

d2-u3: Ejercicios de Planificación de la CPU**Objetivo**

- Determinar la secuencia en que un conjunto de procesos, con diferentes tiempos de arribo y ráfaga de tiempo, se les asigna la CPU para un determinado algoritmo de planificación de procesos y calcular los tiempos medios para determinar, cuál de los algoritmos tiene mejor rendimiento.
- Determinar las predicciones de Burst time utilizando el algoritmo de promedio exponencial
- Determinar la utilidad del comando nice en sistemas Linux/UNIX

Descripción**Ejercicio 1:**

- a. Grafique el diagrama de Gantt correspondiente a la planificación de los procesos de la tabla 1 utilizando el algoritmo FCFS. Para esto debe llenar la tabla 2, la cual le permitirá ordenar los eventos de arribo y terminación, y los tiempos de uso de la CPU.

Tabla 2
FCFS

Time	CPU Use	Ready Queue (at end of time)	Event
0-6	A	B,C,D,E	B arrives in 2, C arrives in 4, D arrives in 5, E arrives in 6
6-8	B	C,D,E	
8-9	C	D,E	
9-11	D	E,F	F arrives in 10
11-15	E	F,G	G arrives in 12
15-18	F	G	
18-20	G		

Tabla 1 Información de la Tabla2 usando FCFS

Diagrama de Gantt**FCFS**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	A	A	A	A	A	B	B	C	D	D	E	E	E	E	F	F	F	G	G

Tabla 2 Diagrama de Gantt FCFS

b. Determine el tiempo de retorno y de espera de cada proceso.

	Process	Arrival Time	Burst Time	Priority	Return Time	Wait Time
	A	0	6	3	6	0
	B	2	2	1	6	4
	C	4	1	3	5	4
	D	5	2	4	6	4
	E	6	4	2	9	5
	F	10	3	1	8	5
	G	12	2	2	8	6
Promedio					6.86	4

Tabla 3 Procesos que arriban al sistema con FCFS

c. Determine el tiempo de retorno y de espera promedio para el conjunto de procesos.

El tiempo de retorno promedio es de 6.86 y el tiempo de espera promedio es de 4

d. Realizar las 3 tareas de arriba para los siguientes algoritmos:

- SJF apropiativo (preemptive)

Tabla SJF apropiativo

Time	CPU Use	Ready Queue (at end of time)	Event
0-2	A	B	B arrives in 2
2-4	B	A,C	A(4)pause in 2, C arrives in 4
4-5	C	A,D	D arrives in 5
5-7	D	A,E	E arrives in 6
7-11	A	F,E	F in 10
11-14	F	G,E	G in 12
14-16	G	E	
16-20	E		

*Tabla 4 Información de la Tabla2 usando SJF apropiativo***Diagrama de Gantt SJF apropiativo**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	A	B	B	C	D	D	A	A	A	A	F	F	F	G	G	E	E	E	E

Tabla 5 Diagrama de Gantt SJF apropiativo

Process	Arrival Time	Burst Time	Priority	Return Time	Wait Time
A	0	6	3	11	5
B	2	2	1	2	0
C	4	1	3	1	0
D	5	2	4	2	0
E	6	4	2	14	10
F	10	3	1	4	1
G	12	2	2	4	2
Promedio				5.43	2.57

Tabla 6 Procesos que arriban al sistema con SFJ apropiativo

El tiempo de retorno promedio es de 5.43 y el tiempo de espera promedio es de 2.57

- **RR quantum 1**

Tabla RR quantum 1

Time	CPU Use	Ready Queue (at end of time)	Event
0-1	A		
1-2	A	B	B arrives in 2
2-3	B	A	
3-4	A	B,C	C arrives in 4
4-5	B	C,A,D	D arrives in 5
5-6	C	A,D,E	E arrives in 6
6-7	A	D,E	
7-8	D	E,A	
8-9	E	A,D	
9-10	A	D,E,F	F arrives in 10
10-11	D	E,F,A	
11-12	E	F,A,G	G arrives in 12
12-13	F	A,G,E	
13-14	A	G,E,F	
14-15	G	E,F	
15-16	E	F,G	
16-17	F	G,E	
17-18	G	E,F	
18-19	E	F	
19-20	F		

Tabla 7 Información de la Tabla2 usando RR Quauntum 1

Diagrama de RR quantum 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	A	B	A	B	C	A	D	E	A	D	E	F	A	G	E	F	G	E	F

