

ALGORITMO FCFS

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	>1	2	3	4	5	6	7	8	9<																9	0
P2	1	5	2	>									1	2	3	4	5<											13	8
P3	2	3	1	>														1	2	3<								15	12
P4	3	7	2	>																	1	2	3	4	5	6	7<	21	14
FCFS			R Queue	1	2	3	4																					14.5	8.5

- Columna de procesos → indica el proceso
- Columna de llegada → indica la llegada del proceso
- Columna CPU → indica ráfagas de CPU
- Columna de Prioridad → tachada porque en este algoritmo no se utiliza.

- Cálculo de columnas a utilizar → suma de ráfagas de CPU.
 - En este caso serían 24 (tiempos) ráfagas, por lo tanto 24 columnas
- Indicar con el símbolo ">" la llegada de cada proceso
- Indicar con el símbolo "<" al terminar cada proceso
- El FCFS va ejecutando los procesos a medida que vayan llegando
- En la última fila donde dice R Queue se van encolando el proceso que se va a ejecutar, cuando ya se ejecutó se tacha
- Al ejecutar cada proceso se anotan en cada celda los tiempos de CPU
 - Se llena la fila del P1 con 1 2 3 4 5 6 7 8 9<
- Durante la ejecución del P1 se encolan los demás procesos que fueron llegando
- La CPU en el tiempo 9 se encuentra ociosa
- Se necesita saber quien toca ejecutarse, será el siguiente en la cola, en este caso el P2
- El P2 se ejecuta sin tener que esperar
 - Se llena la fila con 1 2 3 4 5<
- Así hasta llenar el P4

- Cálculo de Tiempo de Espera
 - tiempo que tuvo que esperar para ser ejecutado
 - tiempo de retorno - cant CPU
 - ej P2: espero 8 unidades de tiempo (13-5)
- Cálculo de Tiempo de Retorno
 - tiempo desde que llegó a la cola el Proceso hasta que finaliza.
 - ej P2: 13 tiempos
- Cálculo total de TE → suma de TEs / cant Procesos = $34/4 = 8.5$
- Cálculo total de TR → suma de TRs / cant Procesos = $58/4 = 14.5$
 - Es útil que sea MÁS BAJO para aquellos lotes de proceso que necesitamos tener una respuesta rápida del proceso en sí mismo.
 - Por ejemplo los procesos que son CPU Bound → necesitamos que tenga un tiempo de retorno bajo para que termine el cálculo lo más rápido posible

ALGORITMO 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	>1	2	3	4	5	6	7	8	9<																9	0
P2	1	5	2	>												1	2	3	4	5<								16	11
P3	2	3	1		>								1	2	3<													10	7
P4	3	7	2			>															1	2	3	4	5	6	7<	21	14
SJF			Queue	1	2	3	4																					14	8

- Columna de procesos → indica el proceso
- Columna de llegada → indica la llegada del proceso
- Columna CPU → indica ráfagas de CPU
- Columna de Prioridad → tachada porque en este algoritmo no se utiliza.

- Marcamos con ">" los procesos que van llegando
- P1 se ejecuta sin ser interrumpido
 - En la fila del P1 se coloca >1 2 3 4 5 6 7 8 9<
- En el instante 9 el algoritmo se fija cual es el siguiente proceso a ejecutar el cual tendrá **la ráfaga de CPU más corta**
 - En este caso sería el P3. Este se ejecuta hasta que termina.
- Así hasta terminar

- Cálculo de Tiempo de Retorno
 - tiempo desde que llegó a la cola el Proceso hasta que finaliza.
 - ej P2: 16
 - Cálculo de Tiempo de Espera
 - tiempo que tuvo que esperar para ser ejecutado
 - tiempo de retorno - cant CPU
 - ej P2: 16-5=11
 - Cálculo total de TE → suma de TEs / cant Procesos = 32/4 = 8
 - Cálculo total de TR → suma de TRs / cant Procesos = 56/4 = 14
- En comparación al algoritmo FCFS se tiene una mejoría debido a que los tiempos son menores.

