

## **EJEMPLOS DE EJERCICIOS RESUELTOS SOBRE LA GESTIÓN DE PROCESOS POR PARTE DEL S.O.**

1. Primero veremos la gestión de procesos **No apropiativos**. (No hay procesos que se apropien de tiempos de CPU de otros procesos -un proceso se apodera de la CPU hasta que termina de ejecutarse):

- **FCFS (First Come First Served).**

Es el famoso **FIFO (Primero en llegar primero en salir)**; es decir, los procesos usan la CPU en el orden que llegaron. Es un algoritmo **no apropiativo**, es decir, a medida que llegan los procesos van usando la CPU y cuando hacen uso de ésta, no la abandonan hasta que se bloqueen por E/S o finalicen.

Supongamos que llegan al sistema 4 procesos en el orden que se presenta a continuación y con la longitud que se indica.

Proceso	Duración	Tiempo de llegada
P3	3	0
P4	4	2
P1	5	5
P2	2	7

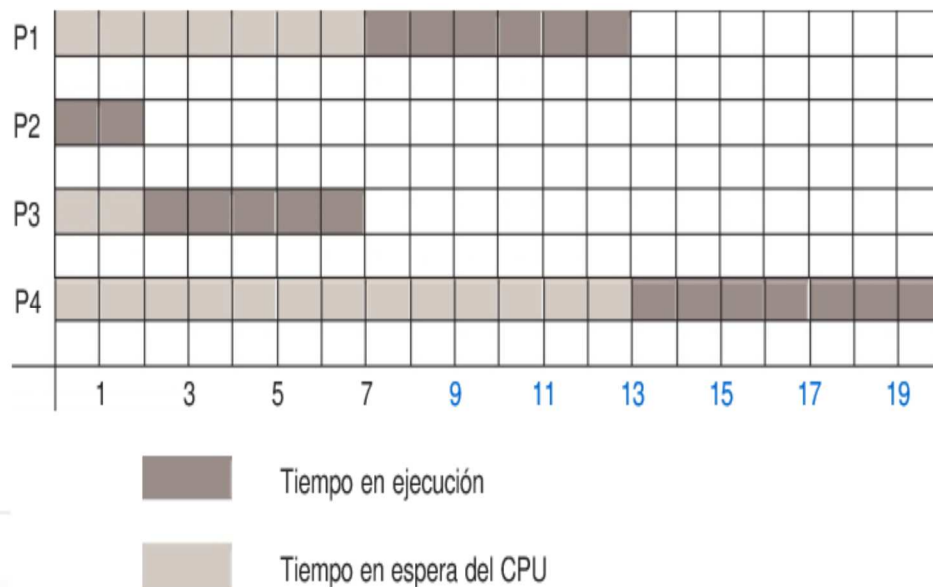
La ejecución en el tiempo usando este algoritmo de planificación se representa en el siguiente gráfico.



Supongamos que llegan al sistema 4 procesos en el orden que se presenta a continuación y con la longitud que se indica.

Proceso	Duración	Tiempo de llegada
P1	6	0
P2	2	0
P3	5	0
P4	7	0

La ejecución en el tiempo usando este algoritmo de planificación se representa en el siguiente gráfico.



2. A continuación veremos la gestión de procesos **apropiativos**. (Una vez que se le ha otorgado la CPU a un proceso, le puede ser retirada):

- **SRTF (Short Remaining Time First).**

Es similar al SJF, con la diferencia de que si un nuevo proceso pasa a listo se activa el dispatcher para ver si es más corto que lo que queda por ejecutar del proceso en ejecución. Si es así, el proceso en ejecución pasa a listo y su tiempo de estimación se decrementa con el tiempo que ha estado ejecutándose. Es un proceso **apropiativo** (hay procesos que se apropian de tiempos de CPU de otros procesos).

Supongamos que llegan al sistema 4 procesos en el orden que se presenta a continuación y con la longitud que se indica.

Proceso	Duración	Tiempo de llegada
P1	5	0
P2	5	1
P3	2	2
P4	2	3

La ejecución en el tiempo usando este algoritmo de planificación se representa en el siguiente gráfico.



