

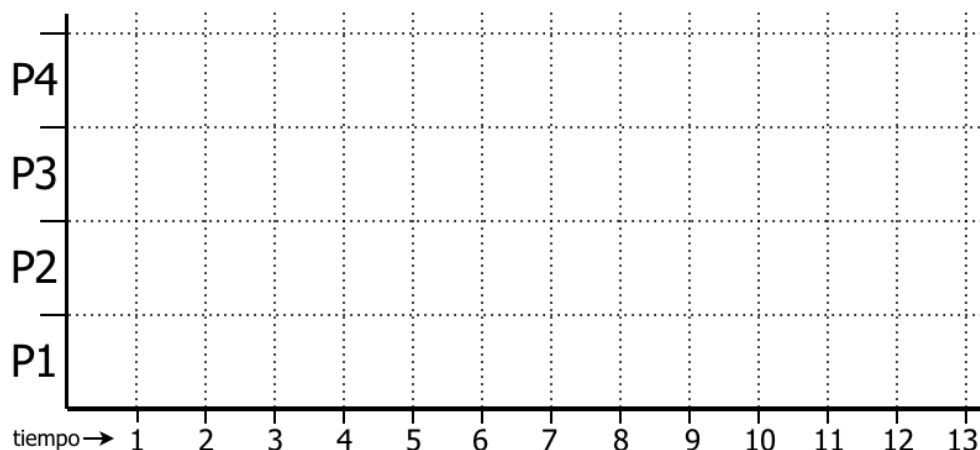


1. Sea un sistema de planificación de procesos caracterizado por la siguiente tabla en la que representamos el tiempo de llegada, la duración de las ráfagas de CPU y el período de entrada/salida para cada uno de los procesos, todo ello expresado en milisegundos.

Proceso	Prioridad	Llegada al sistema	Tiempo de CPU	Tiempo de E/S	Tiempo de CPU
P0	4	0	1	1.5	4
P1	3	1	5		
P2	1	2	2		
P3	2	3	1		

Se pide simular la planificación conjunta de estos 4 procesos mediante el diagrama de Gantt adjunto y calcular los tiempos medios de compleción (T_r) y espera (T_e) para ese conjunto de 4 procesos. El ejercicio debe completarse de forma individual para cada uno de los siguientes planificadores:

- First Come First Served (FCFS).
- Shortest Job First (SJF).
- Asignación por prioridad no expropiativa.
- Tiempo compartido (*Round Robin*) con un *quantum* de 1 milisegundo.
- Shortest Remaining Time First (SRTF).
- Prioridad expropiativa.



2. Sea un sistema operativo con planificación de colas de realimentación multinivel en el que se ejecutan tres procesos que llegan al mismo tiempo, estableciéndose el orden de comienzo P1, P2 y P3. P1 requiere 8 ciclos de CPU, P2 10 ciclos y P3 7 ciclos. Tan sólo P2 realiza una operación de entrada/salida, que se completa muy brevemente, gastando un ciclo adicional justo a la conclusión de su tercer ciclo de CPU. La siguiente tabla resume todos estos datos.

	CPU	E/S	CPU
P1	8		
P2	3	1	7
P3	7		

El planificador es expropiativo por prioridad, existiendo tres colas C1, C2 y C3 de mayor a menor prioridad. Los procesos comienzan su ejecución en C1, donde se establece una estrategia *round-robin* (tiempo compartido) con un *quantum* de 2 ciclos. Para C2 se establece *round-robin* con un *quantum* de 3, y para C3 un algoritmo *SRTF* (Shortest Remaining Time First).

Considera las siguientes reglas para el encolado de procesos:

- Si un proceso agota su *quantum*, baja de nivel, y si no lo hace, asciende de nivel.
- Un proceso desalojado por prioridad mantiene ésta, y pasa al final de la cola en la que partió, donde recibirá un *quantum* completo.
- Un proceso que reanuda su ejecución tras un bloqueo también recibe un *quantum* completo.