ALGORITMO RR-TV Q=4

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	>1	2	3	4												5	6	7	8					9<	24	15
P2	1	5	2	>				1	2	3	4											5<						19	14
P3	2	3	1	>								1	2	3<														9	6
P4	3	7	2				>								1	2	3	4						5	6	7<		20	13
RR-TV	Q=4		Queue	1	2	3	4	1	2	4	1																	18	12

- Columna de procesos → indica el proceso
- Columna de llegada → indica la llegada del proceso
- Columna CPU → indica ráfagas de CPU
- Columna de Prioridad → tachada porque en este algoritmo no se utiliza.

- Marcamos con ">" los procesos que van llegando
- Analizar qué está pasando en cada tiempo
 - En el tiempo 0 solo se encola en P1
- P1 se ejecuta al menos 4 veces debido a que el quantum = 4, se analiza que pasa en cada tiempo y luego se vuelve a la cola de listos
 - En la fila del P1 se coloca >1 2 3 4
 - En el tiempo 1 se encolo el P2, en el tiempo 2 se encolo el P3, en el tiempo 3 se encola P4
 - En el tiempo 4, el P1 sale de ejecución y se vuelve a encolar en el Queue
 - Luego empieza a ejecutarse el P2, se pone en la fila del P2 1 2 3 4
 - Se vuelve a encolar el P2
 - ...
- Así hasta terminar

- Cálculo de Tiempo de Retorno
 - tiempo desde que llegó a la cola el Proceso hasta que finaliza.
- Cálculo de Tiempo de Espera
 - tiempo que tuvo que esperar para ser ejecutado
 - tiempo de retorno - cant CPU
 - ej P1: $23 - 9 = 14$ (casillas donde no se ejecutó)
- Cálculo total de TE → suma de TEs / cant Procesos
- Cálculo total de TR → suma de TRs / cant Procesos

ALGORITMO RR-TF Q=4

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	>1	2	3	4									5	6	7	8				9<					21	12
P2	1	5	2		>			1	2	3	4									5<								16	11
P3	2	3	1			>						1	2	3<														9	6
P4	3	7	2			>								1							2	3	4		5	6	7<	21	14
RR-TF	Q=4		Queue	1	2	3	4	1	2	4	1	4																16.7	10.7

La diferencia entre TV y TF es que en el TV el Quantum va a ir variando para todos los procesos que es seleccionado para ejecutarse

- Columna de procesos → indica el proceso
- Columna de llegada → indica la llegada del proceso
- Columna CPU → indica ráfagas de CPU
- Columna de Prioridad → tachada porque en este algoritmo no se utiliza.

- Como $Q=4$, cada 4 tiempos se pone una “marca” para recordar que se tiene que cambiar de proceso
- Marcamos con “>” los procesos que van llegando
- Analizar qué está pasando en cada tiempo
 - En el tiempo 0 solo se encola en P1
- P1 se ejecuta 4 tiempos ($Q = 4$), se analiza que pasa en cada tiempo y luego se vuelve a la cola de listos
 - En la fila del P1 se coloca >1 2 3 4
 - En el tiempo 4, el P1 sale de ejecución y se vuelve a encolar en el Queue. Empieza a ejecutarse el P2, se pone en la fila del P2 1 2 3 4
 - Se vuelve a encolar el P2
 - Se ejecuta el P3 y termina, el $Q=1$, por lo tanto empieza a ejecutarse el P4 con $Q=1$ (es decir que se ejecuta una sola vez) y se vuelve a encolar

Queue	1	2	3	4	1	2	4
-------	---	---	---	---	---	---	---

- Se ejecuta el P1 con $Q=4$, se vuelve a encolar porque no terminó
- Se ejecuta el P2, termina en $Q=1$ por lo tanto se empieza a ejecutar P4.
- ...

- Cálculo de Tiempo de Retorno
 - tiempo desde que llegó a la cola el Proceso hasta que finaliza.
- Cálculo de Tiempo de Espera
 - tiempo que tuvo que esperar para ser ejecutado
 - tiempo de retorno - cant CPU
 - ej P1: $23-9 = 14$ (casillas donde no se ejecutó)
- Cálculo total de TE → suma de TEs / cant Procesos
- Cálculo total de TR → suma de TRs / cant Procesos

ALGORITMO PRIORIDADES

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	>1																2	3	4	5	6	7	8	9<	24	15
P2	1	5	2	>1					2	3	4	5<																8	3
P3	2	3	1		>1	2	3<																					3	0
P4	3	7	2			>							1	2	3	4	5	6	7<									13	6
Prioridad			Queue1	3																								12	6
			Queue2	2	2	4																							
			Queue3	1	1																								

