

ALGORITMOS

DE

PLANIFICACIÓN

CRITERIO FCFS

- Servicio por orden de llegada (*First Come, First Served*).
- Cuando un proceso solicita uso de CPU (pasa al estado de 'Listo'), su BCP se pone el último en la cola FIFO de los procesos en espera de tiempo de CPU.
- Es un algoritmo **apropiativo**.
- Puede ser por **criterio de llegada inicial** o por **criterio de llegada a cola de listos**.
- Es un algoritmo poco eficiente en cuanto a tiempo de espera medio.
- Es un algoritmo que favorece la aparición de *efecto convoy*, es decir, predominio de los procesos que usan CPU frente a los que usan E/S.
- Es un algoritmo que penaliza a los procesos **más cortos**.

PROCESO	LLEGADA	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Criterio de llegada inicial:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A																															
B																															
C																															
D																															
E																															

EJECUCIÓN LISTO ESPERA E/S TERMINADO

Criterio de llegada a la cola de listos:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A																															
B																															
C																															
D																															
E																															

B C C C C C A A A A D A A D D E E E C C E E D D
 A A A A D D D E D D E C C D D
 D D D E E E E C C C D D

CRITERIO ROUND-ROBIN

- Periódicamente se genera una interrupción de reloj.
- Cuando se genera la interrupción, el proceso que está en ejecución se sitúa en la cola de 'Listo' y se selecciona el siguiente proceso.
- Es un algoritmo **no apropiativo**.
- Está especialmente diseñado para sistemas de tiempo compartido.
- La efectividad depende del **tamaño del cuanto** que escojamos.
- Dos criterios posibles: **orden de llegada** o el **orden de llegada a la cola FIFO**.

PROCESO	LLEGADA	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Cuanto q = 1:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	Yellow	Yellow	Blue	Yellow	Red	Red	Blue	Blue	Yellow	Blue	Blue	Blue	Blue	Yellow	Green																
B			Yellow	Blue	Yellow	Blue	Yellow	Blue	Blue	Blue	Yellow	Blue	Blue	Blue	Blue	Yellow	Blue	Blue	Blue	Yellow	Green										
C					Blue	Yellow	Blue	Blue	Blue	Yellow	Blue	Blue	Blue	Blue	Yellow	Blue	Blue	Blue	Yellow	Red	Red	Red	Red	Blue	Yellow	Yellow	Yellow	Yellow	Yellow	Green	
D							Blue	Yellow	Blue	Blue	Blue	Blue	Yellow	Blue	Blue	Blue	Blue	Yellow	Blue	Blue	Yellow	Blue	Yellow	Red	Red	Red	Red	Red	Yellow	Green	
E									Blue	Blue	Blue	Yellow	Blue	Blue	Blue	Blue	Yellow	Red	Red	Blue	Blue	Yellow	Blue	Yellow	Green						
<div> <div>A</div> <div>B</div> <div>C</div> <div>B</div> <div>D</div> <div>A</div> <div>C</div> <div>B</div> <div>E</div> <div>D</div> <div>A</div> <div>C</div> <div>B</div> <div>E</div> <div>D</div> <div>C</div> <div>B</div> <div>D</div> <div>E</div> <div>D</div> <div>E</div> <div>C</div> </div> <div> <div>A</div> <div>C</div> <div>B</div> <div>E</div> <div>D</div> <div>A</div> <div>C</div> <div>B</div> <div>E</div> <div>D</div> <div>C</div> <div>B</div> </div> <div> <div>C</div> <div>B</div> <div>D</div> <div>A</div> <div>C</div> <div>B</div> <div>E</div> <div>D</div> </div>																															

Cuanto q = 3:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
A	Yellow	Yellow	Yellow	Red	Red	Blue	Blue	Blue	Blue	Yellow	Yellow	Green																			
B			Blue	Yellow	Yellow	Yellow	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Yellow	Yellow	Green													
C					Blue	Blue	Yellow	Yellow	Yellow	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Red	Red	Red	Red	Yellow	Yellow	Yellow	Blue	Yellow	Green	
D							Blue	Blue	Blue	Blue	Blue	Yellow	Yellow	Yellow	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Yellow	Red	Red	Red	Red	Red	Yellow	Green		
E									Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Yellow	Red	Red	Blue	Yellow	Yellow	Green						
<div> <div>B</div> <div>C</div> <div>C</div> <div>A</div> <div>D</div> <div>D</div> <div>A</div> <div>A</div> <div>A</div> <div>D</div> <div>D</div> <div>B</div> <div>B</div> <div>B</div> <div>B</div> <div>E</div> <div>C</div> </div> <div> <div>A</div> <div>D</div> <div>D</div> <div>B</div> <div>B</div> <div>E</div> <div>C</div> </div> <div> <div>B</div> <div>B</div> <div>E</div> <div>C</div> <div>C</div> <div>C</div> <div>D</div> <div>C</div> <div>D</div> <div>D</div> <div>C</div> <div>D</div> <div>D</div> <div>D</div> </div>																															

