

ALGORITMO ROUND ROBIN

El algoritmo de Round Robin es **apropiativo**.

Este algoritmo es rotativo, lo que quiere decir que a cada proceso se le da unos instantes de tiempo para ejecutarse, y si no termina dentro de esos instantes de tiempo pasara a la cola en el caso que haya otro proceso esperando.

Dentro del Quantum puede estar ejecutándose sin que nadie le interrumpa, pero en el momento en el que el proceso cumple con su quantum, si hay un proceso que esté esperando, el proceso que se está ejecutando pasa a la cola y el proceso que estaba esperando avanza para ejecutarse.

Round Robin (con Quantum=2)

(Apropiativo)

Proceso	Llegada	Tiempo de CPU
A	0	10
B	1	3
C	5	5
D	7	1

Estado de cada proceso

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	↓1	2			3	4				5	6				7	8		9	10F
B		↓	1	2			3F												
C						↓		1	2				3	4			5F		
D								↓				1F							

Proceso ejecutándose y cola de procesos listos

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
CPU	↓A	A	B	B	A	A	BF	C	C	A	A	DF	C	C	A	A	CF	A	AF
1º		↓B	A	A	B	B	C	A	A	D	D	C	A	A	C	C	A		
2º						C	A	↓D	D	C	C	A							
3º																			

↓ Proceso llega y está listo para ejecutarse en ese instante

F Proceso termina en ese instante

Proceso se está ejecutando (# representa el instante de ejecución)

Proceso está esperando en la cola de procesos listos

ALGORITMO POR PRIORIDADES FIJAS

Es un algoritmo **apropiativo**. Tenemos una lista de procesos que van llegando en diferentes tiempos y necesitan distintos tiempos de CPU para ejecutarse. Tienen asociada una prioridad, que es la que marca qué proceso es el más importante. Si llega un proceso con una prioridad alta y se está ejecutando un proceso con una prioridad inferior, este parará su ejecución para dar paso al proceso que tiene más prioridad.

Por prioridades fijas

(Apropiativo)

Proceso	Llegada	Tiempo de CPU	Prioridad
A	0	10	1
B	1	3	3
C	5	5	2
D	7	1	4

Estado de cada proceso

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	↓1				2							3	4	5	6	7	8	9	10F
B		↓1	2	3F															
C						↓1	2		3	4	5F								
D								↓1F											

Proceso ejecutándose y cola de procesos listos según prioridades

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
CPU	↓A	↓B	B	BF	A	↓C	C	↓D	C	C	CF	A	A	A	A	A	A	A	AF
1º		A	A	A		A	A	C	A	A	A								
2º								A											
3º																			

↓ Proceso llega y está listo para ejecutarse en ese instante

F Proceso termina en ese instante

Proceso se está ejecutando (# representa el instante de ejecución)

Proceso está esperando en la cola de procesos listos

ALGORITMO SRTF (EL TIEMPO MÁS CORTO PRIMERO)

Es un algoritmo **apropiativo**. El proceso que llega tiene un tiempo de ejecución más corto que aquel proceso que se está ejecutando. Este algoritmo no se puede implementar en la vida real, porque no se puede estimar el tiempo de CPU que va a necesitar un proceso para terminar, ya que no sabe cuanto tiempo va a estar ejecutándose un proceso.

El menor tiempo restante primero

SRTF (Shortest Reamining Time First)

(Apropiativo)

Proceso	Llegada	Tiempo de CPU
A	0	10
B	1	3
C	5	5
D	7	1

Estado de cada proceso

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	↓1				2							3	4	5	6	7	8	9	10F
B		↓1	2	3F															
C						↓1	2		3	4	5F								
D								↓1F											

Proceso ejecutándose y cola de procesos listos

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
CPU	↓A	↓B	B	B	A	↓C	C	↓D	C	C	CF	A	A	A	A	A	A	A	AF
1º		A	A	A		A	A	C	A	A	A								
2º								A											
3º																			

↓ Proceso llega y está listo para ejecutarse en ese instante

F Proceso termina en ese instante

Proceso se está ejecutando (# representa el instante de ejecución)

