TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Unidad 3

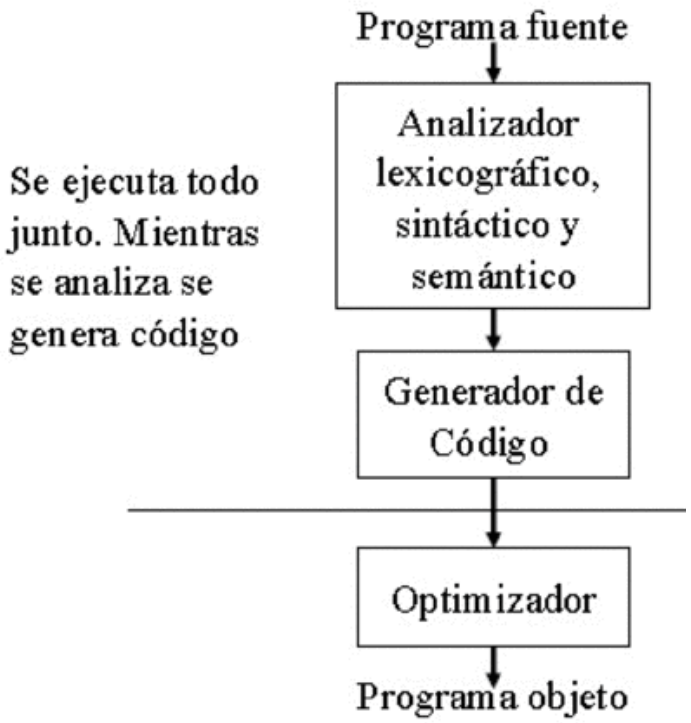
ALUMNOS- Christiam Iván Rodríguez moreno
NO.CONTROL: 13480619
CARRERA: Ingeniería en Sistemas Computacionales
ASESOR INTERNO: Juan Pablo Rosas Baldazo

Cd. Guadalupe, N.L. Mayo, 2018

Índice

Introducción

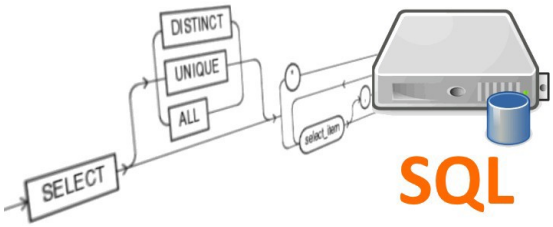
Las optimizaciones se realizan en base al alcance ofrecido por el compilador Y va a depender del lenguaje de programación y es directamente proporcional es un proceso que tiene a minimizar o maximizar alguna variable de rendimiento, generalmente tiempo, espacio, procesador Veremos varios tipos de optimizaciones tales como: Optimizaciones Globales,Optimizaciones de Ciclo,Optimización de Mirilla,Optimizaciones Locales Tipos de optimización Dentro de los tipos de optimización se derivan los tipos de optimización local, optimización de ciclo, optimización global y optimización de mirilla.



Optimización

La optimización va a depender del lenguaje de programación y es directamente proporcional al tiempo de compilación; es decir, entre más optimización, más tiempo de compilación. Algunos editores ofrecen una versión de depuración y otra de entrega o final.
La optimización es un proceso que tiene a minimizar o maximizar alguna variable de rendimiento, generalmente tiempo, espacio, procesador, entre otros. Desafortunadamente no existen optimizador que hagan un programa más rápido y que ocupe menor espacio.

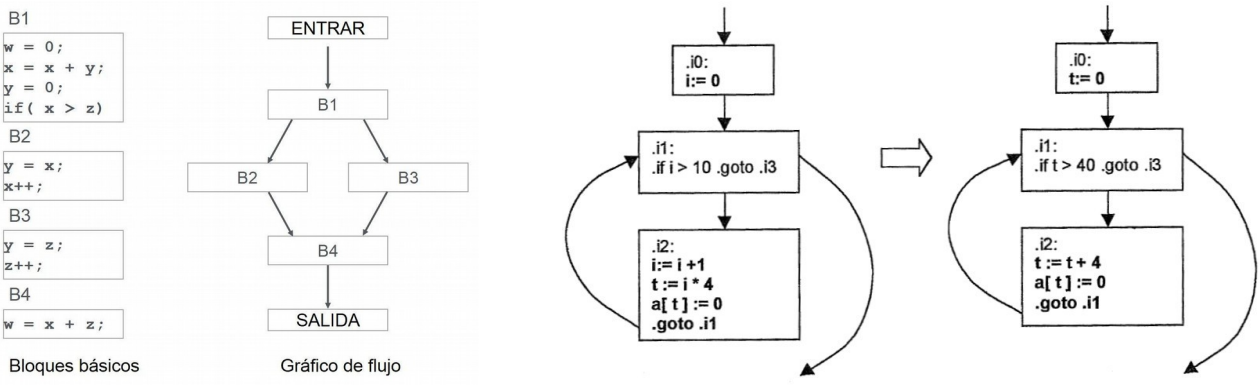
Locales:La optimización local se realiza sobre módulos del programa. En la mayoría de las ocasiones a través de funciones, métodos, procedimientos, clases, etc. La característica de las optimizaciones locales es que solo se ven reflejados en dichas secciones. La optimización local sirve cuando un bloque de programa o sección es crítico por ejemplo: E/S, la concurrencia, la rapidez y confiabilidad de un conjunto de instrucciones. Como el espacio de soluciones es más pequeño la optimización local es más rápida. Como el espacio de soluciones es más pequeño la optimización local es más rápida.