Se pide lo siguiente:

- a) Simular la planificación de los procesos mediante un diagrama temporal de barras horizontales (Gantt) donde se refleje qué proceso ocupa la CPU en cada momento.
 - b) Si la cola C3 cambiara el algoritmo SRTF por SJF, ¿en qué orden finalizarían los procesos ahora? ¿y si el algoritmo de C3 fuera FCFS (cola FIFO)?
3. Sea un sistema operativo con planificación de colas de realimentación multinivel en el que se ejecutan tres procesos que llegan al mismo tiempo, estableciéndose el orden de comienzo P1, P2 y P3. P1 requiere 8 ciclos de CPU, P2 7 ciclos y P3 10 ciclos. Tan sólo P1 realiza una operación de entrada/salida, que se completa en 4 ciclos justo a la conclusión de su cuarto ciclo de CPU. La siguiente tabla resume la situación.

	CPU	E/S	CPU
P1	4	4	4
P2	7		
P3	10		

Utilizamos un planificador expropiativo con tres colas C1, C2 y C3 de mayor a menor prioridad. Los procesos comienzan su ejecución en C1, donde se establece una estrategia *round-robin* (tiempo compartido) con un *quantum* de 2 ciclos. Para C2 usamos *round-robin* con un *quantum* de 3 ciclos, y para C3 un algoritmo *FCFS* (First Come First Served).

El encolado de procesos se rige por las siguientes normas:

- Si un proceso agota su *quantum* baja de nivel, pero si no lo hace mantiene su nivel.
- Un proceso desalojado por prioridad también mantiene ésta, y pasa al final de la cola en la que partió, donde recibirá un *quantum* completo cuando vuelva a ocupar la CPU.
- Un proceso que reanuda su ejecución tras un bloqueo también recibe un *quantum* completo.

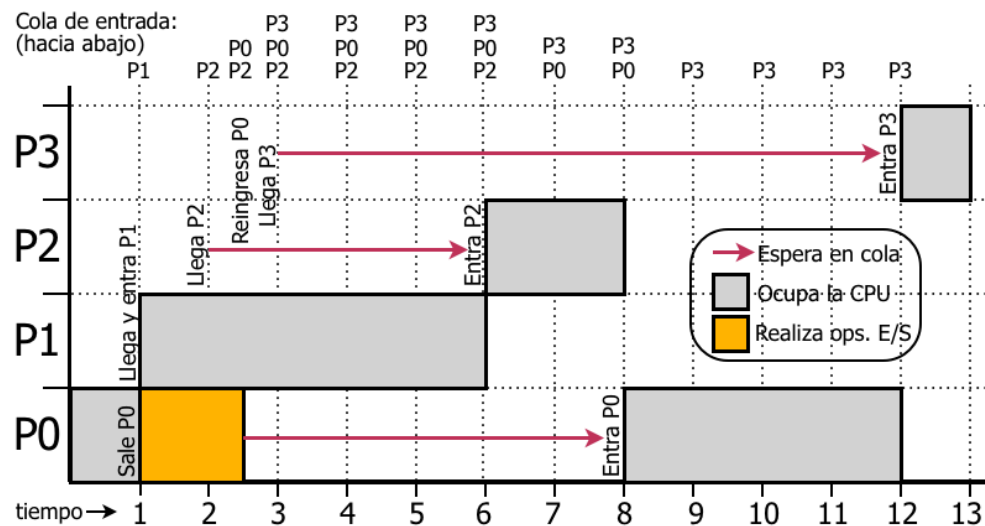
Se pide lo siguiente:

- a) Simular la planificación de los procesos mediante un diagrama temporal de barras horizontales (Gantt) donde se refleje qué proceso ocupa la CPU en cada momento.
- b) Si la cola C3 cambiara el algoritmo FCFS por SJF, ¿en qué orden finalizarían los procesos ahora?