Proceso está esperando en la cola de procesos listos

ALGORITMO SJF (EL TRABAJO MÁS CORTO PRIMERO)

Es un algoritmo **no apropiativo**. La forma de establecer que proceso de los que hay listos es el que tiene prioridad es el más corto primero (el que tenga menos tiempo de CPU)

El trabajo más corto primero

SJF (Shortest Job First)

(No apropiativo)

Proceso	Llegada	Tiempo de CPU
A	0	10
B	1	3
C	5	5
D	7	1

Estado de cada proceso

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	↓1	2	3	4	5	6	7	8	9	10F									
B		↓										1	2	3F					
C						↓									1	2	3	4	5F
D								↓			1F								

Proceso ejecutándose y cola de procesos listos

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
CPU	↓A	A	A	A	A	A	A	A	A	AF	DF	B	B	BF	C	C	C	C	CF
1º		↓B	B	B	B	B	B	↓D	D	D	B	C	C	C					
2º						↓C	C	B	B	B	C								
3º								C	C	C									

↓ Proceso llega y está listo para ejecutarse en ese instante

F Proceso termina en ese instante

Proceso se está ejecutando (# representa el instante de ejecución)

Proceso está esperando en la cola de procesos listos

ALGORITMO FCFS/FIFO (PRIMERO EN LLEGAR, PRIMERO EN SER SERVIDO)

Es un algoritmo **no apropiativo**. Es como una cola en el que el primero que llega se ejecuta, y hasta que no termina no se ejecuta el siguiente. Se van ordenando en el orden que van llegando.

El primero en llegar, primero en ejecutarse

FCFS (First Come, First Served) - FIFO (First In, First Out)
(No apropiativo)

Proceso	Llegada	Tiempo de CPU
A	0	10
B	1	3
C	5	5
D	7	1

Estado de cada proceso

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	↓1	2	3	4	5	6	7	8	9	10F									
B		↓									1	2	3F						
C						↓								1	2	3	4	5F	
D								↓											1F

Proceso ejecutándose y cola de procesos listos

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
CPU	↓A	A	A	A	A	A	A	A	A	AF	B	B	BF	C	C	C	C	CF	DF
1º		↓B	B	B	B	B	B	B	B	B	C	C	C	D	D	D	D	D	
2º						↓C	C	C	C	C	D	D	D						
3º								↓D	D	D									

- ↓ Proceso llega y está listo para ejecutarse en ese instante
- F Proceso termina en ese instante
- # Proceso se está ejecutando (# representa el instante de ejecución)
- Proceso está esperando en la cola de procesos listos

ESTADOS EN LA GESTIÓN DE PROCESOS

- Listo o preparado.** Puede pasar a estado de ejecución si el planificador del sistema operativo lo selecciona, esto es, cuando llegue su turno (según el orden de llegada o prioridad).
- En ejecución.** Se está ejecutando en el procesador en un momento dado.
- Bloqueado.** Está esperando la respuesta de algún otro proceso para poder continuar con su ejecución, por ejemplo, una operación de entrada/salida.



La **planificación no apropiativa** (en inglés, no preemptive) es aquella en la que, cuando a un proceso le toca su turno de ejecución, ya **no puede ser suspendido**; es decir, no se le puede arrebatar el uso de la CPU, hasta que el proceso no lo determina no se podrá ejecutar otro proceso. Este esquema **tiene sus problemas**, puesto que, si el proceso contiene **ciclos infinitos**, el resto de los procesos pueden quedar aplazados indefinidamente. Otro caso puede ser el de los procesos largos que penalizarían a los cortos si entran en primer lugar.

La **planificación apropiativa** (en inglés, preemptive) supone que el sistema operativo **puede arrebatar el uso de la CPU** a un proceso que esté ejecutándose. En la planificación apropiativa existe un reloj que lanza interrupciones periódicas en las cuales el planificador toma el control y se decide si el mismo proceso seguirá ejecutándose o se le da su turno a otro proceso.