



Tecnológico de Monterrey

Reflexión - Actividad de programación 4
Diseño de sistemas embebidos avanzados

Alumnos:

Ricardo Sierra Roa A01709887

Profesor:

Pedro Oscar Pérez Murueta

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Querétaro

Fecha de entrega:

11 de noviembre de 2024

Uso de Chat GPT

ChatGPT fue un recurso esencial para abordar y resolver las dudas que surgieron durante la implementación del problema del barbero, un reto de programación concurrente. Desde el inicio, ChatGPT proporcionó orientación clara sobre cómo utilizar `mutex` y `condition_variable` para gestionar la sincronización entre los hilos del barbero y los clientes, garantizando un acceso seguro a los recursos compartidos y previniendo condiciones de carrera. A lo largo del desarrollo, ChatGPT ayudó a identificar problemas en la lógica de los contadores que registraban cuántas veces cada cliente era atendido y cuántas veces encontraba la barbería llena. Sugerí soluciones detalladas para ajustar y verificar que los contadores (`attendCount` y `fullShopCount`) se actualizaran adecuadamente en base a los eventos del programa. Esto implicó la modificación de la estructura de los bucles y la lógica de control para asegurar que el registro de intentos y la impresión de resultados fueran precisos y reflejaran la realidad de la ejecución del programa.

Cuando enfrenté dificultades para detener el programa de manera correcta tras alcanzar el límite de intentos por cliente, ChatGPT proporcionó un enfoque claro para implementar una verificación de condición de paro que asegurara que todos los clientes cumplieran con el requisito de intentos antes de finalizar. Este ajuste fue crucial para evitar que algunos clientes superaran el número de intentos permitidos o que el programa continuara ejecutándose innecesariamente. Además de brindar soluciones, ChatGPT ofreció ejemplos prácticos y explicaciones sencillas que facilitaron la comprensión de conceptos complejos de la programación multihilo, como la coordinación de hilos y la gestión de recursos compartidos sin generar bloqueos. Esto ayudó a mejorar la eficiencia en la corrección de errores y a aplicar buenas prácticas de programación concurrente. Gracias a la ayuda de ChatGPT, el código se refinó para ser más preciso y funcional, cumpliendo con los requisitos establecidos de forma robusta. Las recomendaciones y análisis proporcionados permitieron un proceso de desarrollo más eficiente y resultaron en un código que gestionaba correctamente la sincronización de hilos y el seguimiento de eventos.

Reflexión

Durante la actividad del problema del barbero, ChatGPT fue una herramienta esencial para superar los desafíos de programación concurrente. Me proporcionó una guía detallada y estructurada para la implementación del código, ayudándome a gestionar la sincronización de hilos con mutex y `condition_variable`, y a depurar la lógica para garantizar que cada cliente realizara exactamente 5 intentos de entrada, ya fuera siendo atendido o encontrando la barbería llena. La capacidad de ChatGPT de simular escenarios y sugerir ajustes precisos permitió identificar y corregir problemas de concurrencia y condiciones de carrera, lo que fue crucial para el desarrollo exitoso de la actividad.

Las explicaciones claras y los ejemplos prácticos proporcionados por ChatGPT facilitaron la comprensión de conceptos complejos, como el manejo de recursos compartidos y la prevención de bloqueos. Esto no solo mejoró la eficiencia durante el proceso de desarrollo, sino que también enriqueció el aprendizaje, permitiéndome adquirir una mejor comprensión de las técnicas de sincronización en programación de multihilos. La asistencia de ChatGPT en la identificación de errores comunes y en la sugerencia de soluciones efectivas fue una parte integral para lograr terminar la actividad con éxito.

Sin embargo, para futuras actividades, sería valioso contar con más herramientas visuales o diagramas de flujo que complementen las explicaciones textuales, ayudando a visualizar la interacción entre los hilos y los recursos. Además, incluir pruebas de estrés automatizadas permitiría evaluar la robustez del código en situaciones de alta concurrencia y prevenir posibles fallos bajo condiciones extremas. A pesar de estas áreas de mejora, la experiencia con ChatGPT fue muy positiva, facilitando la solución de problemas complejos y mejorando la confianza en el desarrollo de software concurrente.