SOLUCIONES A LOS EJERCICIOS:

1. Resolvemos el análisis de cada estrategia de planificación por separado.

- a) **Planificación FCFS** (*First Come First Served* - el primero que llega es el primero que se atiende): Según llega cada proceso (o termina una operación de entrada/salida y recupera su estado *Ready*), ingresa en la cola que marca el orden de uso de la CPU. Al ser un algoritmo no expropiativo, el proceso que ocupa la CPU permanece allí hasta que complete su ráfaga de CPU, esto es, hasta que finalice su ejecución o realice una operación de entrada/salida, momento en que es reemplazado por el primero de la cola. El siguiente diagrama ilustra cómo se comparte la CPU siguiendo este algoritmo de planificación, donde la parte superior muestra el estado de la cola que marca el siguiente proceso en ocupar la CPU en cada momento. Lo más significativo que acontece es que cuando P0 finaliza sus 15 milisegundos de entrada/salida ya han llegado P1 y P2, por lo que se reubica detrás de ellos, y poco después llega P3, que ya será el último proceso en ejecutarse.

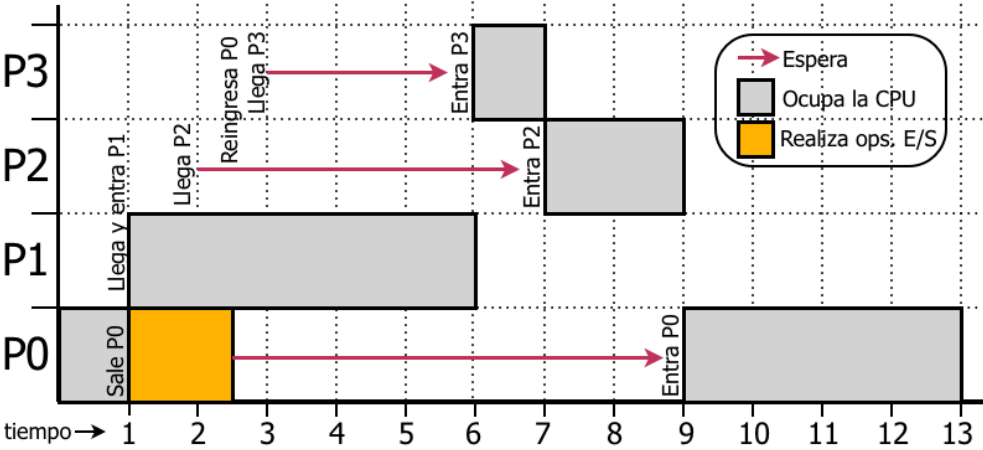


Para calcular los tiempos de compleción (T_r), nos fijamos en la llegada y finalización de cada proceso, restando el primero al segundo. Por otro lado, el tiempo de espera (T_e) se ha marcado para cada proceso con una flecha roja en el diagrama anterior. Todo esto nos lleva a la siguiente tabla, que nos ayuda en el cálculo de los valores medios, $T_r = 82,5$ y $T_e = 46,25$:

Proceso	Llega	Acaba	T_r	T_e
P0	0	12	12	5,5
P1	1	6	5	0
P2	2	8	6	4
P3	3	13	10	9
Valor promedio			8,25	4,62

- b) **Planificación SJF** (*Shortest Job First*, primero el trabajo más corto): En este caso, la planificación de los dos primeros procesos no cambia respecto al caso anterior, puesto que son la única elección del planificador en los instantes 0 y 1. A partir del tercer proceso que llega, momento en el que ya sí debe elegirse, ocupará la CPU aquel proceso que tiene pendiente menos tiempo de CPU. Por lo tanto, el proceso P0, que es el que más tiempo necesita el recurso es relegado al último lugar.

