

# UNIVERSIDAD DE GUADALAJARA

## UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Tema: ( Algoritmos de planificación de procesos )



PRESENTA:

( Jared Isaías Monje Flores )

Docente:

( Javier Rosales Martínez )

Materia:

( Seminario de Solución de Sistemas Operativos )

## **Algoritmos de planificación de procesos**

Los algoritmos de planificación son fundamentales en la gestión de procesos de un sistema operativo. Estos determinan el orden en que las tareas se ejecutan en la CPU. Entre los enfoques destacados se encuentran Round Robin, que asigna tiempos de CPU equitativos a los procesos; SJF (Shortest Job First), que prioriza los trabajos más cortos; FIFO (First-In-First-Out), que sigue el orden de llegada; y los algoritmos basados en prioridades, que ejecutan primero las tareas de mayor importancia. Exploraremos cómo estos métodos influyen en la eficiencia y el rendimiento del sistema, optimizando la utilización de recursos y tiempos de respuesta.

### **Algoritmo Round Robin**

Round Robin es usado para ayudar a crear procesos o trabajos programados para asegurar que cada proceso requerido para hacer un trabajo, tiene bastante tiempo para hacerlo. En los ordenadores, la CPU puede hacer divisiones de tiempo para configurar un tiempo determinado para cada proceso a usar en cada ciclo. Usando Round Robin asigna una porción de tiempo para cada proceso que está en funcionamiento.

Por ejemplo, en un ordenador el usuario empieza tres aplicaciones, que pueden ser la aplicación de correo, el navegador Web y un procesador de texto. Estas aplicaciones son cargadas en el sistema de memoria, y cada uno de ellos puede funcionar de forma transparente para el usuario.

El uso de Round Robin ayuda al ordenador seguir las necesidades del usuario y manejar de forma efectiva los procesos de todas las aplicaciones.

Con este sistema se puede mantener todos los trabajos en progreso mediante el uso de ciclos de tiempo. Esta es la forma en que podemos hacer varias cosas al mismo tiempo en nuestro ordenador.

Es especialmente útil en sistemas de tiempo compartido porque asigna tiempo de CPU a cada proceso de manera justa y equitativa. Cada proceso recibe el mismo tiempo de CPU en función de su posición en la cola de procesos y la duración del quantum.

Permite una respuesta rápida a las solicitudes de los usuarios porque cada proceso

se ejecuta durante un intervalo de tiempo fijo. Esto significa que los procesos de corta duración pueden completarse rápidamente, y los procesos más largos pueden ejecutarse en varias iteraciones.

Tiene un overhead mínimo porque no necesita mantener información adicional sobre los procesos en ejecución. Sólo se necesita una cola de procesos y un temporizador para implementar el algoritmo.

### **SJF ( Shortest Job First ) - Primero el trabajo más corto**

Este algoritmo supone que los tiempos de ejecución se conocen de antemano. Cuando hay varios trabajos de igual importancia a ser iniciados en la cola de entrada, el planificador selecciona el trabajo mas corto primero SJF(Shortest Job First) y en caso de empate prioriza con el algoritmo primero en entrar primero en salir (FIFO, First In First Out). Algunas de las sus características son:

- Asociar a cada proceso el tiempo de ráfaga de CPU: una vez llega cada proceso se le asigna tiempo de ráfaga de CPU, que son las unidades de tiempo que requieren para que el proceso se ejecute completamente.
- Selecciona el proceso con menor ráfaga de CPU: antes de ejecutar cada proceso el algoritmo prioriza cual es el proceso mas corto a ejecutar.
- En caso de empate aplicar FIFO: si llega a presentarse empate en dos o mas procesos de rafaga de CPU, el algoritmo cambia a priorizar primero en entrar primero en salir, algoritmo FIFO.
- Algoritmo no expulsivo: una vez se empieza a ejecutar cada proceso no será interrumpido hasta finalizar.

### **FIFO ( El primero en llegar, el primero en salir )**

El algoritmo FIFO (First-In-First-Out), en el contexto de la planificación de procesos, es uno de los enfoques más simples. Funciona de la siguiente manera:

Cola de Listos: Mantén una cola de procesos listos para ejecutarse. Cuando un proceso llega y está listo para ejecutarse, se coloca al final de esta cola.

