INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Ingeniería en sistemas computacionales

Lenguaje y Autómatas 2

Proyecto 3 Unidad 3 Optimización

Catedrático

Juan Pablo Rosas Baldazo

Alumno

Rafael Salazar Rodríguez 16/04/2018

Introducción

La optimización no es más que una mejora en el rendimiento en este caso de un programa, en este caso puede representar el tiempo de ejecución, el espacio, el uso del procesador que requiere entre otras.

El compilador determina las optimizaciones que pueden o no realizarse, además está estrechamente relacionada con el lenguaje de programación.

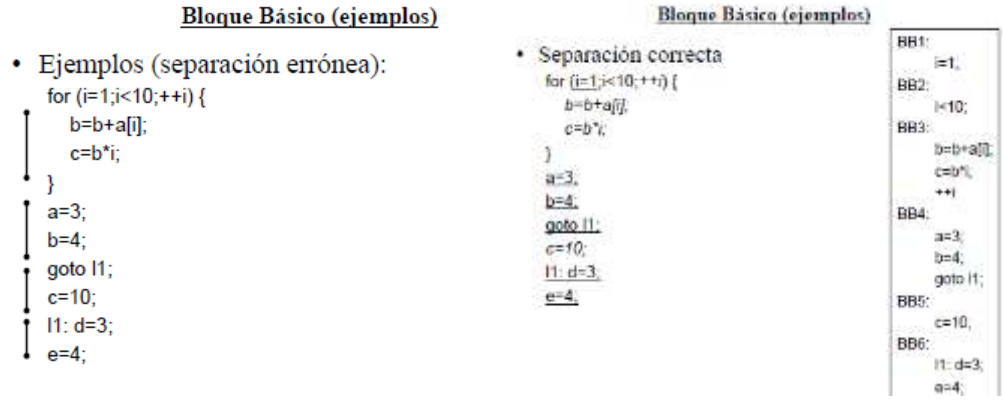
A continuación hablaremos de este tema y los diversos tipos y costos que conlleva desarrollar un software optimizado.

Capítulo 1. Tipos de optimización

La optimización puede dividirse en local, ciclo, global y mirilla hablaremos de cada una de ellas para conocerlas a detalle.

- **Sección 1.1: Locales**

Es realizada en los módulos del programa con funciones, métodos, procedimientos y clases, estos se ven reflejados en dichas secciones, un módulo es básicamente un fragmento de código con una única entrada, salida y que sus instrucciones se ejecutan simultáneamente.



Ejemplos: Folding, propagación de constantes, reducción de potencia y reducción de subexpresiones comunes.

- **Sección 1.2: Ciclos**

Un ciclo es una cierta parte del código que se repite hasta que se cumple con un cierto valor, el repetir una y otra vez las acciones de este puede conllevar muchos errores si tiene errores el problema pude crecer exponencialmente y más si existe código dentro del ciclo que no debe repetirse.

Sea el ejemplo:

```
while(a == b) {
 int c = a;
 c = 5;
 …;
}
```

En este caso es mejor pasar el Int c =a; fuera del ciclo de ser posible.

El principal conflicto de los ciclos es que es muy difícil determinar las funciones del código, así que no podemos optimizar todo el código. A pesar de ello este tipo de optimización permite mejorar las consultas en SQL y en aplicaciones remotas.

- **Sección 1.3: Globales**

Se da en todo el código, lo que la lleva a ser más lenta, pero mejora el desempeño de todo el código, depende más de la arquitectura de la máquina. Las variables globales agilizan los procesos aunque consuman más memoria. Destacan utilizar variables registros del CPU y utilizar instrucciones en ensamblador.

- **Sección 1.4: Mirilla**

Estructura el flujo del programa, principalmente centrándose en las instrucciones de bifurcación como decisiones, ciclos y saltos de rutinas.