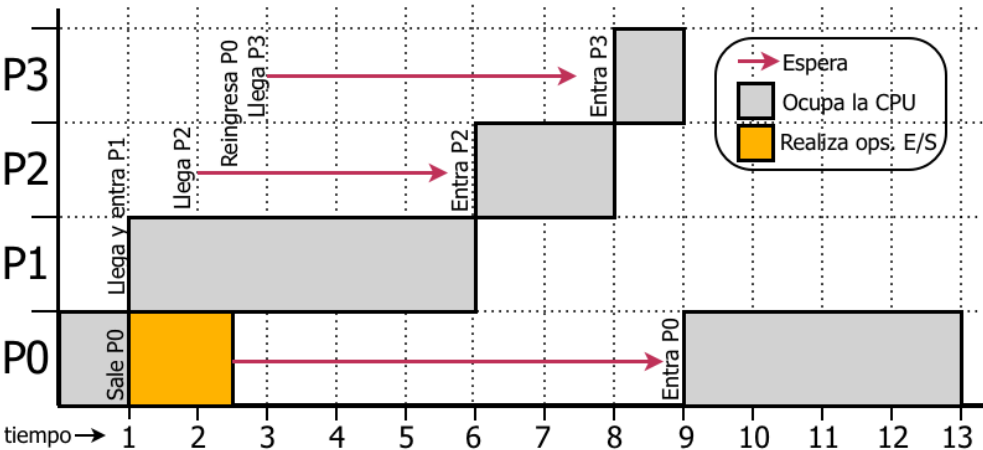
En este caso, no necesitamos mantener la cola de entrada en la parte superior del diagrama de Gantt, pues basta con otorgar la CPU al proceso más corto. La tabla de tiempos se elabora de forma similar al caso anterior con las nuevas cotas temporales de finalización para cada proceso. El resultado es el siguiente:



Proceso	Llega	Acaba	T_r	T_e
P0	0	13	13	6,5
P1	1	6	5	0
P2	2	9	7	5
P3	3	7	4	3
Valor promedio			7,25	3,625

Se aprecia una reducción en el tiempo medio de compleción, T_r , siendo éste el principal objetivo del algoritmo SJF. La espera media, T_e , también se ha reducido, y todo a costa de postergar al proceso P0 (el más extenso), en beneficio de los procesos más ágiles.

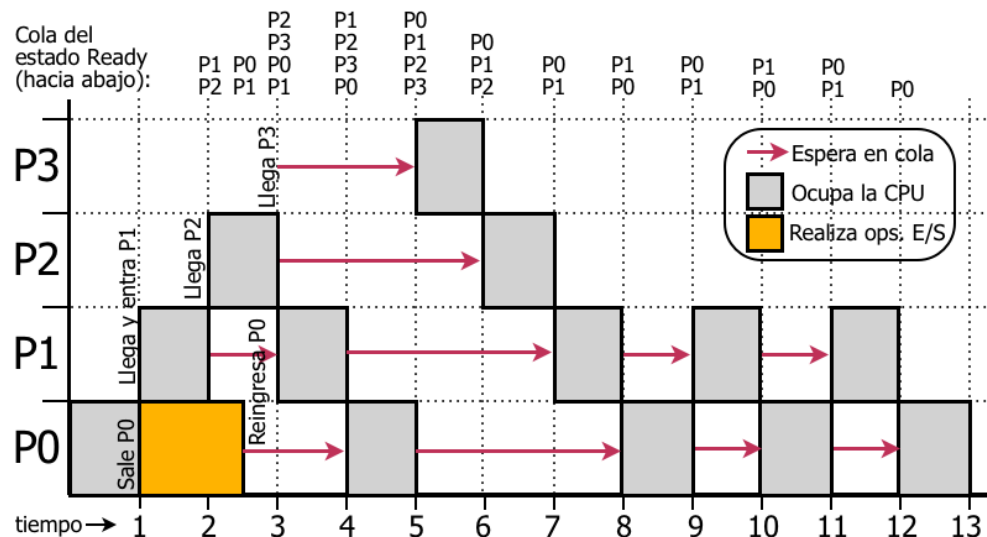
c) **Planificación por prioridad no expropiativa.** Es quizá el ejercicio más sencillo, puesto que hay que ordenar la ejecución de mayor a menor prioridad, esto es, primero los procesos con números más bajos en la columna de la tabla que nos informa acerca de su prioridad. Eso sí, P0 es de nuevo el primero en tomar la CPU porque, aunque tiene la prioridad más baja, es el único que llega al sistema en el instante inicial. Esa baja prioridad le penalizará cuando finalice sus operaciones de entrada/salida, puesto que para entonces los otros tres procesos ya han entrado al sistema y su ejecución restante queda relegada para la última.



