Algoritmos de planificación de sistemas operativos

Mario arnedo

Primero en llegar primero en ser servido. FCFS (First Come First Served).

▶ Idea general del algoritmo

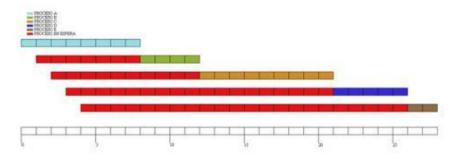
Este proceso se basa en donde el procesador ejecuta cada proceso hasta que termina, por tanto, los procesos que en cola de procesos preparados permanecerán encolados en el orden en que lleguen hasta que les toque su ejecución.

Primero en llegar primero en ser servido. FCFS (First Come First Served).

► Ejemplo de uso del procesador

Proceso A → Tiempo ejecución → Tiempo llegada → Tiempo finaliza → Tiempo retorno → Tiempo espera

Proceso	Tiempo de ejecución	Tiempo de Ilegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0	8	8	0
В	4	1	9	12	12-1 = 11	11-4=7
C	9	2	13	21	21-2 = 19	19 - 9 = 10
D	5	3	21	26	26-3 = 23	23 - 5 = 18
E	2	4	26	28	28-4 = 24	24 -2 = 22



Primero en llegar primero en ser servido. FCFS (First Come First Served).

Ventajas

- No apropiativa
- Es justa, aunque los procesos largos hacen esperar a los cortos
- Predecible
- El tiempo medio de los procesos es variado

Inconvenientes

Se trata de una política muy simple y sencilla de llevar a la práctica, pero muy pobre en cuanto a su comportamiento.

Prioridad al más corto.SJF (Shortest Job First)

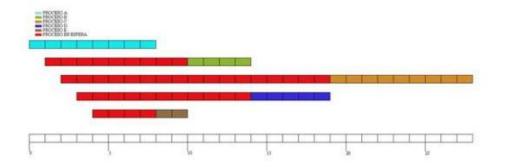
▶ Idea general del algoritmo

En este algoritmo, da bastante prioridad a los procesos más cortos a la hora de ejecución y los coloca en la cola.

Prioridad al más corto.SJF (Shortest Job First)

► Ejemplo de uso del procesador

Proceso	Tiempo de ejecución	Tiempo de llegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0	8	8	0
В	4	1	10	14	14 - 1= 13	13-4=9
С	9	2	19	28	28 -2 = 26	26 - 9 = 17
D	5	3	14	19	19 - 3 = 16	16-5=11
E	2	4	8	10	10 - 4= 6	6-2=4



Prioridad al más corto.SJF (Shortest Job First)

Ventajas

- Minimiza el tiempo de finalización promedio
- Da el mínimo tiempo de espera para un conjunto de procesos
- ▶ Entra en el CPU el proceso mas breve
- Tiempo de espera menor

Desventajas

- Riesgo de perdida de los ficheros grandes
- La dificultad de este sistema es conocer la longitud del próximo proceso que va a entrar en el CPU

Round Robin.

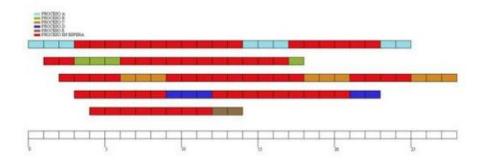
▶ Idea general del algoritmo

Es un método para seleccionar todos los elementos en un grupo de igual manera y en un orden en concreto, desde el primero elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento.

Round Robin.

► Ejemplo de uso del procesador

Proceso	Tiempo de ejecución	Tiempo de Ilegada	Tiempo de comienzo	Tiempo de finalización	Tiempo de retorno	Tiempo de espera
A	8	0	0-14 -23	3 -17 -25	25	25 -8 = 17
В	4	1	3 -17	6 -18	18 - 1= 17	17 -4 = 13
C	9	2	6-18-25	9 -21 -28	28 -2 = 26	26 - 9 = 17
D	5	3	9-21	12 -23	23 - 3 = 20	20 - 5 = 15
E	2	4	12	14	14 - 4= 4	10 -2 = 2



Round Robin

Ventajas

- Es un buen método para ficheros de distinto tamaño
- Dependiendo de los ficheros, puede llegar a ser mas eficientes que los demás.