Considera tener los saltos más cerca de las llamadas, haciendo el salto lo más pequeño posible. Un ejemplo de instrucción de bifurcación es los Break evitando que se ejecute cierta parte del código y salte a otra.

```
Switch (expresión que estamos evaluando)
{
Case 1: cout << “Hola” ;
Break;
Case 2: cout << “amigos”;
Break;
}
```

Capítulo 2. Costos

El costo es un gran factor que determina la optimización de un programa ya que estos se reflejan más en el equipo de desarrollo que en el programa en sí. Por lo cual una optimización que reduzca el espacio pueda salir muy costosa en tiempo, en el caso de los

ciclos una mejora siempre será un beneficio pues debido a la naturaleza de este la mejora crecerá exponencialmente.

Por ejemplo:

for(int i=0; i < 10000; i++); si la ganancia es de 30 ms 300s

- **Sección 2.1: Costo de ejecución (memoria, registros, pilas)**

Son aquellos que conlleva ejecutar el programa, muchos de estos tienen un mínimo costo debemos considerar el espacio y la velocidad del microprocesador que se deben optimizar para poder ejecutar programas más modernos y complejos.

En el caso de los videojuegos estos requieren un gran costo de ejecución lo que ocasiona que su optimización de desempeño sea crítica por esta razón se requieren procesadores rápidos, tarjetas de video y mucha memoria. Actualmente las aplicaciones móviles para Smartphone también deben optimizarse pues a diferencia del caso de las pc su pequeño tamaño complica la ejecución de los programas debido a que no cuentan con procesadores rápidos o mucha memoria.

- **Sección 2.2: Criterios para mejorar el código**

Para facilitar la optimización es preferible decirles a los programadores que lo hagan desde el inicio, porque si lo realizan después conlleva un mayor gasto de tiempo del proyecto lo que llevara a pagarles más pues podrían encontrar cada día después una manera mejor de optimización.

Los criterios están definidos por el compilador estos pueden modificarse con directivas desde el código mismo o de manera externa existen herramientas que realizan este proceso como los ofuscadores para código móvil y el código para este.

- **Sección 2.3: Herramientas para el análisis de flujo de datos**

Existen 2 herramientas principales para el análisis de flujo de datos los depuradores y desensambladores, el proceso de optimización es un arte y no se ha podido sistematizar del todo.

Conclusiones

La optimización es un proceso vital para el desarrollo de software sino todos los programas y aplicaciones que conocemos no funcionarían con los dispositivos con los que contamos, además de que los usuarios se molestarían si estos tardaran mucho en cargar o se ejecuta de mala forma, en el caso de los videojuegos esto se soluciona incluyendo requisitos mínimos para poder jugar y así conocer si nuestro equipo puede ejecutar el juego de forma óptima en el desarrollo de software para empresas es necesario recabar los requisitos funcionales y hacer el sistema de acuerdo a lo que se necesita y que puedan ejecutarlo ya sea en sus pc's o dispositivos móviles. Por lo cual empezar desde cero un

software conociendo las limitaciones que tendrá como base será la mejor opción y facilitara el trabajo a los desarrolladores. El costo de ejecución ya no representa un gran problema para los usuarios pues debido al gran avance tecnológico que se está viviendo el hardware moderno ya no resulta tan costoso y se puede mantener el equipo por más tiempo debido a que los nuevos procesadores y memorias son más eficientes, aunque también podría avanzar más el campo del software debería ser un gran cambio para necesitar hardware más potente del que se está ofreciendo al mercado. Finalmente el rendimiento de un software es importante pues e esa forma se mantiene contento al usuario y esto nos beneficiara obteniendo más de ellos, teniendo así más ganancias con el software un claro ejemplo de ello es el cuphead un videojuego con pocos requisitos y vistoso que como consecuencia permite tener más mercado para su venta.

Conceptos

Optimización. La optimización es un proceso que tiene a minimizar o maximizar alguna variable de rendimiento, generalmente tiempo, espacio, procesador, etc.

