

3.2 Costos

Los costos son el factor más importante a tomar en cuenta a la hora de optimizar ya que en ocasiones la mejora obtenida puede verse no reflejada en el programa final, pero si ser perjudicial para el equipo de desarrollo. La optimización de una pequeña mejora tal vez tenga una pequeña ganancia en tiempo o en espacio, pero sale muy costosa en tiempo en generarla.

Pero en cambio si esa optimización se hace por ejemplo en un ciclo, la mejora obtenida puede ser N veces mayor por lo cual el costo se minimiza y es benéfico la mejora.

Por ejemplo:

```
for(int i=0; i < 10000; i++);
```

si la ganancia es de 30 ms 300s

3.2.1 Costo de ejecución (Memoria, registros, pilas)

Los costos de ejecución son aquellos que vienen implícitos al ejecutar el programa. En algunos programas se tiene un mínimo para ejecutar el programa, por lo que el espacio y la velocidad de los microprocesadores son elementos que se deben optimizar para tener un mercado potencial más amplio.

Las aplicaciones multimedia como los videojuegos tienen un costo de ejecución alto por lo cual la optimización de su desempeño es crítica, la gran mayoría de las veces requieren de procesadores rápidos (e.g. tarjetas de video) o de mucha memoria. Otro tipo de aplicaciones que deben optimizarse son las aplicaciones para dispositivos móviles.

Los dispositivos móviles tienen recursos más limitados que un dispositivo de cómputo convencional razón por la cual, el mejor uso de memoria y otros recursos de hardware tiene mayor rendimiento. En algunos casos es preferible tener la lógica del negocio más fuerte en otros dispositivos y hacer uso de arquitecturas descentralizadas como cliente/servidor o P2P.

Organización de la memoria en tiempo de ejecución

Las técnicas de gestión de la memoria durante la ejecución del programa difieren de unos lenguajes de programación a otros, e incluso de unos compiladores a otros.

Equipo Amarillo

Memoria

La memoria es uno de los recursos más importantes de la computadora y, en consecuencia, la parte del sistema operativo responsable de tratar con este recurso, el gestor de memoria, es un componente básico del mismo. El gestor de memoria del sistema operativo debe hacer de puente entre los requisitos de las aplicaciones y los mecanismos que proporciona el hardware de gestión de memoria.

Para que una variable pueda ser almacenada en memoria estática, es necesario conocer su tamaño (número de bytes necesarios para su almacenamiento) en tiempo de compilación. Como consecuencia, aunque una variable (u objeto) sea de ámbito global, no podrán ocupar almacenamiento estático: " Los objetos que correspondan a procedimientos o funciones recursivas, ya que en tiempo de compilación no se conoce el número de variables que serán necesarias. "

Registros

Los registros del procesador se emplean para controlar instrucciones en ejecución, manejar direccionamiento de memoria y proporcionar capacidad aritmética. Los registros son espacios físicos dentro del microprocesador con capacidad de 4 bits hasta 64 bits dependiendo del microprocesador que se emplee. Los registros son direccionables por medio de una viñeta, que es una dirección de memoria. Los bits, por conveniencia, se numeran de derecha a izquierda (15,14,13.... 3,2,1,0), los registros están divididos en seis grupos los cuales tienen un fin específico.

Tipos De Registros

- Registros de Segmento
- Registros de Apuntador de Instrucciones
- Registros Apuntadores
- Registros de Propósito General
- Registros Índices
- Registros de Banderas

Equipo Amarillo

Pilas

Algunos lenguajes de programación utilizan la pila para almacenar datos que son locales a espacio para los datos locales se asigna a los temas de la pila cuando el procedimiento se introduce, y son borradas cuando el procedimiento termina.

3.2.2 Criterios para mejorar el código

La mejor manera de optimizar el código es hacer ver a los programadores que optimicen su código desde el inicio, el problema radica en que el costo podría ser muy grande ya que tendría que codificar más y/o hacer su código más legible. Los criterios de optimización siempre están definidos por el compilador.

Muchos de estos criterios pueden modificarse con directivas del compilador desde el código o de manera externa. Este proceso lo realizan algunas herramientas del sistema como los ofuscadores para código móvil y código para dispositivos móviles.

3.2.3 Herramientas para el análisis del flujo de datos

Existen algunas herramientas que permiten el análisis de los flujos de datos, entre ellas tenemos los depuradores y desensambladores. La optimización al igual que la programación es un arte y no se ha podido sistematizar del todo.

Depurador

Un depurador (en inglés: debugger) es un programa usado para probar y depurar (eliminar) los errores de otros programas (el programa "objetivo"). El código a ser examinado puede alternativamente estar corriendo en un simulador de conjunto de instrucciones (ISS), una técnica que permite gran potencia en su capacidad de detenerse cuando son encontradas condiciones específicas pero será típicamente algo más lento que ejecutando el código directamente en el apropiado (o el mismo) procesador. Algunos depuradores ofrecen dos modos de operación - la simulación parcial o completa, para limitar este impacto.

Desensamblador

Un desensamblador o disassembler es un programa de computador que traduce el lenguaje de máquina a lenguaje ensamblador, la operación inversa de la que hace el ensamblador. Un desensamblador difiere de un descompilador, en que éste tiene como objetivo un lenguaje de alto nivel en vez de al lenguaje ensamblador. La salida de un desensamblador, el desensamblado, es a menudo formateada para la legibilidad humana en vez de ser adecuada para la entrada a un ensamblador, haciendo que éste sea principalmente una herramienta de ingeniería inversa.

Equipo Amarillo

BIBLIOGRAFIA

López Figueroa, A. B. (2019). Lenguajes y Autómatas 2. Recuperado 16 de mayo de 2021, de Blogger. website: <https://alexisbladimirlopezfigueroa1998.blogspot.com/2019/10/32-costos.html>

Vargas, M. (2020). 3.2 Costos Lenguajes y Autómatas II. Recuperado 16 de mayo de 2021, de Prezi website: <https://prezi.com/p/lvwwwbwze8pu/32-costos-lenguajes-y-automatas-ii/>

Cortez Rosendo, C. R. (2020). 3.2 Costos Lenguajes y Autómatas II. Recuperado 16 de mayo de 2021, de Prezi website: <https://prezi.com/p/g-ds2g06nq9b/32-costos-lenguajes-y-automatas-ii/>

Equipo Amarillo

La información investigada aquí es por parte de todo el equipo sin menospreciar el trabajo de alguno al igual que en el desarrollo de la pagina