto, no se produce cambio de contexto, no se le asigna la CPU.
- 5) Termina B. En el instante 3, ¿quién hay en la cola de preparados? A con un tiempo de CPU de 5, C con un tiempo de 7 y D con tiempo 3. Asignamos la CPU a D.
- 6) D consume el tiempo de CPU. En la cola tenemos a B y C, ¿a quién se le asigna? Al proceso A (le queda menor tiempo).
- 7) Termina A y finalmente se le asigna la CPU al proceso C.

¿Cómo podemos conocer la duración de la siguiente ráfaga de CPU?

Ejemplos

- Sólo podemos estimar su duración.
- Podemos estimar el valor de la siguiente ráfaga basándonos en las ráfagas previas de CPU y utilizando una media exponencial, por ejemplo, para la $n+1$ ráfaga:
 T_n = duración actual de la n -ésima ráfaga
 Y_n = valor estimado de la n -ésima ráfaga
 $0 \leq W \leq 1$
 Definimos: $Y_{n+1} = W * T_n + (1-W) Y_n$
 Y_0 = valor inicial (constante o promedio global del sistema)

- $W = 0$
 $Y_{n+1} = Y_n \rightarrow$ La historia reciente no influye.
- $W = 1$
 $Y_{n+1} = T_n \rightarrow$ Sólo cuenta la ráfaga actual.
- Si desarrollamos la formula:
 $Y_{n+1} = W * T_n + (1-W) * W * T_{n-1} + (1-W)^2 * W * T_{n-2} + \dots + (1-W)^q * W * T_{n-q}$
 Si $W = \frac{1}{2} \rightarrow$ cada término sucesivo tiene menos efecto.

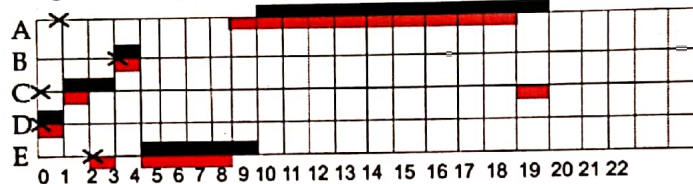
Planificación por prioridades

- Asociamos a cada proceso un número de prioridad (un núm. entero).
Valor de prioridad bajo → significa mayor prioridad.
- Se asigna la CPU al proceso con mayor prioridad (enteros menores = mayor prioridad)
 - Apropiativa
 - No apropiativa
- **Problema:** Inanición -- los procesos de baja prioridad pueden no ejecutarse nunca.
(por lo tanto tiempos de respuesta o la penalización muy malos)
- **Solución:** Envejecimiento -- con el paso del tiempo se incrementa la prioridad de los procesos.
(Sol: aumentar el nivel de prioridad, se irá bajando ese número)
Lo que significa que las prioridades son dinámicas durante la vida de un proceso.

Procesos	Tº llegada	Ráfaga	Prioridad
A	1	10	3
B	3	1	1
C	0	2	3
D	0	1	2
E	2	5	2

■ No apropiativo
■ Apropiativo

Diagrama de ocupación de la CPU



En el caso de que tengamos 2 procesos en la misma cola con la misma prioridad y sea la más alta, se utilizaría FCFS (por orden de llegada a la cola de preparados).

No apropiativo

- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? Los procesos C y D. El planificador a corto plazo le da la CPU al D porque su valor de prioridad es menor. Hasta que termine su ráfaga completa de CPU.
- 3) En el instante 1 D ha consumido su tiempo, ¿quién hay en la cola? A y C. Como tienen = prioridad le asignamos el procesador a C (por el tº de llegada).
- 4) C consume tº de CPU. En el instante 3, cola: A, B, E. Se le asigna a B (+ prioritario).
- 5) Termina B. En el instante 4, ¿quién hay? A y E. Asignamos la CPU a E (+ prioritario).
- 6) Se activa de nuevo planificador se le asigna CPU a A, hasta que termine.

Apropiativo

- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? Los procesos C y D. El planificador a corto plazo le da la CPU al D porque su valor de prioridad es menor.
- 3) En el instante 1 A llega, y D ha terminado, ¿quién hay en la cola? A y C. Como tienen = prioridad le asignamos el procesador a C (por el tº de llegada).
- 4) En el instante 2 llega E a la cola. Comprobamos según el algoritmo cuál tiene preferencia. Será el + prioritario. E + prioritario que C, se le asigna la CPU.
- 5) En el instante 3 llega B a la cola. B + prioritario que E, se llamaría al despachador que haría el cambio de contexto y se le asigna la CPU.
- 6) B termina, cola: A, C, E. Se le asigna a E hasta que consume su CPU. Después A y C. A primero porq = prioridad, pero a C se le retiró la CPU, y vuelve al final de la cola.