Tabla 8 Diagrama de Gantt RR Quantum 1

Process	Arrival Time	Burst Time	Priority	Return Time	Wait Time
A	0	6	3	14	8
B	2	2	1	3	1
C	4	1	3	2	1
D	5	2	4	6	4
E	6	4	2	13	9
F	10	3	1	10	7
G	12	2	2	6	4
Promedio				7.71	4.86

Tabla 9 Procesos que arriban al sistema con RR Quantum 1

El tiempo de retorno promedio es de 7.71 y el tiempo de espera promedio es de 4.86

- RR quantum 2**

Tabla RR quantum 2

Time	CPU Use	Ready Queue (at end of time)	Event
0-2	A	B	B arrives in 2
2-4	B	A,C	C arrives in 4
4-6	A	C,D,E	D arrives in 5, E arrives in 6
6-7	C	D,E,A	
7-8	D	E,A	
8-9	D	E,A	

9-11	E	A,F	F arrives in 10
11-13	A	F,E,G	G arrives in 12
13-15	F	E,G	
15-17	E	G,F	
17-19	G	F	
19-20	F		

Tabla 10 Información de la Tabla2 usando RR Quantum 2

Diagrama de RR quantum 2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	A	B	B	A	A	C	D	D	E	E	A	A	F	F	E	E	G	G	F

Tabla 11 Diagrama de Gantt RR Quantum 2

	Process	Arrival Time	Burst Time	Priority	Return Time	Wait Time
	A	0	6	3	13	7
	B	2	2	1	2	0
	C	4	1	3	3	2
	D	5	2	4	4	2
	E	6	4	2	11	7
	F	10	3	1	10	7
	G	12	2	2	7	5
Promedio					7.14	4.29

Tabla 12 Procesos que arriban al sistema con RR Quantum 2

El tiempo de retorno promedio es de 7.14 y el tiempo de espera promedio es de 4.29

- **Prioridad, siendo P0 la máxima y P4 la mínima**

Tabla Prioridad

Time	CPU Use	Ready Queue (at end of time)	Event
0-2	A	B(P1)	B arrives in 2
2-4	B	A(P3)C(P3)	C arrives in 4
4-6	A	C(P3) D(P4) E(P2)	D arrives in 5, E arrives in 6
6-10	E	C(P3) A(P3) D(P4) F(P1)	F arrives in 10
10-13	F	G(P2) C(P3) A(P3) D(P4)	G arrives in 12
13-15	G	C(P3) A(P3) D(P4)	
15-16	C	A(P3) D(P4)	
16-18	A	D(P4)	
18-20	D		

*Tabla 13 Información de la Tabla2 usando Prioridad***Diagrama de Prioridad**

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

A	A	B	B	A	A	E	E	E	E	F	F	F	G	G	C	A	A	D	D
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Tabla 14 Diagrama de Gantt Prioridad

Process	Arrival Time	Burst Time	Priority	Return Time	Wait Time
A	0	6	3	18	12
B	2	2	1	2	0
C	4	1	3	12	11
D	5	2	4	15	13
E	6	4	2	4	0
F	10	3	1	3	0
G	12	2	2	3	1
Promedio				8.14	5.29

Tabla 15 Procesos que arriban al sistema con Prioridad

El tiempo de retorno promedio es de 8.14 y el tiempo de espera promedio es de 5.29

Ejercicio 2:

Sea un algoritmo de planificación multi-colas con retroalimentación donde la primera cola (cola0) se gestiona con RR de $Q=1u$ de tiempo de CPU, y la segunda cola (cola1) se gestiona con un algoritmo SJF. La planificación entre colas es del tipo prioridades siendo la cola más prioritaria la 0. Un proceso pasará de la cola 0 a la cola 1 cuando se agote su quantum Q sin finalizar su ejecución. Los procesos nuevos, y los procedentes del estado BLOQUEADO, entran a la cola 0. Suponga, que las tareas del OS no consumen tiempo y que las operaciones de I/O se efectúan sobre un único dispositivo de I/O y don atendidas usando FCFS, obtener el diagrama de Gannt, el tiempo medio de retorno, y el tiempo medio de espera, al aplicar la siguiente carga de trabajo.