Rendimiento. La idea rendimiento refiere a la proporción que surge entre los medios empleados para obtener algo y el resultado que se consigue.

El tiempo de ejecución. Es el período en el que un programa es ejecutado por el sistema operativo.

Compilador. Un compilador es un programa informático, que se encarga de traducir el código fuente de una aplicación que este en desarrollo, es decir convierte un programa hecho en lenguaje de programación de alto nivel a un lenguaje de máquina, el cual es conocido como de bajo nivel, de tal forma que sea más entendible y mucho más fácil de procesar en el equipo en el que se está ejecutando.

Espacio. Se refiere a la cantidad de espacio de datos disponible en el disco duro

Procesador. Chip de distintos tipos, formando múltiples microprocesadores en conexión, un microprocesador típico se compone de registros, unidad de control, unidad aritmética-lógica, entre otras.

Folding. Es remplazar las expresiones por su resultado cuando se pueden evaluar en tiempo de compilación (resultado constante).

Ejemplo: A=2+3+A+C -> A=5+A+C

Ejemplo de Folding

Expresión: `3-(5+6)-A*10+4`

Árbol:

```
-
+- +
|  +- -
|  |  +- 3
|  |  +- +
|  |     +- 5
|  |     +- 6
|  +- *
|     +- A
|     +- 10
+- 4
```

Términos:
3
-5
-6
-(A*10)
4

Términos constantes:
3-5-6+4 = -4

Resultado: `-4-(A*10)`

Propagación de constantes. Desde que se asigna a una variable un valor constante hasta la siguiente asignación, se considera a la variable equivalente a la constante.

Ejemplo: Propagación Ensambla miento

PI=3.14 -> PI=3.14 -> PI=3.14

G2R=PI/180 -> G2R=3.14/180 -> G2R=0.017

PI y G2R se consideran constantes hasta la próxima asignación.

Reducción de potencia. Reemplazar una operación por otra equivalente menos costosa

x2 ! x * x

2*x ! x + x (suma); x<<1 (despl. izq.)

4*x, 8*x,... ! x<<2, x<<3,...

x / 2 ! x>>2

Ciclo. Es una sentencia que se realiza repetidas veces a un trozo aislado de código, hasta que la condición asignada a dicho bucle deje de cumplirse.

Instrucciones de bifurcación. Interrumpen el flujo normal de un programa, es decir que evitan que se ejecute alguna instrucción del programa y salta a otra parte del programa.

Ofuscadores. Es el proceso que intenta modificar un programa para hacer más compleja su comprensión.

Depuradores. Es un ejecutable cuya misión es permitir la ejecución controlada de un segundo ejecutable. Se comporta como un envoltorio dentro del cual se desarrolla una ejecución normal de un programa, pero a la vez permite realizar una serie de operaciones específicas para visualizar el entorno de ejecución en cualquier instante.

Desensambladores. Es exactamente lo contrario de un ensamblador. Tal como un ensamblador convierte código escrito en ensamblador en código máquina binario, un desensamblador invierte el proceso e intenta recrear el código en ensamblador partiendo del código máquina binario.

Bibliografía o referencias.

http://itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20III.pdf

http://noeliy22.blogspot.mx/2013/11/tipos-de-optimizacion.html

http://carlossotero.blogspot.mx/

https://definicion.de/rendimiento/

http://www.alegsa.com.ar/Dic/tiempo_de_ejecucion.php

http://ingsistemascompilador.blogspot.mx/p/conceptos-basicos-sobre-compiladores.html

http://conceptodefinicion.de/procesador/

https://prezi.com/nnlt6_puphob/que-es-un-ciclo-en-la-programacion-bucle/

https://eperdomo89.wordpress.com/2010/02/28/tipos-de-instrucciones/

http://informatica.blogs.uoc.edu/2015/10/26/ofuscacion-de-codigo-te-reto-a-que-entiendas-este-programa/