- Columna de procesos → indica el proceso
- Columna de llegada → indica la llegada del proceso
- Columna CPU → indica ráfagas de CPU
- Columna de Prioridad → en este algoritmo se utiliza

- A menor prioridad, mayor necesidad de que se ejecute el proceso
- Se genera una cola por los distintos números de prioridades que se tiene
 - En este caso se tiene 3 prioridades, por lo tanto se van a generar tres colas
- Marcamos con ">" los procesos que van llegando
- Analizar qué está pasando en cada tiempo
 - En el tiempo 0 → llega el P1 con prioridad 3: se encola en el cola 3
 - Siempre se desencola primero de la Q1 luego de la Q2, como no hay nada en ellas se desencola lo de la Q3.
 - En el tiempo 1 → como P2 llega con mayor prioridad, se encola en la Q2; P1 deja de ejecutarse, se encola en Q3. Luego se ejecuta P2
 - En el tiempo 2 → como P3 llega con mayor prioridad, se encola en Q1; P2 deja de ejecutarse, se encola en Q2. Luego se ejecuta P3 y como es el de mayor prioridad no va a ser interrumpido.
 - Por otro lado en el tiempo 2, se encola en Q2 el P4
 - En el tiempo 5 → ya se terminó de ejecutar el P4, se empieza a ejecutar el P2 el cual se desencola de la Q2. Nadie lo va a interrumpir hasta terminar
 - En el tiempo 9 → se comienza a ejecutar el P4 por completo.
 - En el tiempo 16 → se termina de ejecutar el P1.

- Cálculo de Tiempo de Retorno
 - tiempo desde que llegó a la cola el Proceso hasta que finaliza.
- Cálculo de Tiempo de Espera
 - tiempo que tuvo que esperar para ser ejecutado
 - tiempo que terminó - tiempo que empezó + 1
- Cálculo total de TE → suma de TEs / cant Procesos
- Cálculo total de TR → suma de TRs / cant Procesos

ALGORITMO SRTF

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	>1																2	3	4	5	6	7	8	9	24	15
P2	1	5	2		>1				2	3	4	5	<															8	3
P3	2	3	1			>1	2	3	<																			3	0
P4	3	7	2				>						1	2	3	4	5	6	7	<								13	6
SRTF			Queue	1	2	3	4																					12	6

Se selecciona el proceso cuya siguiente rafaga de CPU sea la más corta entre los que están encolados en la cola de listos.
Es la versión apropiativa del SJF.

- Columna de procesos → indica el proceso
- Columna de llegada → indica la llegada del proceso
- Columna CPU → indica ráfagas de CPU
- Columna de Prioridad → en este algoritmo se utiliza

- Marcamos con ">" los procesos que van llegando
- Analizar qué está pasando en cada tiempo
 - En el tiempo 0 → se encola el P1 al llegar. Como no hay otro en la cola de listos, se ejecuta. Ahora P1 tendrá 8 de CPU.
 - En el tiempo 1 → se encola el P2 al llegar. Como en CPU tiene 5 y es menor a 8, deja de ejecutarse el P1 y se lo encola. Se empieza a ejecutar el P2 y ahora en CPU tiene 4
 - En el tiempo 2 → se encola el P3 al llegar. Como en CPU tiene 3 y es menor a 4, deja de ejecutarse el P2 y se lo encola. Se empieza a ejecutar el P3 y ahora en CPU tiene 2
 - En el tiempo 3 → se sigue ejecutando el P3 hasta terminar debido que el siguiente que llega tiene más tiempo en CPU (7). Se encola el P4.
 - En el tiempo 4 → se ejecuta el que tiene menos tiempo en CPU, por lo tanto se desencola y ejecuta el P2 hasta terminar.
 - En el tiempo 8 se desencola y ejecuta el P4 hasta terminar.
 - Por último se desencola y ejecuta el P1.

- Cálculo de Tiempo de Retorno
 - tiempo desde que llegó a la cola el Proceso hasta que finaliza.
- Cálculo de Tiempo de Espera
 - tiempo que tuvo que esperar para ser ejecutado
 - tiempo que terminó - tiempo que empezó + 1
- Cálculo total de TE → suma de TEs / cant Procesos
- Cálculo total de TR → suma de TRs / cant Procesos