

Completely Fair Scheduler

El *Completely Fair Scheduler* (CFS) es el planificador de procesos predeterminado en el kernel de Linux a partir de la versión 2.6.23. Su objetivo es distribuir el tiempo de CPU entre los procesos de manera más justa que los planificadores tradicionales, especialmente cuando se ejecutan múltiples procesos de baja prioridad o cuando el sistema está bajo carga.

Características:

1. Objetivo de justicia en el reparto de CPU:

El objetivo principal de CFS es asegurar que todos los procesos reciban su "porción justa" de tiempo de CPU.

2. Estructura basada en árbol rojo-negro:

CFS utiliza un árbol rojo-negro (*red-black tree*) para almacenar y ordenar los procesos listos para ejecutarse. Cada proceso se coloca en el árbol según el tiempo total que ha usado la CPU.

3. Tiempo virtual:

El tiempo virtual es una métrica clave en CFS. Se utiliza para representar el tiempo total de CPU que cada proceso ha consumido. Los procesos con menos tiempo virtual tienen prioridad sobre aquellos que han usado más tiempo de CPU.

4. Latencia mínima:

CFS asegura que todos los procesos se ejecuten en un período de tiempo conocido como "latencia mínima".

5. Cálculo de tiempos de ejecución:

Aunque el CFS no se basa en "quantums" de tiempo de manera estricta, sigue utilizando "slices" o fragmentos de tiempo para determinar cuánto tiempo puede ejecutarse un proceso antes de ser considerado para ser interrumpido. Estos "slices" están influenciados por la cantidad de procesos en el sistema y sus respectivas prioridades.

- A más procesos ejecutándose, el *slice* disponible para cada proceso se reduce.
- Los procesos con mayor prioridad (menor valor de *niceness*) reciben más tiempo de CPU antes de ser interrumpidos en comparación con los procesos de menor prioridad.

6. Latencia de programación:

El CFS introduce el concepto de latencia de programación (*scheduling latency*), que es el tiempo máximo que un proceso puede estar esperando antes de ser ejecutado. Este valor depende del número de procesos activos

7. Control de grupos (cgroups):

CFS también soporta el uso de *cgroups* para agrupar procesos y asignarles recursos de CPU de manera equitativa entre los grupos. Esto es útil en entornos multiusuario o de contenedores, donde diferentes grupos de procesos pueden requerir distintas cuotas de CPU.

8. Ventajas sobre planificadores anteriores:

En comparación con el planificador $O(1)$, que simplemente distribuye *quantums* de CPU a cada proceso sin mucha equidad, el CFS asegura una distribución más justa del tiempo de CPU