

Conceptos de Sistemas Operativos

Practica 4

Ramiro Cabral

October 15, 2023

Contents

1	Procesos	2
1.1	PCB (Process Control Block)	2
1.2	Tiempos de los procesos	2
2	Schedulers/Planificadores	2
2.1	Tipos de Schedulers	2
2.2	Apropiacion vs No apropiacion	4
3	Algoritmos de planificacion	4
3.1	FCFS (First Come First Served)	4
3.2	SJF (Shortest Job First)	4
3.3	SRTF (Shortest Remaining Time First)	4
3.4	RR (Round Robin)	5
	3.4.1 Timer Variable	5
	3.4.2 Timer Fijo	5
3.5	Prioridades	5
3.6	CPU + I/O	6
3.7	Criterios de desempate	6
4	Colas Multinivel	6
5	Planificacion con multiples procesadores.	6
5.1	Clasificaciones	7
5.2	Otra clasificacion:	7

1 Procesos

- Programa en ejecucion.
- Segun su historial de ejecucion, podemos clasificarlos en:
 - CPU bound (ligados a la CPU).
 - I/O bound (ligados a la E/S).

1.1 PCB (Process Control Block)

- Una por proceso.
- Contiene informacion del proceso.
- Es lo primero que se crea cuando se realiza un fork y lo ultimo que se desaloca cuando termina.

1.2 Tiempos de los procesos

- **Retorno:** Tiempo que transcurre entre que el proceso llega al sistema hasta que completa su ejecucion.
- **Espera:** Tiempo que el proceso se encuentra en el sistema esperando, es decir el tiempo que pasa sin ejecutarse. (Tiempo de retorno - Tiempo de CPU)
- **Promedios:** Tiempos promedio de los anteriores.

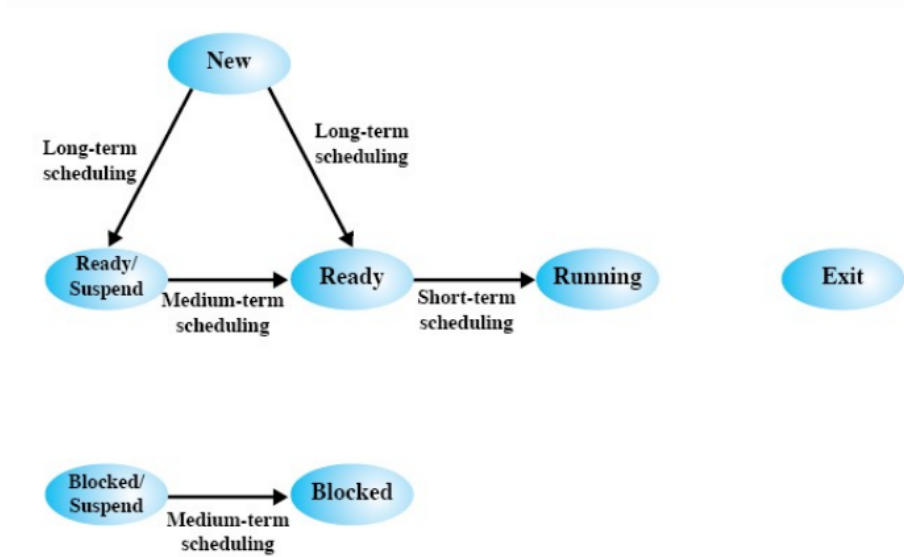
2 Schedulers/Planificadores

- Es la clave de la multiprogramacion.
- Esta diseñado para cumplir con los siguientes objetivos:
 - Menor tiempo de respuesta.
 - Mayor rendimiento.
 - Uso eficiente del procesador.

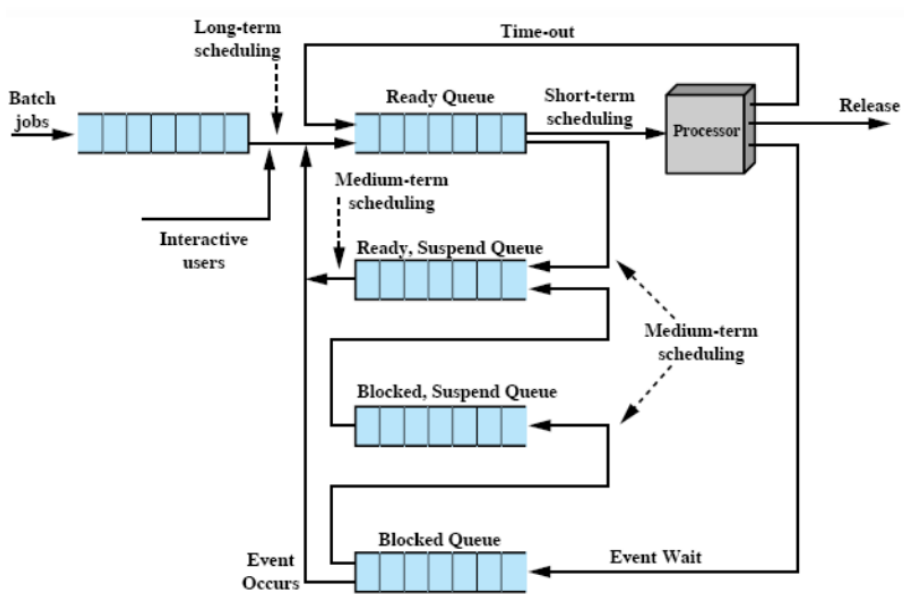
2.1 Tipos de Schedulers

- **Long term scheduler:** Admite nuevos procesos a memoria (controla el grado de multiprogramacion).