CRITERIO SPN

- Primero el proceso más corto (*Shortest Process Next*).
- Es un algoritmo **apropiativo**.
- Se selecciona el proceso con menor tiempo esperado de ejecución.
- Posibilidad de inanición para los procesos largos.
- Puede ser por **lotes** o **interactivo**.
- En el caso de los procesos interactivos, miramos el tamaño de la siguiente ráfaga.
- En el caso de por lotes, se suman todos los tiempos de CPU de cada proceso y se selecciona el menor.

PROCESO	LLEGADA	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Proceso por lotes:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A																																	
B																																	
C																																	
D																																	
E																																	

Proceso interactivo:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A																																	
B																																	
C																																	
D																																	
E																																	
P _A			2						2																								
P _B			6																														
P _C									4		4		4																				
P _D									5		5		5																				
P _E									2		2																						

CRITERIO SRT

- Mejor tiempo restante (*Shortest Remaining Time*).
- Es una versión mejorada del algoritmo SPN.
- Es un algoritmo **no apropiativo**.
- Igual que su predecesor, puede ser por **lotes** o con procesos **interactivos**.
- En el caso de lotes, utilizamos la siguiente función de selección:

$$T_{\text{total}} - T_{\text{consumido}}$$

- Para el caso de procesos interactivos, usamos la siguiente fórmula:

$$S_{n+1} = \alpha t_n + (1 - \alpha)S_n$$

PROCESO	LLEGADA	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Proceso por lotes:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
A	Yellow	Yellow	Yellow	Red	Red	Yellow	Yellow	Green																										
B			Blue	Yellow	Yellow	Blue	Blue	Yellow	Yellow	Yellow	Yellow	Green																						
C					Blue	Blue		Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Yellow	Yellow	Yellow	Red	Red	Red	Red	Yellow	Yellow	Yellow	Yellow	Green
D							Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue	Yellow	Yellow	Blue	Blue	Yellow	Yellow	Yellow	Red	Red	Red	Red	Red	Yellow	Green						
E									Blue	Blue	Blue	Blue	Yellow	Yellow	Red	Red	Yellow	Yellow	Green															

Proceso interactivo para $S_1 = 3$:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A																																	
B																																	
C																																	
D																																	
E																																	

[A]: $S_1 = 3 // S_2 = 3$

[B]: $S_1 = 3$

[C]: $S_1 = 3 // S_2 = 3.6$

[D]: $S_1 = 3$

[E]: $S_1 = 3$

CRITERIO HRRN

- En HRRN elegimos el proceso con la tasa más alta.
- La tasa se calcula utilizando la siguiente fórmula:

$$tasa = \frac{\text{tiempo consumido esperando al procesador} + \text{tiempo de servicio esperado}}{\text{tiempo de servicio esperado}}$$

- El tiempo consumido esperando al procesador es el tiempo usado en el sistema hasta ese momento, esperando o ejecutando.
- El tiempo de servicio es el tiempo total de servicio requerido por el proceso. Si no se conoce, esta cantidad debe ser estimada mediante la fórmula (procesos interactivos). En otro caso, se toma como referencia la ráfaga de CPU anterior.
- Es un algoritmo **apropiativo**.

PROCESO	LLEGADA	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33		
A	■	■	■	■	■	■	■	■	■	■	■	■																							
B			■	■	■	■	■	■	■	■	■	■																							
C					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
D							■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
E									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

CRITERIO DE REALIMENTACIÓN MULTINIVEL

- Penaliza los trabajos que han estado ejecutándose durante más tiempo.
- No se conoce el tiempo de ejecución restante del proceso.
- Es un algoritmo de política FIFO **no apropiativo**.
- Una vez ejecuta en una cola de prioridad, baja a la siguiente.

PROCESO	LLEGADA	CPU	E/S	CPU
A	0	3	2	2
B	2	6		
C	4	4	4	4
D	6	5	5	1
E	8	2	2	2

Con 5 colas de prioridad:

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
A	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
B			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
D							■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
E									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

En este ejemplo, la primera cola de prioridad usa un *Round-Robin* con un turno de 2 cuantos de tiempo. Las demás usan el mismo algoritmo, pero con un turno de 1 cuanto de tiempo.