Desventajas

Uno de los algoritmos de planificación de procesos más complejos y difíciles, dentro de un sistema operativo asigna a cada proceso una porción de tiempo equitativa y ordenada,

Planificación por prioridad

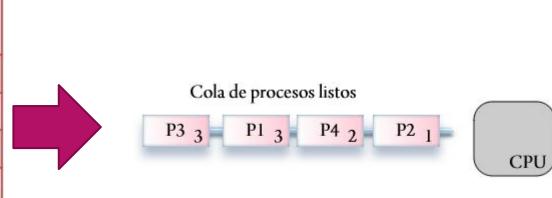
► Idea general del algoritmo

En este algoritmo a cada proceso se le asocia un número entero de prioridad. Mientras menor sea este entero pues mayor prioridad tiene el proceso.

Planificación por prioridad

► Ejemplo de uso del procesador

Proceso	Prioridad	Tiempo de ejecución (te)
P1	3	10
P2	1	1
Р3	3	2
P4	2	5



Planificación por prioridad

Ventajas

- Es una mejora del SFJ, por lo que las ventajas las hereda de esas
- Lo que mejora es el retraso excesivo de procesos largos y el favoritismo por procesos cortos

Desventajas

Posibilidad de bloqueo indefinido, porque los procesos con mas prioridad pueden necesitar la ayuda de unos con menos prioridad y no avanzar.

Planificación garantizada

► Idea general del algoritmo

Cosiste en hacer como un "pacto", si hay n usuarios en sesión mientras usted está trabajando, usted recibirá aproximadamente 1/n de la capacidad de la CPU.

Planificación garantizada

▶ Ejemplo de uso del procesador

Con los datos anteriores y el registro de procesos en curso de ejecución, el sistema calcula y determina qué procesos están más alejados por defecto de la relación "1 / n" prometida y prioriza los procesos que han recibido menos CPU de la prometida.

Planificación garantizada

Ventajas

- Si eres monousuario, puedes tener el rendimiento optimo
- Cada cola tiene su propio algoritmo de planificación

Desventajas

 A cuantos mas usuarios tengas, el sistema puede quedar afectado debido al rendimiento

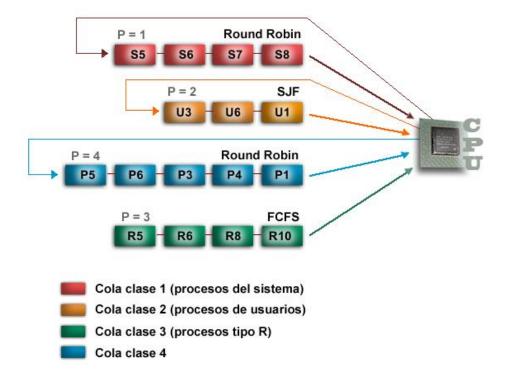
▶ Idea general del algoritmo

Son procesos que requieren una planificación de tiempo compartido adecuada, pero quizás haya que ejecutar también procesos de tiempo real, que no pueden estar sujetos a una expulsión por tiempo.

Para este algoritmo se requieren dos niveles de planificación:

- Planificación dentro de cada cola: Cada cola puede utilizar su propia política de planificación, de acuerdo a la clase de procesos que acoge, la cual puede ser usando diferentes algoritmos (FCFS, Round Robín, etc.).
- Planificación entre colas:
 - ▶ Se le asigna una prioridad (P) a cada cola.
 - ▶ Se le asigna un Quantum de CPU a cada cola, que se reparte entre los procesos de cada cola.

Ejemplo de uso del procesador



Ventajas

- Soporta bien la sobrecarga.
- Adaptable a las necesidades del sistema.
- Algoritmo más general.

Desventajas

- Un fallo en el maestro hace que falle el sistema completo
- ► El maestro puede llegar a ser un cuello de botella para el rendimiento del sistema