Selección del Proceso: Cuando se necesita seleccionar un proceso para ejecutarse,

el algoritmo FIFO selecciona el proceso que está en la parte frontal de la cola de listos. Es decir, el proceso que ha estado esperando más tiempo en la cola.

**Ejecución del Proceso:** El proceso seleccionado se ejecuta en la CPU hasta que se completa su ejecución o se bloquea. Si se bloquea, se mueve a la cola de bloqueados y se selecciona el siguiente proceso de la cola de listos.

**Liberación de la CPU:** Cuando un proceso termina su ejecución, se retira de la CPU y se elimina de la cola de listos.

El principio fundamental del algoritmo FIFO es que los procesos se ejecutan en el mismo orden en que llegaron, sin considerar sus necesidades de tiempo de CPU o prioridades. Este enfoque es fácil de entender e implementar, pero puede no ser eficiente en términos de tiempo de respuesta, ya que los procesos largos pueden bloquear procesos más cortos que están detrás en la cola. Por lo tanto, no es adecuado para sistemas donde se requiera una alta priorización de procesos o una respuesta rápida.

### **Algoritmo de planificación por prioridad**

En este algoritmo a cada proceso se le asigna un número entero de prioridad. Mientras menor sea este número pues mayor es la prioridad que tiene el proceso, por lo que la esencia del algoritmo es planificar la entrada de procesos de la CPU de acuerdo a la prioridad asociada de cada uno de ellos.

Consiste en el máximo aprovechamiento del sistema, lo que implica proveer un buen servicio a los procesos existentes en un momento dado. En otras palabras, busca equidad, o sea, que todos los procesos sean atendidos.

**Forma de planificación:** Apropiativa o Cooperativa, En la apropiativa expulsará de la CPU el proceso actual si la prioridad del proceso que acaba de llegar es mayor, mientras que en la cooperativa pondrá el nuevo proceso al principio de la cola de procesos preparados.

**Ventajas:**

- Algunas prioridades las pueden definir los usuarios.
- Se pueden ir incrementando gradualmente la prioridad de los procesos.

**Desventajas:**

- Los procesos de prioridad más baja pueden sufrir de inanición. Este problema ocurre, como era de esperar, si hay un flujo continuo de procesos listos de

alta prioridad.

En resumen, los algoritmos de planificación de procesos desempeñan un papel esencial en la gestión eficiente de recursos y tiempos en un sistema operativo. Hemos explorado varios enfoques clave, como Round Robin, SJF, FIFO y la planificación por prioridad.

Round Robin distribuye el tiempo de CPU de manera equitativa, ideal para sistemas de tiempo compartido. SJF prioriza la ejecución de trabajos más cortos para minimizar los tiempos de espera. FIFO, el enfoque más simple, sigue el orden de llegada de los procesos, aunque puede no ser óptimo en términos de tiempo de respuesta. La planificación por prioridad asigna prioridades a los procesos y se adapta a las necesidades específicas, pero puede llevar a problemas de inanición para procesos de baja prioridad.

La elección del algoritmo depende de los requisitos y objetivos del sistema operativo y su entorno de uso.

## Referencias

6 Algoritmos de Planificación I. (2023). Ehu.eus.  
<https://lsi.vc.ehu.es/pablogn/docencia/manuales/SO/TemasSOuJaen/PLANIFICACIONDEPROCESOS/6AlgoritmosdePlanificacionI.htm>

Beckerola. (2020, June 2). *¿Qué es el Algoritmo Round Robin?* Ordenadores Y Portátiles.  
<https://ordenadores-y-portatiles.com/round-robin/>

Lozano, D. (2018, March 26). *SJF(Shortest Job First) - Algoritmo Primero el trabajo mas corto*. Diloentutos Pc. <https://www.diloentutospc.com/sjfshortest-job-first-algoritmo-primero-el-trabajo-mas-corto/>

*Algoritmo de Planificación FIFO*. (2018). Prezi.com.  
<https://prezi.com/hqa5h-lyr0dn/algoritmo-de-planificacion-fifo/>

*Algoritmo de Planificación por Prioridad*. (2018). Prezi.com.  
<https://prezi.com/rvshbfizzxw/algoritmo-de-planificacion-por-prioridad/>