Proceso	Llega	Acaba	T_r	T_e
P0	0	13	13	6,5
P1	1	6	5	0
P2	2	8	6	4
P3	3	9	6	5
Valor promedio			7,5	3,875

- d) **Planificación por tiempo compartido con $quantum\ q = 1$.** Este método prioriza la justicia en el reparto de recursos frente a la eficiencia, lo que explica que los tiempos promedio sean superiores a los de casos anteriores. Lo más importante para resolver correctamente la asignación de la CPU es que para establecer el turno rotatorio de *quantums* de tiempo entre los procesos se lleva una cola ligada a los procesos que comparten el estado *Ready*, y cuando un proceso acaba su *quantum* se sitúa al final de dicha cola para dar antes oportunidad a todos los demás.

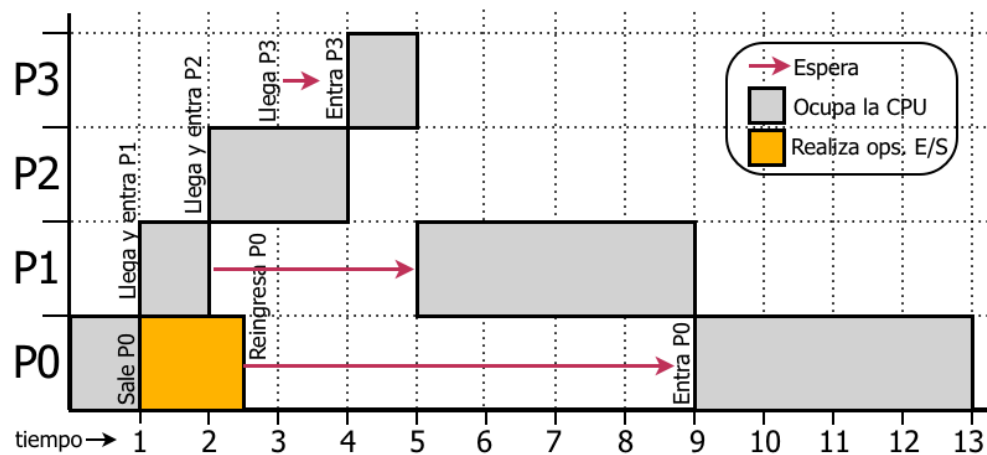
Teniendo todo esto en cuenta, el diagrama de Gantt y la tabla de tiempos son los siguientes:



Proceso	Llega	Acaba	T_r	T_e
P0	0	13	13	6,5
P1	1	12	11	6
P2	2	7	5	3
P3	3	6	3	2
Valor promedio			8	4,375

- e) **Planificación SRTF (Shortest Remaining Time First, primero el trabajo al que le quede menos ráfaga de CPU).** Es la versión expropiativa del método SJF (*Shortest Job First*), por lo que en cuanto entra un proceso al que le queda menos tiempo de CPU que el que la tiene, le quitará la CPU para quedársela. En nuestra secuencia, esto ocurre cuando llega P2, sacando a P1. P3 es el proceso más corto, pero llega justo cuando a P2 le queda ese mismo tiempo para terminar, por lo que el Sistema Operativo, a igualdad de condiciones, se queda con P2 para no tener que cambiar de contexto. Los grandes perjudicados aquí son los procesos más largos, P0 y P1, que sólo pueden acabar una vez lo han hecho los dos más cortos, P2 y P3.

