Memoria de P6

Gestión de Procesos

Shunya Zhan

05 de mayo del 2025

ÍNDICE

Introducción

En el ámbito de la programación de sistemas, la gestión de la planificación de procesos es crucial para el rendimiento y la eficiencia del sistema operativo. Este programa en C está diseñado para interactuar con las políticas de planificación del sistema operativo, proporcionando información detallada sobre las políticas y prioridades del proceso en curso. Además, permite realizar cambios en estas políticas y prioridades, asegurando que los cambios se apliquen correctamente.

El programa se centra en tres políticas de planificación principales: SCHED_OTHER, SCHED_FIFO y SCHED_RR. Cada una de estas políticas tiene características específicas que afectan cómo se gestionan los procesos:

- SCHED_OTHER: Es la política de planificación por defecto en muchos sistemas operativos basados en Linux. Utiliza un algoritmo de planificación de tiempo compartido.
- SCHED_FIFO: Es una política de planificación en tiempo real que sigue un orden de primer en entrar, primero en salir (FIFO). Los procesos con esta política tienen prioridad sobre los procesos con SCHED_OTHER.
- SCHED_RR: Similar a SCHED_FIFO, pero con un quantum de tiempo asignado a cada proceso, después del cual el proceso es interrumpido y se le da la oportunidad a otro proceso de ejecutarse.

El objetivo de este programa es proporcionar una herramienta que no solo muestre los valores actuales de estas políticas, sino que también perm ita modificar la política y prioridad del proceso actual. Esto es especialmente útil para desarrolladores y administradores de sistemas que necesitan optimizar el rendimiento de sus aplicaciones y sistemas.

Desarrollo

El desarrollo de este programa se basa en el uso de las funciones proporcionadas por la biblioteca <sched.h> para interactuar con las políticas de planificación del sistema operativo. A continuación, se detallan las principales tareas que realiza el programa:

- Mostrar valores de las políticas de planificación: El programa imprime los valores de las políticas SCHED_OTHER, SCHED_FIFO y SCHED_RR. Esto proporciona una visión general de las políticas disponibles y sus características.
- Obtener y mostrar la política y prioridad actuales: Utilizando las funciones sched_getscheduler y sched_getparam, el programa obtiene y muestra la política de planificación y la prioridad actuales del proceso en ejecución. Esto es útil para entender el estado actual del proceso.
- Mostrar valores de prioridad mínimo y máximo: Las funciones sched_get_priority_min y sched_get_priority_max se utilizan para obtener y mostrar los valores de prioridad mínimo y máximo para cada política de planificación. Esto ayuda a comprender el rango de prioridades que se pueden asignar a los procesos.
- Mostrar el quantum de tiempo para SCHED_RR: Para la política SCHED_RR, el programa utiliza la función sched_rr_get_interval para obtener y mostrar el quantum de tiempo asignado a los procesos. Este valor es importante para entender cómo se gestiona el tiempo de CPU entre los procesos en esta política.
- Cambiar la política y prioridad del proceso: El programa cambia la política de planificación y la prioridad del proceso actual. Primero, cambia entre SCHED_OTHER y SCHED_FIFO, y luego asigna SCHED_RR con la prioridad máxima. Esto se realiza utilizando las funciones sched_setscheduler y sched_setparam.
- Verificar los cambios realizados: Después de realizar los cambios en la política y prioridad, el programa verifica que los cambios se hayan aplicado correctamente. Esto asegura que las modificaciones se han realizado de manera efectiva y que el proceso se está ejecutando con la nueva configuración.

Memoria

Inicialmente, se intentó ejecutar el programa en un contenedor Docker. Los pasos seguidos fueron:

- Entrar en modo root en el contenedor y ejecutar el programa.
- Agregar al usuario del contenedor al grupo sudo.
- Reiniciar el contenedor.

Sin embargo, se encontraron problemas relacionados con los permisos y la configuración del contenedor, lo que impidió cambiar las políticas de planificación. La solución fue utilizar una máquina virtual (VM) y entrar en modo superusuario (sudo su). Esto resolvió los problemas de permisos y permitió ejecutar el programa correctamente.

Compilación y ejecución

Compilación: Abre una terminal en la VM (máquina virtual) e introducir en terminal siguiente comando para compilar el código:

- gcc -o p6 p6.c

Ejecución: En la terminal, ejecuta p6 con el siguiente comando:

- ./p6

Nota: si sale error de permiso, entonces entra en modo de superusuario con sudo

- sudo ./p6

Prueba y resultado

En mi ordenador ha habido problema de VM, y para una práctica he prestado favor al compañero que me ejecute el código, y que me pasara captura de pantalla de la salida.

Conclusión

El programa desarrollado cumple con los objetivos planteados, proporcionando una herramienta eficaz para obtener y modificar información sobre las políticas de planificación y prioridades del proceso en curso. A través del uso de la biblioteca <sched.h>, el programa es capaz de:

- Mostrar los valores de las políticas de planificación SCHED_OTHER, SCHED_FIFO y SCHED_RR.
- Obtener y mostrar la política y prioridad actuales del proceso.
- Mostrar los valores de prioridad mínimo y máximo para cada política.
- Mostrar el quantum de tiempo para la política SCHED_RR.
- Cambiar la política y prioridad del proceso, verificando que los cambios se apliquen correctamente.

Durante el desarrollo, se enfrentaron problemas relacionados con los permisos y la configuración al intentar ejecutar el programa en un contenedor Docker. La solución fue utilizar una máquina virtual con permisos de superusuario, lo que permitió ejecutar el programa sin inconvenientes.

Este proyecto destaca la importancia de comprender las limitaciones del entorno de ejecución y adaptarse a ellas para lograr los objetivos del proyecto. Además, proporciona una base sólida para futuras mejoras y extensiones, como la inclusión de más políticas de planificación o la optimización del manejo de errores.

En resumen, el programa no solo cumple con los requisitos técnicos, sino que también ofrece una valiosa lección sobre la adaptabilidad y la resolución de problemas en el desarrollo de software.

Anexo

- Código fuente
- Campus Virtual
- Copilot