

3)- Para los siguientes algoritmos de scheduling. Ejemplifique Algoritmos

FCFS : es un algoritmo no preescriptivo (no cede lugar ante pedidos del cpu). Se basa en una vez que arranca un proceso se debe continuar hasta terminarlo para darle paso al siguiente proceso. Mientras tanto si llegan nuevos procesos en el medio se van encolando para ser procesados luego de que termine el proceso actual. NO SE TOMAN LAS PRIORIDADES. EN CASO DE EMPATE 1* se toma el proceso que llego 1ro, si el empate continua. Se toma el del PID mas chico. TR= Tiempo desde la llegada hasta el fin del proceso. TE= Tiempo de espera donde el proceso espera a ser atendido.

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	24		TR	TE
P1	0	9	-	>1	2	3	4	5	6	7	8	9<																	9	0	
P2	1	5	-		>								1	2	3	4	5<												13	8	
P3	2	3	-			>												1	2	3<									15	12	
P4	3	7	-				>														1	2	3	4	5	6	7<		21	14	
FCFS(First Coome First Served)			QUEUE	4	2	3	4																				PROMEDIO		14,5	8,5	

SJF : es un algoritmo no preescriptivo (no cede lugar ante pedidos del cpu). Se basa en una vez que arranca un proceso se debe continuar hasta terminarlo para darle paso al siguiente proceso. Mientras tanto si llegan nuevos procesos en el medio se van encolando para ser procesados luego. El una vez terminado el proceso actual se verifica cual de los procesos encolados tienela rafaga de uso de procesador mas corta, y ese proceso es el que sigue para ser ejecutado.

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	24		TR	TE
P1	0	9	-	>1	2	3	4	5	6	7	8	9<																	9	0	
P2	1	5	-		>											1	2	3	4	5<									16	11	
P3	2	3	-			>							1	2	3<														10	7	
P4	3	7	-				>														1	2	3	4	5	6	7<		21	14	
SJF (Shortest Job First)			QUEUE	4	2	3	4																					PROMEDIO	14	8	

RR - TV (Round River - Tiempo Variable) : es un algoritmo no preescriptivo (no cede lugar ante pedidos del cpu). **Similar al FCFS en cuanto al uso de la cola.** Una vez que arranca un proceso se debe continuar hasta terminarlo o que se terminen los **QUANTUMS** de la operacion para darle paso al siguiente proceso. Mientras tanto si llegan nuevos procesos en el medio se van encolando para ser procesados con la misma logica que en **FCFS**. Luego de que termine el proceso actual(ya sea por finalizacion o por Quantum), se prosigue con el siguiente de la cola. Si el siguiente proceso finaliza antes que el quantum, el quantum restante se desecha. inicando el contador nuevamente. ESTE ALGORITMO NOS EVITA LA INANICION DE PROCESOS al ejecutar todos en una primera pasada.

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	24		TR	TE
P1	0	9	-	>1	2	3	4												5	6	7	8					9<			24	15
P2	1	5	-		>			1	2	3	4									6			5<							19	14
P3	2	3	-			>						1	2	3<																9	6
P4	3	7	-				>								1	2	3	4						5	6	7<				20	13
RR - TV Q=4			QUEUE	1	2	3	4	4	2	4	4																PROMEDIO			18	12

RR - TF (Round River - Tiempo Fijo) : es un algoritmo no preescriptivo (no cede lugar ante pedidos del cpu). Similar al FCFS en cuanto al uso de la cola. Una vez que arranca un proceso se debe continuar hasta terminarlo o que se terminen los **QUANTUMS** de la operacion para darle paso al siguiente proceso. Mientras tanto si llegan nuevos procesos en el medio se van encolando para ser procesados con la misma logica que en FCFS. Luego de que termine el proceso actual(ya sea por finalizacion o por Quantum), se prosigue con el siguiente de la cola. Si el siguiente proceso finaliza antes que el quantum, el quantum restante se utiliza para el siguiente proceso.

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	24		TR	TE
P1	0	9	-	>1	2	3	4									5	6	7	8				9<						21	12	
P2	1	5	-		>			1	2	3	4									5<									16	11	
P3	2	3	-			>						1	2	3<															9	6	
P4	3	7	-				>								1						2	3	4		5	6	7<		21	14	
RR - TF Q=4			QUEUE	1	2	3	4	4	2	4	1	4															PROMEDIO	16,75	10,75		

Prioridades : Este algoritmo maneja N cantidad de colas (1 por cada numero de prioridad diferente). EL algoritmo es **apropiativo** eso quiere decir que cuando ingrese un proceso de mayor prioridad (numero 1>prioridad) entonces se encola el proceso actual y se ejecuta el mas prioritario. TENER EN CUENTA: Los procesos van encolados en la cola de **SU** 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	24		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
Prioridades			QUEUE 1	3																									PROMEDIO	12	6
			QUEUE 2	2	2	4																									
			QUEUE 3	4	4																										

SRTF (Shortest remaining time first) : EL algoritmo es la version **apropiativo** del **SJF**(shortest job first). La logica de este algoritmo es tener en cuenta la cantidad de Tiempo de cpu de los procesos cuando se ingresan nuevos. Esto evalua si el proceso actual tiene menos tiempos de cpu que el nuevo ingresado haciendo que, si el nuevo es mas corto (en cuanto a rafagas de CPU), tome el lugar del cpu del que estaba. ESTE ALGORITMO ES BENEFICIOSO PARA LOS I/O BOUND

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	24		TR	TE
P1	0	9	-	>1																2	3	4	5	6	7	8	9<			24	15
P2	1	5	-		>1				2	3	4	5<																		8	3
P3	2	3	-			>1	2	3<																						3	0
P4	3	7	-				>						1	2	3	4	5	6	7<											13	6
SRTF			QUEUE	4	2	1	2	3	4																				PROMEDIO	12	6