

Tips para hacer los diagramas

- Para calcular la cantidad de columnas: sumamos la cantidad de tiempo de CPU
- Debemos marcar las llegadas de cada proceso con ">"
- Tiempo de Retorno (TR): Tiempo desde que llega el proceso a R QUEUE, hasta que finaliza completamente el proceso.  
Podemos calcularlo como:  
Limite Superior - Limite Inferior + 1  
O tambien lo podemos pensar como  
Nro de tiempo que finalizo - numero de tiempo que arranco + 1
- Tiempo de Espera (TE): Tiempo que tuvo que esperar para ser ejecutado.  
Para chequear que el tiempo de espera este bien, podemos hacer:  
TR - Tiempo CPU

La columna Llegada, indica el orden en el que llegan los procesos y debemos encolarlos en la R QUEUE.

TPR (Tiempo Promedio de Respuesta): SUMAMOS TR Y LO DIVIDIMOS POR LA CANTIDAD DE PROCESOS  
TPE (Tiempo Promedio de Ejecución): SUMAMOS TE Y LO DIVIDIMOS POR LA CANTIDAD DE PROCESOS

FCFS (First Come, First Served)

- Ejecución: Los procesos se ejecutan en el orden en que llegan.
- Prioridad: No tiene en cuenta el tiempo de ejecución ni la prioridad.
- Ventaja: Simple y fácil de implementar.
- Desventaja: Puede causar el problema de convoy effect (procesos cortos esperando a que terminen procesos largos).

Proceso	Llegada	CPU	Prioridad	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	TR	TE
P1	0	8	0	>																								24	0
P2	1	5	2		>																							9	0
P3	2	3	1			>																						8	0
P4	3	7	2				>																					13	6
P5	4	2	3					>																				10	0

En FCFS ejecuta los procesos en orden (a medida de que llegan a la R QUEUE), y un proceso se va a ejecutar hasta que termine.  
Cuando termina un proceso, en el proximo tiempo de cpu inicia el otro proceso que est en la R QUEUE (si es que hay otro proceso)

SJF (Shortest Job First)

- Ejecución: Los procesos con menor tiempo de ejecución son atendidos primero.
- Prioridad: Se basa en la duración estimada del trabajo (prioriza trabajos más cortos).
- Ventaja: Minimiza el tiempo promedio de espera.
- Desventaja: Requiere conocer los tiempos de ejecución de los procesos de antemano, puede provocar inanición de procesos largos.

Proceso	Llegada	CPU	Prioridad	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	TR	TE
P1	0	8	0																									24	0
P2	1	5	2																									9	0
P3	2	3	1																									8	0
P4	3	7	2																									13	6
P5	4	2	3																									10	0

Ejecuta primero los procesos con menor tiempo de CPU. Siempre va a ejecutar el proceso 1, y despues va a ejecutar aquellos procesos mas cortos. Con SJF los procesos mas cortos, seran ejecutados primero.  
DATO: Los encola en orden (por llegada), pero en la ejecucion le da prioridad a los mas cortos.

Round Robin con quantum = 4 y Timer fijo

- Ejecución: Los procesos se ejecutan durante un intervalo de tiempo fijo (quantum). Si no terminan, se colocan al final de la cola.
- Prioridad: Todos los procesos tienen la misma prioridad; el quantum fija la duración de cada turno.
- Ventaja: Proporciona equidad (ningún proceso queda bloqueado por completo).
- Desventaja: Puede haber overhead por cambios de contexto si el quantum es demasiado pequeño.

Proceso	Llegada	CPU	Prioridad	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	TR	TE
P1	0	8	0																									24	0
P2	1	5	2																									9	0
P3	2	3	1																									8	0
P4	3	7	2																									13	6
P5	4	2	3																									10	0

RR con timer fijo: Se le asigna una determinada cantidad de quantum, el cual, si a un proceso le sobra quantum, otro proceso usa la cantidad de quantum restante hasta llegar a 0. Cuando el quantum llega a 0, se genera un cambio de contexto, es decir, vuelve devuelta a seleccionar otro proceso de una cola

Round Robin con quantum = 4 y Timer Variable

- Ejecución: Similar al Round Robin con quantum fijo, pero si un proceso termina antes de consumir el quantum, el timer ajusta dinámicamente al siguiente proceso.
- Prioridad: Equidad similar al Round Robin fijo, pero con menor desperdicio de tiempo.
- Ventaja: Reduce el tiempo de espera innecesario y mejora el uso del CPU.
- Desventaja: Complejidad algo mayor al gestionar el timer variable.

cada vez que termina un quantum, hay que encolar el que acabas de ejecutar (si no termino). Pero si llega un proceso en ese instante, encolas primero el que llega y dsp el que estaba con la cpu

Si mientras esta ejecutando se termina el quantum y llega otro proceso: encola primero el proceso que llega, y despues encola el proceso que venia ejecutando

RR Timer Variable con Quantum=4  
Los tiempos empeoraron, pero nos aseguramos que no haya inanición (inanición=un proceso no sea ejecutado nunca)

Resumen:

- FCFS: Se ejecuta en orden de llegada.
- SJF: Se ejecuta en función del tiempo más corto primero.
- Round Robin (Timer Fijo): Cada proceso tiene un tiempo fijo de ejecución antes de ser interrumpido.
- Round Robin (Timer Variable): Igual que el anterior, pero el tiempo se ajusta dinámicamente si un proceso termina antes.

RESUMEN RÁPIDO			
Algoritmo	Apropiativo	Criterio de Ejecución	Nota Importante
FCFS	No	Orden de llegada	Ejecución completa por proceso (no se interrumpe).
SJF	No	Menor tiempo de CPU	Selecciona proceso más corto (por tiempo de CPU).
RR Variable	Si	Quantum fijo (tiempo sobrante no transferido).	Igualdad de quantum para todos los procesos.
RR Fijo	Si	Quantum fijo (sobra transfendiendo al siguiente).	Usa el quantum restante del proceso previo.
Prioridades	Si	Prioridad (menor número = mayor prioridad).	Los procesos de alta prioridad interrumpen a otros.

Algoritmo de prioridades:

Vamos a tener una cola (queue) por cada nivel de prioridad

A menor prioridad, mayor necesidad que se ejecute.

Es decir, menor numero de prioridad, mayor prioridad para ejecutarse

Este algoritmo es apropiativo, es decir si en el medio de la ejecucion de un proceso, entra un proceso de mayor prioridad, se va a sacar el proceso que se estaba ejecutando y va a ejecutar ese el que tenga mayor prioridad

Proceso	Llegada	CPU	Prioridad	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	TR	TE
P1	0	9	3	>																								24	15
P2	1	5	2		>																							9	3
P3	2	3	1			>																						8	0
P4	3	7	2				>																					13	6
Prioridad				Queue1	0																								
				Queue2	0	0	4																						
				Queue3	0	0	0																						

SRTF (Shortest Remaining Time First)

Se va a seleccionar el proceso, cuya siguiente rafaga de cpu sea la mas corta entre todos los que esten encolados en la R QUEUE

Es decir, mientras un proceso se esta ejecutando, y llega otro proceso con la rafaga mas corta, el proceso con la rafaga mas corta va a interrumpir al proceso que se venia ejecutando

Es la version apropiativa del SJF

Proceso	Llegada	CPU	Prioridad	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	TR	TE
P1	0	9	3	>																								24	15
P2	1	5	2		>																							9	3
P3	2	3	1			>																						8	0
P4	3	7	2				>																					13	6
SRTF				Queue	1	0	0	4																					