Por turnos (Round-Robin)

- La CPU se asigna a los procesos en intervalos de tiempo (quantum)
- Procedimiento:
 - » Si el proceso finaliza o se bloquea antes de agotar el quantum, libera la CPU. Se toma el siguiente proceso de la cola de ejecutables (la cola es FIFO) y se le asigna un quantum completo.
 - FIFO/FCFS → por orden de llegada
 - » Si el proceso no termina durante ese quantum, se interrumpe y se coloca al final de la cola de ejecutables.
- Es apropiativo.

Nota: En los ejemplos supondremos que si un proceso A llega a la cola de ejecutables al mismo tiempo que otro B agota su quantum, la llegada de A a la cola de ejecutables ocurre antes de que B se incorpore a ella.

Procesos	Tº llegada	Ráfaga
A	0	3
B	1	9
C	3	2
D	9	5

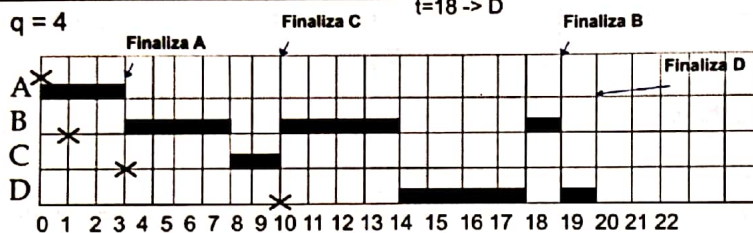
Estado de la cola de ejecutables

t=0 → A; t=3 → BC; t=7 → CB

t=9 → BD; t=13 → DB; t=17 → BD

t=18 → D

***Estado de la cola de ejecutables:**
tenemos puestos en cada instante quién está en las colas (por orden).



- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU. ¿Durante cuánto tiempo? Un máximo de 4 ms (quantum)
- 3) En el instante 3 A ha consumido su tiempo, ¿quién hay en la cola? B y C. Le asignamos el procesador a B (por el tº de llegada). Durante máx. 4 ms.
- 4) B consume 4 y vuelve al final de la cola. En el instante 7 el primer proceso de la cola es C.
- 5) C solo requiere 2ms, en el instante 9 finaliza o se bloquea. Cola: B y D, se le asigna a B porque llegó antes a la cola.
- 6) B consume 4 más y vuelve al final de la cola. Otro cambio de contexto, se le asigna el procesador a D durante máx. 4 ms.
- 7) D consume 4, y se le asigna de nuevo a B, que acaba. Después acaba D.

Por turnos (y III)

- Los valores típicos del quantum están entre 1/60sg y 1sg.
- Penaliza a todos los procesos en la misma cantidad, sin importar si son cortos o largos.
- Las ráfagas muy cortas están más penalizadas de lo deseable.
(Porque si el tº de servicio que requiere un proceso es muy pequeño, el cociente será alto).
(Recordar: penalización = tº de respuesta / tº de ráfaga (tº servicio de CPU que requiere) .)
- ¿Valor del quantum?
 - * muy grande (excede del tº de servicio de todos los procesos) → se convierte en FCFS
 - * muy pequeño → el sistema monopoliza la CPU haciendo cambios de contexto (tº del núcleo muy alto)

Colas múltiples

- La cola de preparados se divide en varias colas y cada proceso es asignado permanentemente a una cola concreta P.ej. interactivos y batch.
- Cada cola puede tener su propio algoritmo de planificación P. ej. interactivos con RR(Robin) y batch con FCFS.
- Requiere además una planificación entre colas
 - Planificación con prioridades fijas. P. ej. primero servimos a los interactivos luego (cuando ya no haya procesos en esa cola donde estaban los interactivos) a los batch.
 - Tiempo compartido -- cada cola obtiene cierto tiempo de CPU que debe repartir entre sus procesos. P. ej. 80% interactivos en RR y 20% a los batch con FCFS en FCFS

