Universidad de Carabobo

Facultad de Ciencia y Tecnologia

Sistemas Operativos

Algoritmos de Planificación del Procesador

Victor Tortolero, 24.569.609

22 de marzo de 2016

1. First Come First Serve (FCFS)

Con este algoritmo los procesos se van ingresando en una cola FIFO(First In, First Out), al llegar el proceso se inserta su PCB en la cola, luego si el CPU esta libre toma el primer elemento de la cola y se elimina este de la cola. Como es no apropiativo, la CPU solo toma otro proceso de la cola al terminar el actual.

Para implantar este algoritmo solo se requiere una cola FIFO.

- **Ventajas**
  - Es fácil de entender e implementar.
  - Aunque normalmente es justo en como dedica tiempo de CPU a los procesos, los procesos largos hacen esperar a los cortos.
- **Desventajas**
  - El tiempo de espera es alto por lo que carece de rendimiento.
  - No apropiativo.

Supongamos que tenemos 3 procesos:

Proceso	Tiempo de Ráfaga
P_0	20
P_1	6
P_2	9

Si los procesos llegan en el siguiente orden, $P_0 \rightarrow P_1 \rightarrow P_2$, tendríamos el diagrama de gantt 1, y el tiempo de espera promedio seria $\frac{0+20+26}{3} = 15{,}333$.
En cambio si los procesos llegaran en el orden $P_1 \rightarrow P_2 \rightarrow P_0$, tendríamos el diagrama de gantt 2, y el tiempo de espera promedio seria $\frac{0+6+15}{3} = 8$.
En este algoritmo es importante el orden en el que llegan los procesos, y si el tiempo entre ellos varia de gran manera, entonces tendremos un tiempo de espera alto.

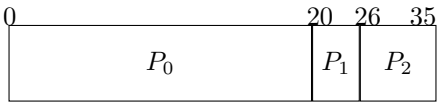

Figura 1: FCFS, Ejemplo 1

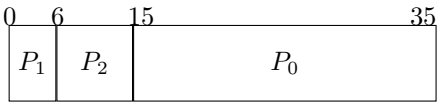

Figura 2: FCFS, Ejemplo 2

2. Shortest Job First (SJF)

Este algoritmo selecciona el proceso con el próximo tiempo de ejecución mas corto. Asocia con cada proceso la duración de la siguiente ráfaga de CPU del proceso. Cuando el CPU esta libre se le asigna el proceso que tiene la siguiente ráfaga de CPU mas corta. Si varios procesos tienen la misma duración de ráfaga, se usa el algoritmo FCFS para elegir el proceso.

Para implantar este algoritmo se requiere una cola, y una manera de saber la duracion de la siguiente ráfaga de CPU para los procesos en la cola, normalmente se usan métodos para hacer aproximaciones.

- **Ventajas**
  - S
- **Desventajas**
  - Difícil de implementar ya que es complicado conocer la duración de la siguiente ráfaga de CPU para un proceso.

Supongamos que tenemos 4 procesos:

Proceso	Tiempo de Ráfaga(ms)	Tiempo de llegada(ms)
P_0	20	3
P_1	6	2
P_2	24	0
P_3	8	1

Primero el CPU trataría el P_2 ya que es el primero en entrar a la cola, luego cuando llega el P_3 el algoritmo verifica y tenemos que el tiempo que le queda a P_2(23ms) es mayor al tiempo que le queda a P_3(8ms), por lo que el CPU trata a P_3. Siguiendo estas reglas tendríamos el diagrama de gantt 4.

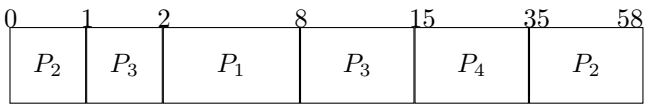

Figura 3: SJF, Ejemplo 1

3. Short Remaining Time First (SRTF)

Este algoritmo selecciona el proceso que esta en ejecución con otro que exija menor tiempo de ejecución. Consigue una buena eficiencia, ya que logra que la lista de procesos preparados sea lo más corta posible.

– **Ventajas**

- Es apropiativo, y eficiente.
- Presenta un buen tiempo promedio de servicio.

– **Desventajas**

- Los procesos largos no se ejecutaran mientras existan procesos cortos en la cola.

Supongamos que tenemos 4 procesos:

Proceso	Tiempo de Ráfaga(ms)	Tiempo de llegada(ms)
P_0	20	3
P_1	6	2
P_2	24	0
P_3	8	1

Primero el CPU trataría el P_2 ya que es el primero en entrar a la cola, luego cuando llega el P_3 el algoritmo verifica y tenemos que el tiempo que le queda a P_2(23ms) es mayor al tiempo que le queda a P_3(8ms), por lo que el CPU trata a P_3. Siguiendo estas reglas tendríamos el diagrama de gantt 4.

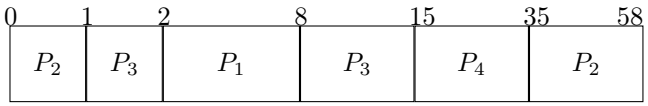

Figura 4: SJF, Ejemplo 1

4. Round Robin (RR)

Consiste en conceder a cada proceso en ejecución un determinado período de tiempo, denominado quantum (Q), transcurrido el cual, si el proceso no ha terminado, se le devuelve al final de la cola de procesos preparados y se pasa al proceso que este de primero en la cola. Esta interrupción periódica continúa hasta que el proceso termine su ejecución, formando una rueda de procesos que serán ejecutados cíclicamente hasta que terminen.