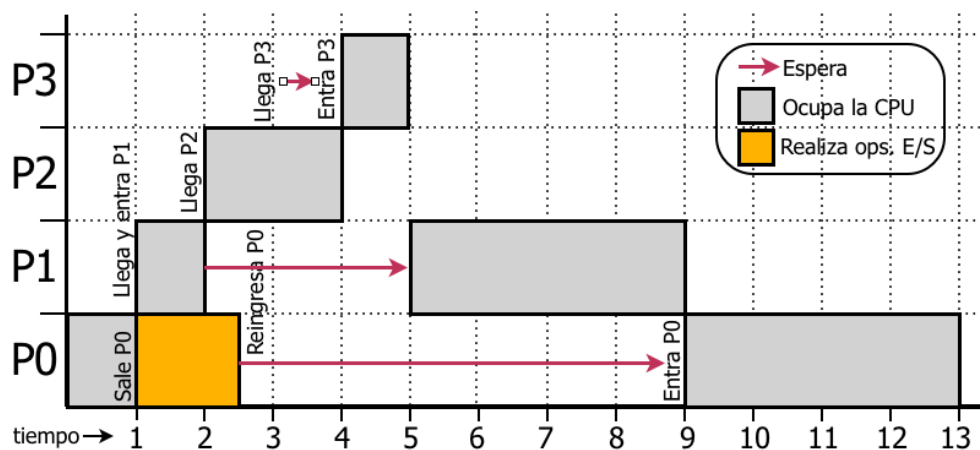
Con todo esto, el diagrama de Gantt y la tabla de tiempos quedan como sigue:



Proceso	Llega	Acaba	T_r	T_e
P0	0	13	13	6,5
P1	1	9	8	3
P2	2	4	2	0
P3	3	5	2	1
Valor promedio			6,25	2,625

f) **Planificación por prioridad expropiativa.** Al igual que la prioridad no expropiativa, la versión expropiativa resulta fácil de simular, ya que basta con fijarse en la columna de la tabla que indica la prioridad. La diferencia entre ambas es que en la primera tomamos la decisión cada vez que acaba el proceso en curso, mientras que aquí lo haremos en cuanto llega un nuevo proceso: Si éste tiene prioridad mayor que el que tiene la CPU (esto es, un número más pequeño), provocará el pertinente relevo en la CPU.

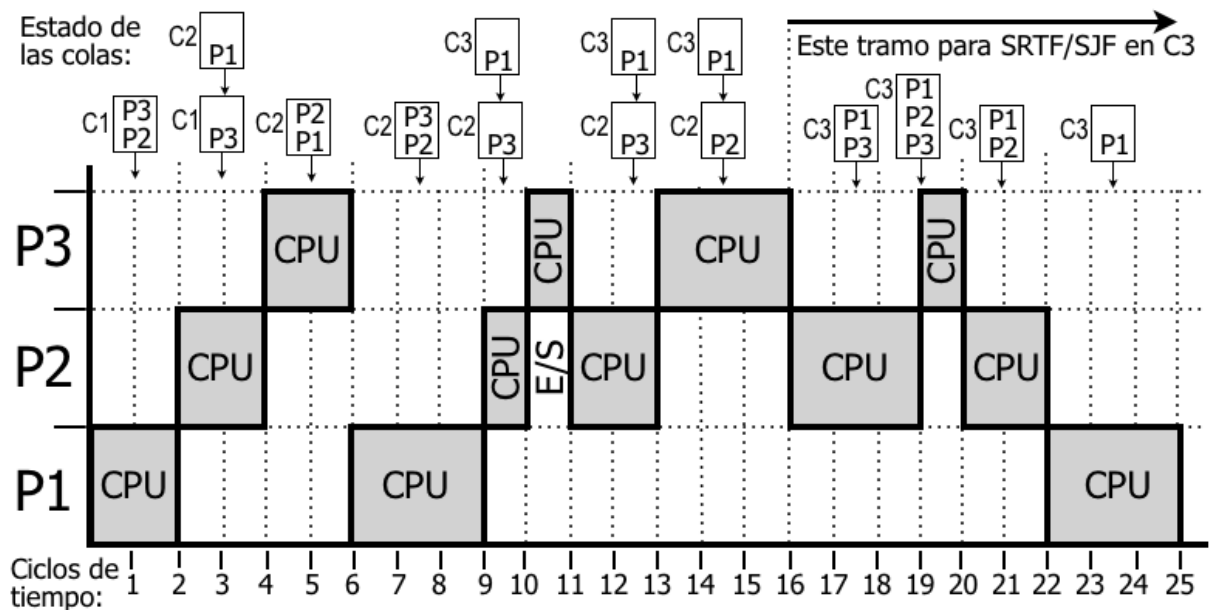
En el caso particular de este ejercicio se da la curiosidad de que las prioridades más altas (números más bajos) se han asignado a los procesos más cortos (P2 y P3), por lo que la asignación de la CPU coincide con el método STRF que acabamos de ver.