- **Medium term scheduler:** Realiza el swapping (intercambio) entre el disco y la memoria cuando el SO lo determina.
- **Short term scheduler:** Determina que proceso pasara a ejecutarse.



(1)



(2)

2.2 Apropiacion vs No apropiacion

- **Nonpreemptive:** Una vez que un proceso esta en estado de ejecucion, continua hasta que termina o se bloquea por algun evento (por ejemplo, I/O).
- **Preemptive:** El proceso en ejecucion puede ser interrumpido y llevado a la cola de ready
 - Mayor overhead pero mejor servicio.
 - Un proceso no monopoliza el procesador.

3 Algoritmos de planificacion

3.1 FCFS (First Come First Served)

- **Nonpreemptive.**
- Cuando hay que elegir un proceso para ejecutar, se selecciona el mas viejo.
- No favorece a ningun tipo de procesos, pero en principio podriamos decir que los CPU Bound terminan al comenzar su primer rafaga, mientras que los I/O Bound no.

3.2 SJF (Shortest Job First)

- **Nonpreemptive.**
- Politica que selecciona el proceso con la rafaga de CPU mas corta.
- Calculo basado en la ejecucion previa.
- Procesos cortos se colocan delante de procesos largos.
- Los procesos largos pueden sufrir starvation.

3.3 SRTF (Shortest Remaining Time First)

- Version **preemptive** de SJF.
- Selecciona el proceso al cual le resta menos tiempo de ejecucion en su siguiente rafaga.
- Favorece al los procesos I/O Bound.

3.4 RR (Round Robin)

- Política basada en un reloj.
- **Quantum (Q)** : medida que determina cuanto tiempo podra usar el procesador cada proceso:
- Cuando un proceso es expulsado de la CPU es colocado al final de la Ready Queue y se selecciona otro (FIFO circular).
- Existe un contador que indica las unidades de CPU en las que el proceso se ejecuto. Cuando el mismo llega a 0 el proceso es expulsado.
- El contador puede ser global o local (PCB de cada proceso).
- Existen dos variantes con respecto al valor inicial del contador cuano un proceso es asignado a la CPU:
 - Timer Variable.
 - Timer Fijo.

3.4.1 Timer Variable

- El contador se inicializa en Q cada vez que un proceso es asignado a la CPU.
- Es el mas utilizado.

3.4.2 Timer Fijo

- El contador se inicializa en Q cuando su valor es cero
- Se puede ver como un valor de Q compartido entre los procesos.

3.5 Prioridades

- Cada proceso tiene un valor que representa su prioridad.
- Se seleccina el proceso de mayor prioridad de los que se encuentran en la Ready Queue.
- Existe una Ready Queue para cada nivel de prioridad.
- Procesos de baja prioridad pueden sufrir starvation.
 - Solucion: permitir a un proceso cambiar su prioridad durante su ciclo de vida (**Aging o Penalty**).
- Puede ser un algoritmo *preemptive* o no.

3.6 CPU + I/O

- Ciclo de vida de un proceso: uso de CPU + operaciones de I/O.
- Cada dispositivo tiene su cola de procesos en espera, un scheduler por cada cola.
- Se considera I/O independiente de la CPU (DMA, PCI, etc). Tenemos uso de CPU y operaciones de I/O en simultaneo.

3.7 Criterios de desempate

1. Orden de llegada de los procesos.
2. **PID** de los procesos(el de menor PID se ejecuta primero).

4 Colas Multinivel

- La ready queue es dividida en varias colas (similar a prioridades).
- Los procesos se colocan en las colas segun una clasificacion que realice el sistema operativo.
- Cada cola posee su propio algoritmo de planificacion (**planificador horizontal**).
- A su vez existe un algoritmo que planifica las colas (**planificador vertical**).
- **Retroalimentacion:** un proceso puede cambiar de una cola a otra.

5 Planificacion con multiples procesadores.

- **Planificacion temporal:** que proceso y durante cuanto.
- **Planificacion espacial:** en que procesador ejecutar:
 - **Huella:** estado que el proceso va dejando en la cache de un procesador.
 - **Afinidad:** preferencia de un proceso para ejecutar en un procesador.
- La asignacion de procesos a un procesador puede ser:
 - **Estatica:** existe una afinidad de un proceso a una CPU.
 - **Dinamica:** la carga se comparte, se da un balanceo de carga.
- La politica puede ser:

- **Tiempo compartido:** se puede considerar una cola global o una cola local a cada procesador.
- **Espacio compartido:**
 - * **Grupos (threads).**
 - * **Particiones.**

5.1 Clasificaciones

- **Procesadores homogéneos:** todas las CPUs son iguales. No existe ventajas físicas sobre el resto.
- **Procesadores heterogéneos:** cada procesador tiene su propia cola, su propio clock y su propio algoritmo de planificación.

5.2 Otra clasificación:

- **Procesadores débilmente acoplados:** cada CPU tiene su propia memoria principal y canales.
- **Procesadores fuertemente acoplados:** comparten memoria y canales.
- **Procesadores especializados:** uno o más procesadores principales de uso general y uno o más procesadores de uso específico.