

Nombre: Steven Facundo Mejía Xolop
Carnet: 202104160
Curso: Sistemas Operativos
Sección: “A”

Actividad 7

características principales y el funcionamiento del Completely Fair Scheduler (CFS) de Linux

El Completely Fair Scheduler (CFS) es un algoritmo de planificación de procesos que se introdujo en el kernel de Linux a partir de la versión 2.6.23. Este algoritmo revolucionó la forma en que Linux administra la CPU, priorizando la equidad en la distribución del tiempo de CPU entre los procesos en ejecución. A diferencia de los algoritmos de planificación anteriores, que a menudo estaban basados en prioridades estáticas, el CFS opera en un principio de tiempo compartido justo, donde todos los procesos tienen la oportunidad de utilizar la CPU de manera equitativa.

El CFS fue diseñado para abordar varios problemas presentes en los algoritmos de planificación anteriores, como el algoritmo $O(1)$ utilizado en versiones anteriores del kernel de Linux. Este último, aunque eficiente, a menudo privilegiaba a los procesos con mayor prioridad y no garantizaba una distribución justa del tiempo de CPU, especialmente en sistemas con cargas de trabajo mixtas.

Las características principales del CFS incluyen:

- **Justicia y Equidad:** El CFS se esfuerza por proporcionar una asignación justa y equitativa del tiempo de CPU entre todos los procesos en ejecución. Esto significa que ningún proceso debe ser favorecido sobre otro, independientemente de su prioridad o de la cantidad de procesos en el sistema. Esta característica es esencial para garantizar un rendimiento uniforme y predecible del sistema.
- **Basado en el Tiempo:** A diferencia de los algoritmos de planificación basados en prioridades estáticas, el CFS opera en un principio de tiempo compartido justo. Utiliza una métrica de tiempo real transcurrido para determinar la cantidad de tiempo de CPU que cada proceso ha consumido. Esto permite que el CFS proporcione una asignación proporcional de la CPU en función del tiempo de espera de cada proceso.
- **Prioridades Dinámicas:** Aunque el CFS no utiliza prioridades estáticas para la planificación, asigna pesos a los procesos. Estos pesos determinan la proporción de tiempo de CPU que un proceso debe recibir en comparación con otros. Cuanto mayor sea el peso asignado a un proceso, mayor será su parte del tiempo de CPU. Este enfoque dinámico permite que el CFS adapte la asignación de CPU según las necesidades cambiantes del sistema.

- **Núcleo Interactivo:** El CFS está diseñado para manejar eficientemente cargas de trabajo interactivas, como las aplicaciones de escritorio y los servidores web. Prioriza la ejecución de procesos interactivos para garantizar una respuesta rápida del sistema, incluso en condiciones de carga pesada. Esto se logra al proporcionar a los procesos interactivos una asignación justa de la CPU, independientemente de la presencia de procesos intensivos en CPU.
- **Sin Bloqueo:** A diferencia de algunos algoritmos de planificación anteriores, el CFS no bloquea los procesos con alta prioridad. Todos los procesos tienen la oportunidad de ejecutarse, aunque algunos pueden tener más tiempo de CPU que otros según sus pesos asignados. Esto garantiza que ningún proceso se vea privado de la CPU, lo que podría provocar bloqueos o cuellos de botella en el sistema.
- **Algoritmo de Selección:** El CFS selecciona el próximo proceso a ejecutar eligiendo aquel con la menor relación de "tiempo de espera virtual". El tiempo de espera virtual es una estimación del tiempo que un proceso ha esperado para ejecutarse en comparación con otros procesos en el sistema. Al seleccionar el proceso con el menor tiempo de espera virtual, el CFS garantiza que los procesos más "hambrientos" de CPU tengan prioridad en la ejecución.

El funcionamiento del CFS se basa en una estructura de datos de árbol rojo-negro para mantener un seguimiento eficiente de los procesos en ejecución y sus tiempos de espera virtuales. Cada vez que se produce un cambio en el estado del sistema, como la finalización de un proceso o la llegada de un nuevo proceso, el CFS reequilibra el árbol rojo-negro para garantizar una distribución justa del tiempo de CPU. Esto garantiza un rendimiento equitativo del sistema incluso en condiciones de carga variable y cargas de trabajo mixtas.