



UNIVERSIDAD AUTONOMA DE CHIAPAS

FACULTAD DE CONTADURIA Y

ADMINISTRACION CAMPUS 1

MATERIA;

TEORIA MATEMATICA DE LA COMPUTACION

DOCENTE:

HECTOR GUERRA CRESPO

TAREA:

TAREA 06

ALUMNO:

CARLOS ANTONIO AGUILAR RAMOS

GRADO Y

GRUPO:3J

LUGAR Y FECHA:

TUXTLA GUTIERREZ CHIAPAS, 03 DE ABRIL DE
2023

Introducción

En programación, el costo de ejecución se refiere a la cantidad de recursos que utiliza un programa mientras se está ejecutando. Estos recursos incluyen la memoria, los registros y las pilas, y su uso puede afectar significativamente el rendimiento y la eficiencia de un programa. La memoria se utiliza para almacenar datos y código, y un programa que utiliza demasiada memoria puede hacer que el sistema se bloquee o se ralentice. Los registros son áreas de almacenamiento en la CPU utilizadas para almacenar datos temporales durante operaciones, y el uso excesivo de registros puede disminuir el rendimiento del programa. La pila es una estructura de datos temporal utilizada para almacenar datos mientras se realiza una función, y su uso excesivo puede provocar desbordamientos y errores.

Para minimizar los costos de ejecución y mejorar la eficiencia del programa, se pueden utilizar técnicas de optimización de memoria, registro y pila. Estas técnicas pueden incluir el uso de estructuras de datos eficientes, la liberación de memoria no utilizada, la reducción del tamaño de la pila y el uso de registros de manera más efectiva.

Costos - costo de ejecución. (memoria, registros, pilas)

En programación, el costo de ejecución se refiere a la cantidad de recursos que utiliza un programa mientras se está ejecutando. Estos recursos pueden incluir memoria, registros y pilas.

La memoria se utiliza para almacenar datos y código. Un programa que utiliza mucha memoria puede provocar que el sistema se quede sin memoria y se bloquee o se ralentice. Para minimizar el costo de la memoria, se pueden utilizar técnicas de optimización de memoria, como el uso de estructuras de datos eficientes y la liberación de memoria cuando ya no se necesita.

Los registros son pequeñas áreas de almacenamiento en la unidad central de procesamiento (CPU) que se utilizan para almacenar datos temporales mientras se están realizando operaciones. El costo de registro se refiere a la cantidad de registros que utiliza un programa mientras se está ejecutando. Al utilizar menos registros, se puede reducir el costo de registro y mejorar la eficiencia del programa.

La pila es una estructura de datos utilizada para almacenar datos temporalmente mientras se realiza una función. El costo de la pila se refiere a la cantidad de espacio que se utiliza en la pila durante la ejecución del programa. Un programa que utiliza demasiada pila puede provocar un desbordamiento de la pila y causar un error. Para minimizar el costo de la pila, se pueden utilizar técnicas de optimización de la pila, como la reducción del tamaño de la pila o el uso de estructuras de datos alternativas.

En general, minimizar los costos de ejecución es importante para mejorar la eficiencia y el rendimiento de un programa. Se pueden utilizar diversas técnicas de optimización para reducir los costos de memoria, registro y pila.

Costos - criterios para mejorar el código

Hay varios criterios que se pueden utilizar para mejorar el código y reducir el costo de ejecución de un programa. Algunos de los criterios más importantes son los siguientes:

Eficiencia: El código debe estar diseñado para realizar las operaciones de la manera más eficiente posible. Se deben evitar las operaciones redundantes y las estructuras de datos ineficientes.

Legibilidad: El código debe ser fácil de leer y entender para que sea más fácil de mantener y modificar en el futuro. Se pueden utilizar comentarios y una estructura de código clara para mejorar la legibilidad.

Modularidad: El código debe estar dividido en módulos independientes que realicen tareas específicas. Esto permite una mayor reutilización de código y facilita la corrección de errores.

Robustez: El código debe estar diseñado para manejar errores y excepciones de manera adecuada. Esto significa que debe ser resistente a entradas incorrectas y asegurarse de que no se produzcan errores inesperados.

Escalabilidad: El código debe estar diseñado para poder manejar grandes volúmenes de datos y escalarse de manera efectiva para futuras necesidades.

Mantenibilidad: El código debe ser fácil de mantener y actualizar a medida que surjan nuevas necesidades y se realicen cambios en los requisitos.

Al seguir estos criterios, se puede mejorar significativamente el código y reducir el costo de ejecución del programa. Al mismo tiempo, se puede garantizar que el programa sea robusto, escalable y fácil de mantener en el futuro.

Costos - herramientas para el análisis del flujo de datos

Existen varias herramientas para el análisis del flujo de datos en un programa. Algunas de las herramientas más utilizadas son:

Analizadores estáticos: Estas herramientas analizan el código fuente del programa para identificar posibles problemas de flujo de datos. Estos problemas pueden incluir variables no utilizadas, variables no inicializadas, bucles infinitos y condiciones siempre verdaderas o siempre falsas.

Ejemplos de analizadores estáticos incluyen Pylint, SonarQube, y Coverity.

Analizadores dinámicos: Estas herramientas analizan el flujo de datos mientras se ejecuta el programa para identificar posibles problemas. Esto puede incluir la identificación de fugas de memoria, accesos a memoria no válidos y uso excesivo de recursos. Ejemplos de analizadores dinámicos incluyen Valgrind y Memcheck.

Herramientas de profiling: Estas herramientas registran el comportamiento del programa durante la ejecución para identificar cuellos de botella y puntos críticos. Esto puede incluir la identificación de funciones que consumen una cantidad excesiva de recursos o que se ejecutan con frecuencia. Ejemplos de herramientas de profiling incluyen Perf, Gprof y Visual Studio Profiler.

Inspección manual: Esta es una técnica de análisis que implica la revisión manual del código fuente para identificar posibles problemas de flujo de datos. Esta técnica puede ser lenta y propensa a errores humanos, pero puede ser útil en casos en los que otras herramientas no son efectivas.

