

Paradigmas de Programación Práctica N° 6

Trabajo Práctico Integrador

Cátedra Paradigmas de Programación - Sistemas Operativos

Esta práctica deberá resolverse en taller en las semanas del 02/10 y 09/10

Modele en UML el diagrama de clases correspondiente al diseño y a la implementación en Smalltalk de una aplicación que permita simular el proceso de planificación de CPU considerando los conceptos aplicados en el TP4 de Sistemas Operativos y utilizando las herramientas de la Programación Orientada a Objetos vistas en la cátedra de Paradigmas de Programación.

Considerar para dicho desarrollo SOLO el ejercicio N° 1 enunciado a continuación, los puntos a) y b).

Administración de CPU – Procesos

1. Se tienen los siguientes jobs que arriban al sistema en el instante 0 (cero).

Job	Tiempo de CPU
A	4 unidades
B	5 unidades
C	4 unidades
D	3 unidades

Realice mediante una tabla de estados, el desarrollo de los Jobs, según los siguientes algoritmos de *Scheduling*:

- FCFS (First Come, First Served)
- SJF (Shortest Job First) – No Apropiativo
- Round Robin con quantum = 2

Para cada algoritmo calcule el Tiempo de Espera TE, Tiempo de Retorno TR, Tiempo Medio de Espera y Tiempo Medio de Retorno.

2. Se tienen los siguientes jobs que arriban al sistema, según la siguiente tabla:

Job	Llegada	Tiempo de CPU
A	0	5 unidades
B	4	3 unidades
C	4	2 unidades
D	6	4 unidades

Realice mediante una tabla de estados, el desarrollo de los Jobs, según los siguientes algoritmos de *Scheduling*:

- FCFS (First Come, First Served).
- SJF (Shortest Job First) – No Apropiativo

- c) Round Robin con quantum = 2
- d) Round Robin con quantum = 3

Para cada algoritmo calcule el Tiempo de Espera TE, Tiempo de Retorno TR, Tiempo Medio de Espera y Tiempo Medio de Retorno.

3.

a) Sea el siguiente esquema de llegada de trabajos a ser procesados:

Job	Instante de llegada	Tiempo de CPU
A	0	6
B	2	3
C	2	5
D	3	4

Además, considere las siguientes operaciones de E/S:

Job	Instante de E/S	Duración de E/S
A	2	1
	4	2
C	1	2

Realice mediante una tabla de estados, el desarrollo de los Jobs, según los siguientes algoritmos de *Scheduling*:

- a) Round Robin con Q = 2
- b) Round Robin con Q = 3
- c) SJF (Shortest Job First) – No Apropiativo

Para cada algoritmo calcule el Tiempo de Espera TE, Tiempo de Retorno TR, Tiempo Medio de Espera y Tiempo Medio de Retorno.

b) Sea el siguiente esquema de llegada de trabajos a ser procesados:

Job	Instante de llegada	Tiempo de CPU
A	0	7
B	3	6
C	5	3
D	7	5
E	12	3

Realice mediante una tabla de estados, el desarrollo de los Jobs descritos en la tabla, aplicando el algoritmo SJF-Apropiativo o SRTF.

Para cada algoritmo calcule el Tiempo de Espera TE, Tiempo de Retorno TR, Tiempo Medio de Espera y Tiempo Medio de Retorno.

4. Supongamos que tenemos los siguientes jobs para ejecutar por una CPU:

Job	Hora de llegada	Tiempo de CPU	Prioridad
A	0	5	3
B	3	5	1
C	3	4	3
D	5	6	4
E	7	4	2

Además, considere las siguientes operaciones de E/S:

Job	Instante de E/S	Duración de E/S
1	2	3
	5	2
4	1	1
	4	2

Realice mediante una tabla de estados, el desarrollo de los Jobs, según los siguientes algoritmos de *Scheduling*:

- Round Robin con $Q = 3$
- Round Robin con $Q = 4$
- c) SJF (Shortest Job First) – No Apropiativo

Para cada algoritmo calcule el Tiempo de Espera TE , Tiempo de Retorno TR , Tiempo Medio de Espera y Tiempo Medio de Retorno.

5. Sea el siguiente esquema de llegada de trabajos a ser procesados con sus respectivas operaciones de E/S

Proceso	Llegada	Tiempo de CPU	de E/S	Prioridad Base
A	0	5	(3,2)	BAJA
B	2	4	(2,2)	ALTA
C	2	6		MEDIA
D	3	3	(1,1)	BAJA

➤ Orden de prioridad: >ALTA > MEDIA > BAJA.

- Envejecimiento: Las prioridades de los procesos aumentan a medida que pasa el tiempo. El método de aumento de prioridad se realiza cuando un proceso permanece en la *cola de listo* sin ejecutarse, cada X unidades de tiempo pasa a la prioridad siguiente y así sucesivamente hasta que tome la CPU, luego de dejar la CPU vuelven a su prioridad base. Cuando un proceso cambia la prioridad por envejecimiento, entra a la *cola de listo* último de su nueva prioridad (o sea, re re-encola en la correspondiente cola según su nueva prioridad).

Realice mediante una tabla de estados, el desarrollo de los Jobs, según los siguientes algoritmos de *Scheduling*:

- Round Robin con $Q = 3$, Envejecimiento de 5 unidades.
- Round Robin con $Q = 2$, Envejecimiento de 5 unidades

Indique el estado de los procesos en los instantes 5, 10, 15 y 20 con un diagrama de 5 estados posibles: A comenzar, Ejecutando, Listo, Bloqueado por E/S, Terminado; para los dos diagramas.

Para cada algoritmo calcule el Tiempo de Espera TE, Tiempo de Retorno TR, Tiempo Medio de Espera y Tiempo Medio de Retorno.

6. Sea el siguiente esquema de llegadas de trabajos a ser procesados con sus respectivas operaciones de E/S

JOB	Inst. Llegada	CPU	E/S (inst, dur)	Prioridad
A	0	6	(2,2) (4,1)	Alta
B	1	5	(3,2)	Media
C	2	4		Baja
D	3	5	(1,2) (4,1)	Media
E	5	4	(2,2)	Alta

Considere un algoritmo de administración de CPU Round Robin con QUANTUM de 3 unidades.

Realice mediante una tabla de estados, el desarrollo de los Jobs, según los siguientes algoritmos de Scheduling:

- a) Round Robin con Q = 3, Envejecimiento de 5 unidades.
- b) Round Robin con Q = 2, Envejecimiento de 5 unidades

Para cada algoritmo calcule el Tiempo de Espera TE, Tiempo de Retorno TR, Tiempo Medio de Espera y Tiempo Medio de Retorno.

Indique el estado de los procesos en los instantes 4, 9, 17 y 22 con un diagrama de 5 estados posibles: A comenzar, Ejecutando, Listo, Bloqueado por E/S, Terminado; para los dos diagramas.

7. Considerando el diagrama de estados de Unix, si un proceso necesita un recurso que estará disponible dentro de 5 horas y los recursos de memoria del sistema son escasos:

Donde cree que este proceso será alojado a la espera del recurso.
Una vez liberado el recurso indique los pasos a seguir hasta la ejecución.

8. Considerando el diagrama de estados de Unix, si un proceso no necesita para su ejecución tomar recursos de hardware, decir en que modo se ejecutaría.

9. Si un proceso se esta ejecutando en modo supervisor y hace una System Call para cambiar de modo, ¿a que modo pasa?