Proceso	Llega	Acaba	T_r	T_e
P0	0	13	13	6,5
P1	1	9	8	3
P2	2	4	2	0
P3	3	5	2	1
Valor promedio			6,25	2,625

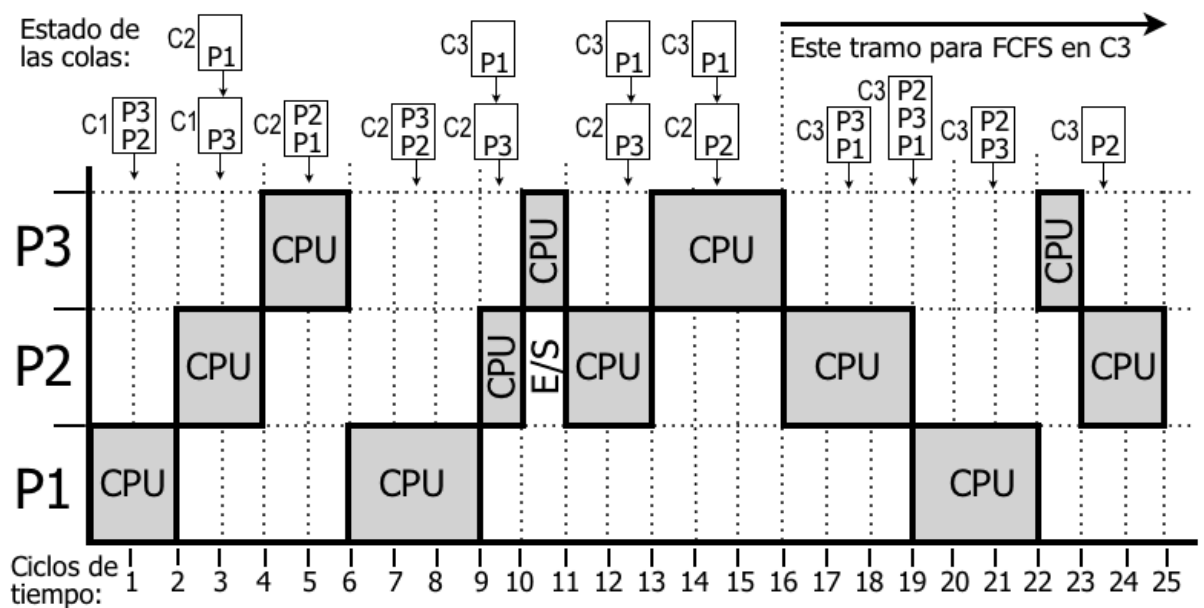
2. En un sistema de planificación de tres colas multinivel, primero se atienden los procesos de la cola 1, y sólo cuando ésta queda vacía comienzan a atenderse los procesos de la cola 2. De forma similar, cuando se agota la cola 2 comienzan a tomarse procesos de la cola 3. De esta manera, podemos considerar que las 3 colas van encadenadas: C1 seguida de C2, y finalmente C3. Así vamos a representarlas en la parte superior del diagrama de Gantt, donde anotaremos todo su movimiento para guiarnos a obtener el siguiente proceso que ocupa la CPU en cada cambio de contexto.

- a) En base a estas premisas y a las reglas que indica el enunciado del ejercicio, la planificación de procesos queda como muestra su diagrama de Gantt a continuación:



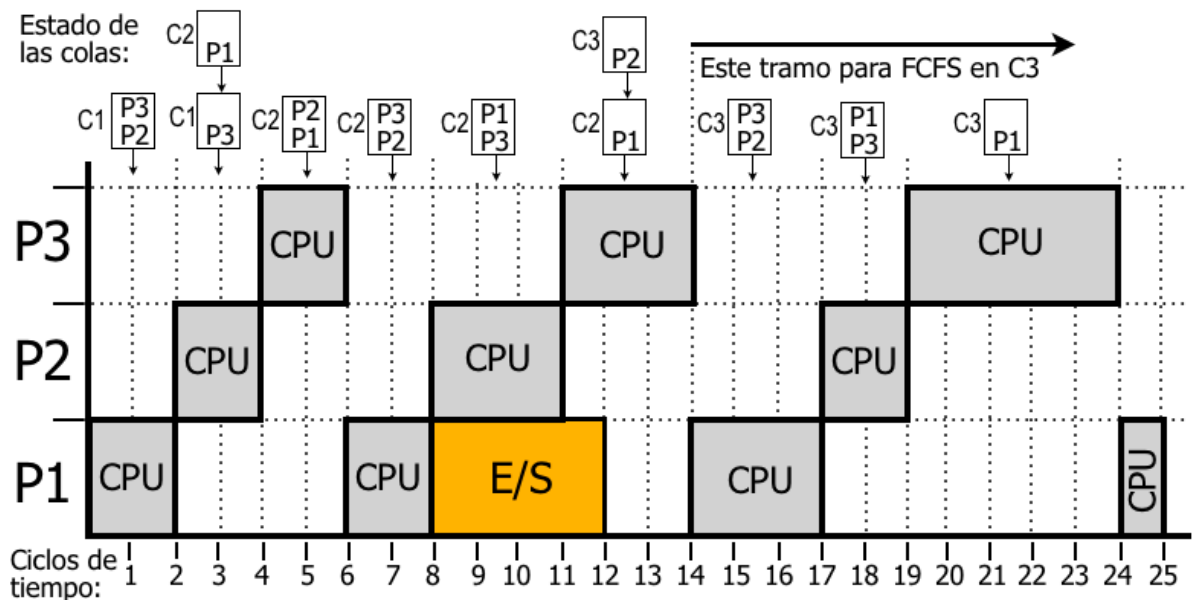
Notar la incorporación del movimiento de los procesos por las 3 colas multinivel en la parte superior. Cada proceso entra a cada cola por arriba y sale por abajo, evolucionando dentro de ella en secuencia vertical.

- b) Respecto a la segunda cuestión que nos plantea cómo finalizan los procesos si la cola C3 cambiara el algoritmo SRTF por SJF o por FCFS, hay que fijarse a partir del ciclo 16 en que el proceso P3 va a parar a la cola C3 donde ya está P1 esperando. Usando SRTF o SJF, P3 se coloca delante de P1, ya que P3 es el proceso al que le quedan menos ciclos de CPU para terminar. En cambio, si usamos FCFS, P3 se colocaría tras P1, puesto que se respeta el orden de llegada (cada nueva entrada se coloca siempre al final). Tomando todas estas consideraciones, la planificación de procesos tomando FCFS en C3 sólo queda alterada a partir de ese ciclo 16, en que se modifica como sigue:



3. Aquí procederemos de forma similar al ejercicio anterior, puesto que las reglas que establecen el movimiento por las colas multinivel son similares. La única diferencia es que en este caso, cuando un proceso no agota su *quantum* de tiempo y es desalojado, se mantiene en la misma cola que estaba en lugar de ascender al siguiente nivel más prioritario.

a) El diagrama de Gantt que ilustra la planificación de tareas se muestra a continuación:



En la parte superior del diagrama se detalla todo el movimiento que se produce en las colas que determinan el orden de entrada al sistema.

- b) Respecto a la segunda cuestión que nos plantea cómo finalizan los procesos si la cola C3 cambiara el algoritmo FCFS por SJF, hay que fijarse a partir del ciclo 14, instante en el que empiezan a acumularse procesos en dicha cola. Si la cola se gestiona con FCFS, el orden de finalización sería el orden de entrada en la cola, que ha sido P2, P3 y P1. Pero con SJF se

elige primero el proceso al que queda menos tiempo de CPU por ejecutar, y por lo tanto el orden de finalización sería P1, P2 y P3. El siguiente diagrama muestra los cambios del planificador si se implementara SJF en la cola C3, en el que se aprecian cambios a partir del ciclo 17.