Proceso	Tiempo de llegada	Duración de ráfaga de CPU y de I/O
A	0	3 (CPU) + 1 (I/O) + 2 (CPU)
B	2	2 (CPU) + 2 (I/O) + 2 (CPU)
C	3	1 (CPU) + 1 (I/O) + 2 (CPU)

Process	Arrival Time	Burst Time	Return Time	Wait Time
A	0	3(CPU) + 1(I/O) + 2(CPU)	10	4
B	2	2(CPU) + 2(I/O) + 2(CPU)	10	4
C	3	1(CPU) + 1(I/O) + 2(CPU)	6	2
Promedio			8.67	3.33

Tabla 16 Procesos que arriban al sistema con multi-colas con retroalimentación

Tabla Multi-colas con retroalimentación

Time	CPU Use	Ready Queue (at end of time)	Event
0-1	A		A pasa a la cola 1 en 1
1-2	A	B(c0)	B arriva en 2

2-3	B	C(c0) A(c1)	C arriva en 3, B va a cola 1 en 3
3-4	C	A (c1) B(c1)	C se bloquea por I/O en 4
4-5	A	B(c1) C(c0)	A se bloquea por I/O en 5, C arriva a cola 0 en 5
5-6	C	B(c1) A(c0)	A arriva a cola 0 en 6, C va a cola 1 en 6
6-7	A	B(c1) C(c1)	A va a cola 1 en 7
7-8	B	C(c1) A(c1)	B se bloquea por I/O en 8
8-9	C	A(c1)	
9-10	A	B(c0)	B arriva a cola 0 en 10
10-11	B		B va a cola 1 en 10
11-12	B		

Tabla 17 Eventos de Multi-colas con retroalimentación

Diagrama de Gantt de Multi-colas con retroalimentación

	1	2	3	4	5	6	7	8	9	10	11	12
CPU:	A	A	B	C	A	C	A	B	C	A	B	B
I/O:					C	A			B	B		

Tabla 18 Diagrama de Gantt usando Multi-colas con retroalimentación

El tiempo de retorno promedio es de 8.67 y el tiempo de espera promedio es de 3.33

Ejercicio 3:

Suponga que a un proceso se le da una longitud de ráfaga esperada predeterminada de 5 unidades de tiempo cuando se crea por primera vez. Considere ahora un proceso P cuyas longitudes de ráfagas de CPU reales son 10, 10, 10, 1, 1, 1, 1 (aunque esta información no se conoce de antemano para ningún algoritmo de programación).

Suponiendo que el algoritmo de promedio exponencial para predecir ráfagas de tiempo de CPU usa $\alpha = 0.5$, calcule los tiempos de ráfaga esperados $e(1)$, $e(2)$, .. , $e(6)$ para este proceso (tenga en cuenta que $e(0) = 5$, el cuál es el valor inicial esperado).

Un algoritmo no puede tener a primera mano la información sobre las ráfagas de CPU que tienen los procesos, por lo tanto, tiene que estimar la longitud de las ráfagas en función de las ráfagas anteriores. En este caso se inicio con una estimación inicial de $e(0) = 5$. Para poder calcular las siguientes ráfagas se debe de usar la siguiente fórmula:

Siendo $e(t)$ el valor predicho para el tiempo actual, $a(t)$ el valor verdadero para el tiempo actual, y $e(t+1)$ el valor predicho para el siguiente tiempo. En este caso $\alpha = 0.5$

$$e(t+1) = \alpha e(t) + \alpha a(t).$$

Resolución:

$$e(1) = 0.5 e(0) + 0.5a(0) = 0.5(5) + 0.5(10) = 7.5$$

$$e(2) = 0.5 e(1) + 0.5a(1) = 0.5(7.5) + 0.5(10) = 8.75$$

$$e(3) = 0.5 e(2) + 0.5a(2) = 0.5(8.75) + 0.5(10) = 9.375$$

$$e(4) = 0.5 e(3) + 0.5a(3) = 0.5(9.375) + 0.5(1) = 5.1875$$

$$e(5) = 0.5 e(4) + 0.5a(4) = 0.5(5.1875) + 0.5(1) = 3.09375$$

$$e(6) = 0.5 e(5) + 0.5a(5) = 0.5(3.09375) + 0.5(1) = 2.046875$$

Ejercicio 4:

El comando nice se usa para establecer el valor nice de un proceso en Linux, así como en otros sistemas UNIX. Explique por qué algunos sistemas pueden permitir que cualquier usuario asigne a un proceso un valor $\text{nice} > 0$ y, sin embargo, permitir que solo el usuario root (o administrador) asigne valores $\text{nice} < 0$

Esto se debe a que los valores de nice value menores a 0 tienen una prioridad bastante alta, por lo tanto el sistema no permitirá que se asignen prioridades de esta magnitud a procesos de usuarios normales debido a que puede entorpecer el funcionamiento y el flujo de otros procesos dentro del sistema.