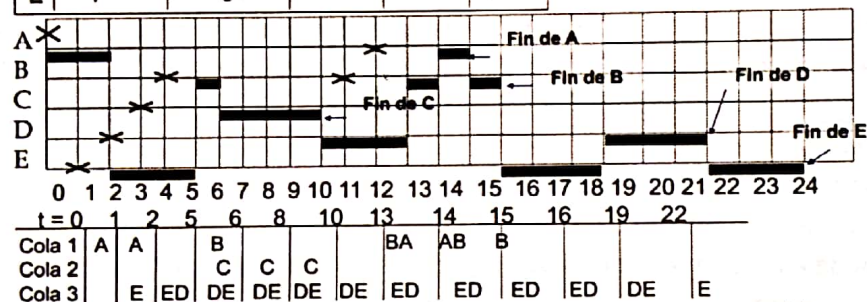
P	T ^o Llegada	Ráfaga	Bloqueo	Ráfaga	Cola
A	0	2	10	1	1
B	4	1	5	2	1
C	3	4	-	-	2
D	2	6	-	-	3
E	1	9	-	-	3

El algoritmo de cada cola es RR con $q=1, 2$ y 3 mlsq. respectivamente.

El algoritmo entre colas es prioridades no apropiativo

***Cola:** cola a la que pertenecen según la prioridad que tienen.

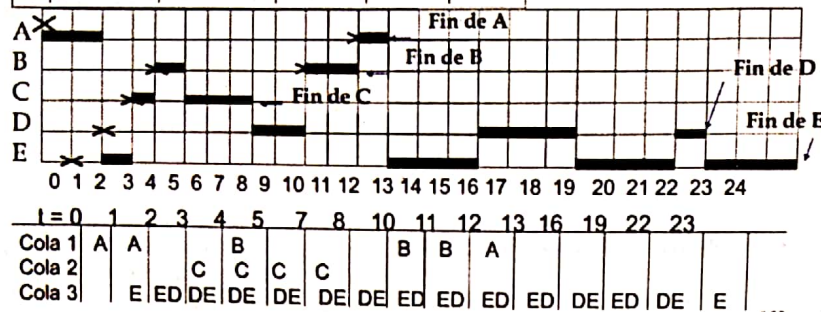
Para cada cola tenemos un algoritmo (en este caso Robin para todas, y solo cambian los quantum de tiempo)



*Ponemos estados de las colas en los instantes de tiempo en los que se activa el planificador a corto plazo.

- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU. ¿Durante cuánto tiempo? Un máximo de 1 ms(quantum)
- 3) A consume 1 y vuelve al final de la cola 1. En este instante procesos A en cola 1, y E en la cola 3, se aplica el algoritmo de prioridades entre colas, si hay procesos en la cola + prioritaria (cola 1), entonces se le asigna el proceso de la cola 1 (A).
- 4) Se aplica RR para cola 1 con quantum de 1, terminando esa ráfaga. Ahora se bloquea durante 10 ms, y se va a desbloquear en el instante 12 (marcamos con x). (A deja de estar en la cola de preparados, pasa a la cola de bloqueados)
- 5) En el instante 2 tenemos en cola procesos E y D (cola 3). Aplicamos RR, se le asigna CPU a E, que es el primero de la cola, durante un máx. de 3 mlsq.
- 6) En el instante 5, tenemos procesos en + de 1 cola, por lo que aplicamos algoritmo entre colas. Tenemos procesos en la cola 1 que es la + prioritaria, en la cola 1 aplicamos RR, se le asigna procesador al proceso B. B termina esa ráfaga, y se bloquea durante 5 (marcamos con x).
- 7) ...

- | P | TºLlegada | Ráfaga | Bloqueo | Ráfaga | Cola |
|---|-----------|--------|---------|--------|------|
| A | 0 | 2 | 10 | 1 | 1 |
| B | 4 | 1 | 5 | 2 | 1 |
| C | 3 | 4 | - | - | 2 |
| D | 2 | 6 | - | - | 3 |
| E | 1 | 9 | - | - | 3 |
- El algoritmo de cada cola es RR con $q=1, 2$ y 3 mlsq. respectivamente.
- El algoritmo entre colas es prioridades *apropiativo*.



- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU. ¿Durante cuánto tiempo? Un máximo de 1 ms (quantum)
- 3) A consume 1 y vuelve al final de la cola 1. En este instante (1) procesos A en cola 1, y E en la cola 3, se aplica el algoritmo de prioridades entre colas, si hay procesos en la cola + prioritaria (cola 1), entonces se le asigna el proceso de la cola 1 (A).
- 4) Se aplica RR para cola 1 con quantum de 1, terminando esa ráfaga. Ahora se bloquea durante 10 ms, y se va a desbloquear en el instante 12 (marcamos con x). (A deja de estar en la cola de preparados, pasa a la cola de bloqueados)
- 5) En el instante 2 tenemos en cola procesos E y D (cola 3). Aplicamos RR, se le asigna CPU a E, que es el primero de la cola, durante un máx. de 3 mlsq.
- 6) En el instante 3, entra un proceso (C) a una cola + prioritaria, se activa el planificador (por algoritmo apropiativo) y se le asigna a C, máx. 2 mlsq.
- 7) C consume 1, y llega un nuevo proceso a una cola + prioritaria (B). Se le asigna CPU al proceso B.
- 8) B consume quantum y ráfaga, se bloquea durante 5 seg.
- 9) ...