- En el tiempo de llegada 0 sólo está disponible el P1 con lo cual empieza a hacer uso de la CPU.
- En el tiempo 1 llega el P2 que tiene de duración 5, como al P1 le quedan 4 se sigue ejecutando, por ser menor su tiempo de ejecución.
- En el tiempo 2 llega el P3 que como tiene de duración 2 y a P1 le quedan 3, P3 se apropia de la CPU y pasa ejecutarse.
- En el tiempo 3 llega P4 que tiene de duración 2, como a P3 le queda uno se sigue ejecutando hasta terminarlo.
- Cuando termina P3 entra a ejecución P4 que lo ejecuta hasta terminar.
- Cuando termina P4, nos quedaba el P2 que aún no ha empezado a ejecutarse con una duración de 5 y el P1 que le quedaba por ejecutar 3 tiempos de CPU. Con lo cual se coge el que menos tiempo necesite de CPU que es el P1, ejecutándolo hasta el final.
- Para finalizar pone en ejecución el P2, ejecutándolo hasta finalizar.

Si os fijáis en todo el tiempo de ejecución de la CPU, hay una apropiación de la CPU en el tiempo 2 por parte del proceso P3, el cual le quita la CPU al proceso P1 (debido a que en ese momento el proceso P3 tiene un tiempo de ejecución de 2 y el que está ejecutándose le quedan 3).

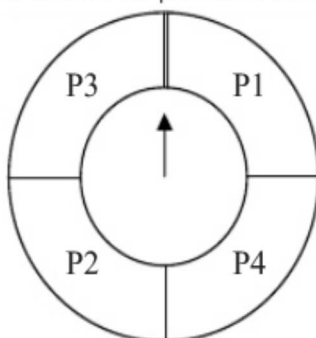
**RR (Round Robin).**

El Round Robin viene siempre acompañado por un slice de tiempo (Quantum), es decir, un tiempo determinado que tienen los procesos para ejecutar; finalizado ese tiempo el proceso abandona la CPU. Debido a esta característica, el algoritmo es **apropiativo**. Obviamente que el proceso puede abandonar antes la CPU en caso de que finalice o se bloquee por una E/S; caso contrario (que se le termina el tiempo), vuelve a la cola de listos. La cola de procesos es **FIFO** (primero en entrar primero en salir) y se estructura como una **cola circular** (cuando llega al último vuelve a empezar con el primero).

Supongamos que llegan al sistema 4 procesos en el orden que se presenta a continuación y con la longitud que se indica

Proceso	Duración	Tiempo de llegada
P1	3	0
P2	2	2
P3	3	3
P4	4	1

Y se tiene un quantum de 2; la cola circular de procesos listos quedaría de la siguiente manera:



La flecha indica el próximo proceso a ejecutarse y gira en sentido de las manecillas del reloj, por lo que el primer proceso a ejecutarse será P1, el segundo P4 y así sucesivamente.

La ejecución en el tiempo usando este algoritmo de planificación se representa en el siguiente gráfico.



Primero se ejecuta P1 que es el primero que llega (cola FIFO, primero que entra primero que se ejecuta), y se ejecuta dos tiempos de CPU (quantum 2). Al terminar su tiempo (aún le queda por ejecutar 1 tiempo más) lo pasa a listo (en la posición que tenía en la cola FIFO) y ejecuta el siguiente que es P4 (ya que es el siguiente en orden de llegada), lo ejecuta los dos tiempos de CPU y lo pasa a listo (aún le queda por ejecutar 2 tiempos de CPU). El siguiente en entrar a ejecución es el P2, lo ejecuta los 2 tiempos del quantum y como este sólo tenía de duración 2 pasa a finalizado y desaparece de la cola de Listos para ejecutarse (cola FIFO). El siguiente en ejecutarse es el P3 consumiendo su quantum y pasando a listo, quedándole otros 2 tiempos por ejecutar. Como la cola FIFO se acabó se sitúa de nuevo al principio repitiéndose todo el proceso.