Para su implantación requiere una cola y el valor optimo de Q, el cual se determina según el tipo de sistema y el numero de procesos.

- **Ventajas**
  - Es justo con todos los procesos.
  - Es apropiativo.
- **Desventajas**
  - Si el valor de Q es mayor que el tiempo requerido por el proceso mas largo, se convierte en FCFS.
  - Si el valor de Q es muy pequeño se producen muchos cambios de contexto lo que es ineficiente.

Supongamos que tenemos 3 procesos:

Proceso	Tiempo de Ráfaga(ms)	Tiempo de llegada(ms)
P_0	5	0
P_1	3	1
P_2	8	2

Para Q = 4, y Q = 8 tenemos:

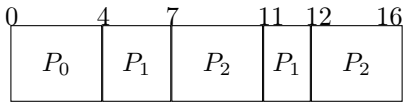

Figura 5: RR, Q = 4

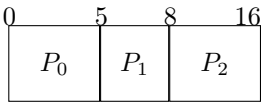

Figura 6: RR, Q = 8

5. Prioridades

Se asocia a cada proceso un numero de prioridad (mientras menor sea el número, mas alta la prioridad). Las prioridades pueden ser definidas interna o externamente. En el primer caso, el sistema operativo se basa en una serie de informaciones medibles para el cálculo y asignación de dichas prioridades (tiempo necesitado de procesador, necesidad de memoria, etc.). Los factores externos son asignados por otro programa o el usuario. Si hay varios procesos con la misma prioridad, se resuelve con FCFS.

Para su implantación se necesita una cola de prioridades, indicar si las prioridades se definirán de manera interna o externa y la razón de incremento si se usa envejecimiento.

- **Ventajas**
  - Puede ser apropiativo o no apropiativo.
- **Desventajas**
  - Si no se usa envejecimiento, un proceso con muy baja prioridad puede llegar a no ejecutarse nunca.

Supongamos que tenemos 3 procesos:

Proceso	Tiempo de Ráfaga(ms)	Tiempo de llegada(ms)	Prioridad
P_0	5	0	3
P_1	3	1	1
P_2	8	2	2

Tenemos el primer ejemplo sin envejecimiento, y el segundo ejemplo con envejecimiento T = 2 (Cada 2ms la prioridad disminuye en 1).

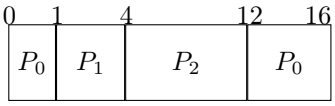

Figura 7: Prioridades, Sin envejecimiento

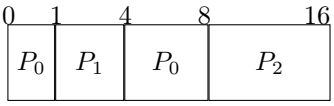

Figura 8: Prioridades, Envejecimiento T=2

6. Colas Multinivel

Con este algoritmo los procesos se agrupan y se asignan a diferentes colas, cada cola puede tener su propio algoritmo de planificación. Para elegir que cola usar se puede usar un algoritmo de prioridades sin envejecimiento,

Para su implantación se necesita una cola de prioridades, indicar si las prioridades se definirán de manera interna o externa y la razón de incremento si se usa envejecimiento.

- **Ventajas**
  - Es apropiativo.
  - Es muy adaptable a las necesidades del sistema, ya que cada cola puede ser gestionada de forma diferente.
- **Desventajas**
  - Si no se usa envejecimiento, un proceso con muy baja prioridad puede llegar a no ejecutarse nunca.

Supongamos que tenemos 3 procesos:

Proceso	Tiempo de Ráfaga(ms)	Tiempo de llegada(ms)	Prioridad
P_0	5	0	3
P_1	3	1	1
P_2	8	2	2

Tenemos el primer ejemplo sin envejecimiento, y el segundo ejemplo con envejecimiento T = 2 (Cada 2ms la prioridad disminuye en 1).

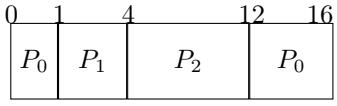

Figura 9: Prioridades, Sin envejecimiento

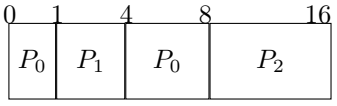

Figura 10: Prioridades, Envejecimiento T=2

Sistemas de Tiempo Real

Es un sistema en el que se requiere no solo que los resultados calculados sean correctos, sino que también se produzcan dentro de un periodo de tiempo especificado.

Glosario

- **Ráfaga CPU**: Tiempo de ejecución en CPU entre dos E/S.
- **Ráfaga E/S**: Tiempo entre solicitud y terminación de E/S.

[1, 2, 3, 4]

Referencias

[1] Info en web. Disponible en http://pachel.tripod.com/materias/material3sis.htm.

[2] Justo Sáez Arenas. Instituto nacional de tecnologias educativas y de formacion del profesorado (intef, españa). Disponible en http://mimosa.pntic.mec.es/~jsaez9/Clases/simr/B-Sist_Operativos/T7-Gestion%20de%20Procesos.pdf.

[3] Mercedes Fernández Redondo. Universitat jaume i. Disponible en http://www3.uji.es/~redondo/so/capitulo2_IS11.pdf.

[4] Abraham Silberschatz & Peter Baer Galvin & Greg Gagne. *Fundamentos de Sistemas Operativos, 7ma Edición*. McGraw Hill, 2005.