- Un proceso se puede mover entre varias colas durante su vida.
- Cada cola puede tener su propio algoritmo de planificación P. ej. interactivos con RR y batch con FCFS.
- Requiere definir los siguientes parámetros:
 - » Número de colas
 - » Algoritmo de planificación para cada cola
 - » Método utilizado para determinar cuando trasladar a un proceso a otra cola
 - » Método utilizado para determinar en qué cola se introducirá un proceso cuando necesite un servicio (es decir, cuando llegue a la cola de preparados)
 - » Algoritmo de planificación entre colas
- Mide en tiempo de ejecución el comportamiento real de los procesos.
- Disciplina de planificación más general (Unix, Linux Windows NT).

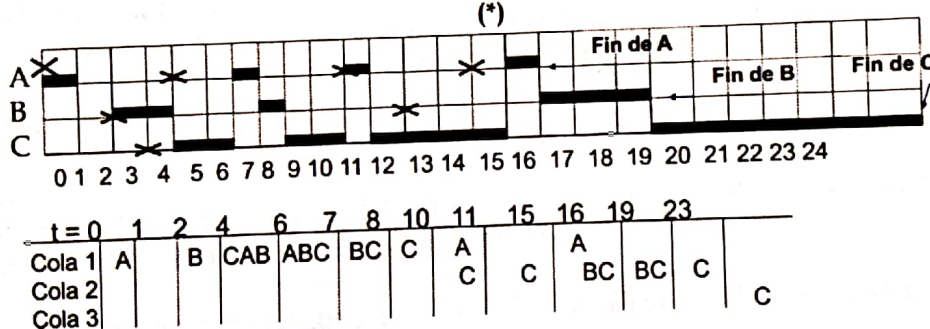
• Ejemplo colas múltiples con realimentación

- Tres colas gestionadas mediante Round Robin:
 - » Cola 1 con quantum = 2 milisegundos
 - » Cola 2 con quantum = 4 milisegundos
 - » Cola 3 con quantum = 8 milisegundos
- Algoritmo entre colas: prioridades no apropiativo, cola 1 mayor prioridad, cola 3 menor prioridad.
- Los procesos entran inicialmente en la cola 1.
- Cuando un proceso se bloquea, al regresar a la cola de preparados sigue en la misma cola si no hay traspaso.
- Un proceso se traspasa a una cola de menor prioridad cuando agota dos quantum de tiempo seguidos (2 quantum de su cola), o bien, agota uno y se bloquea en el siguiente.

P	Tº Llegada	Tº Servicio total	Ráfaga	Bloqueo
A	0	4	1	3
B	2	6	3	4
C	3	23	-	-

(*) Los procesos tienen un comportamiento cíclico

**Tº servicio total: ej. el proceso A consume ráfaga de 1, se bloquea durante 3, luego otro, se bloquea otros 3...*



Hasta completar los 4 de servicio total

- 1) Marcamos con la x los instantes de llegada.
- 2) En el instante 0, ¿quién hay en la cola de preparados? El proceso A. El planificador a corto plazo le da la CPU. ¿Durante cuánto tiempo? Un máximo de 2 mls. (q)
- 3) A consume 1 (ráfaga) y se bloquea durante 3.
- 4) El planificador no se activa hasta que B llega (también a la cola de más prioridad). Se le asigna CPU durante un máx. de 2. Y los consume. Vuelve al final de la cola.
- 5) En instante 4 ya ha llegado C, A desbloqueado, y B el último de la cola. (Notamos que C (que llega en a mismo tiempo que A se desbloquea) va primero.)
- 6) Se le asigna CPU al C, que consume q completo y vuelve a la cola de preparados.
- 7) A pasa al primero en la cola, y se le asigna CPU durante ráfaga de 1 y se bloquea 3.
- 8) En inst. 8 tenemos B y C se le asigna a B que consume 1 (ráfaga completa), y se bloquea 4.
- 9) En inst. 10 tenemos solo C y consumimos quantum. C ha cumplido uno de los requisitos para traspaso de cola (ha agotado 2 quantum de tiempo seguidos). Pasa a la cola 2.
- 10) ...