Ciclos : Los ciclos son una de las partes más esenciales en el rendimiento de un programa dado que realizan acciones repetitivas, y si dichas acciones están mal realizadas, el problema se hace n veces más grandes. La mayoría de las optimizaciones sobre ciclos tratan de encontrar elementos que no deben repetirse en un ciclo. 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 problema de la optimización en ciclos y en general radica es que muy difícil saber el uso exacto de algunas instrucciones. Así que no todo código de proceso puede ser optimizado. Otros uso de la optimización pueden ser el mejoramiento de consultas en SQL o en aplicaciones remotas (sockets, E/S, etc.)



Folding (Ensamblamiento)

Este consiste en reemplazar las expresiones por su resultado cuando se pueden evaluar en tiempo de compilación (resultado constante) permite que el programador utilice cálculos entre constantes representados explícitamente sin introducir in eficiencias

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

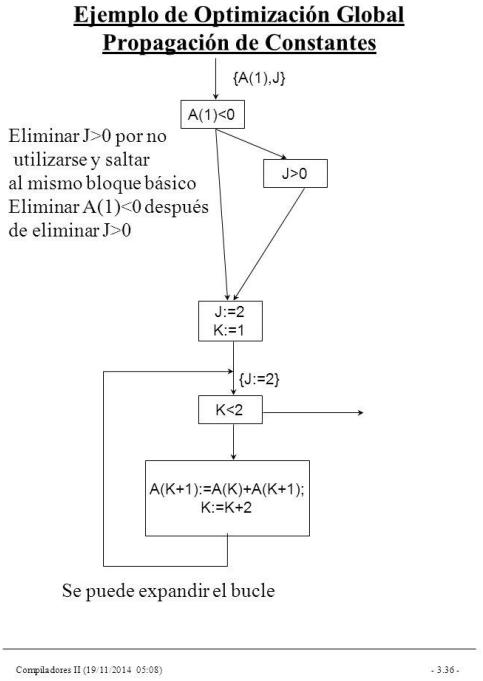
Resultado: `-4-(A*10)`

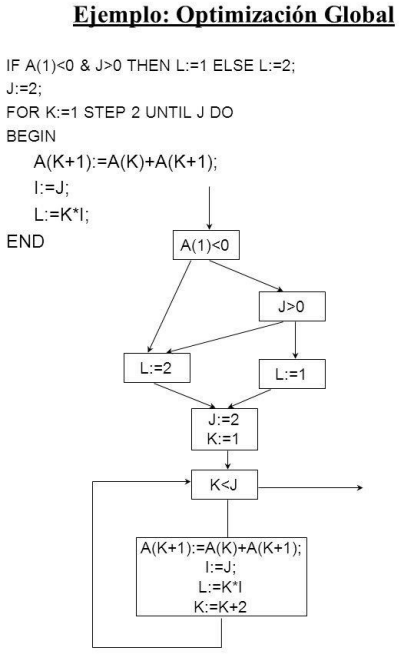
Globales

La optimización global se da con respecto a todo el código. Este tipo de optimización es más lenta pero mejora el desempeño general de todo programa. Las optimizaciones globales pueden depender de la arquitectura de la máquina. En algunos casos es mejor mantener variables globales para agilizar los procesos (el proceso de declarar variables y eliminarlas toma su tiempo) pero consume más memoria. Algunas optimizaciones incluyen utilizar como variables registros del CPU, utilizar instrucciones en ensamblador.

Tipos de optimización global

- Estocásticos: Evalúan f sobre una muestra aleatoria sobre la religión de interés. Sin embargo, la eficiencia es una característica de estos métodos ya que a problemas de gran escala (+100 variables) son resueltos mejor con estos métodos
- Determinísticos: No envuelve ningún elemento aleatorio de forma más confiable por medio de métodos de puntos. Estos métodos calculan valores de la función sobre puntos de muestras (incapaces de resolver confiable mente un problema de optimización global). Otro método también es el método de bordes, este ya mencionado consiste en calcular cotas sobre conjuntos compactos. Si son implementados apropiadamente y considerando errores de redondeo, pueden producir soluciones globales rigurosas





De mirilla

La optimización de mirilla trata de estructurar de manera eficiente el flujo del programa, sobre todo en instrucciones de bifurcación como son las decisiones, ciclos y saltos de rutinas. La idea es tener los saltos lo más cerca de las llamadas, siendo el salto lo más pequeño posible. 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.

Por ejemplo:

el “break” Switch (expresión que estamos evaluando)

{ Case 1: cout << “Hola” ;

Break;

Case 2: cout << “amigos”; Break; }

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

Registros

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.

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 direccionales por medio de una viñeta, que es una dirección de memoria. Los bits, por conveniencia, se enumeran de derecha a izquierda. Estos registros están divididos en seis grupos los cuales tienen un fin específico:

Registros de segmento

- Registros de apuntador de instrucciones
- Registros apuntadores
- Registros de propósito general

Memoria Estática

La forma más sencilla de almacenar el contenido de una variable en memoria, en tiempo de ejecución, es hacerlo en la memoria estática. Así, el almacenamiento de dichas variables será permanente (durante la ejecución del programa). Por ello, resulta obvio que los datos etiquetados como constantes y las variables globales de un programa tengan asignada la memoria necesaria durante toda la ejecución del programa. Sin embargo, no todas las variables pueden almacenarse estáticamente.

Para que una variable pueda ser almacenada en memoria estática, es necesario conocer su tamaño (número de bytes necesario para su almacenamiento) en tiempo de compilación. Como consecuencia, aunque una variable (u objeto) sea de ámbito global, no podrán ocupar almacenamiento estático.

Pilas

La aparición de lenguajes con estructura de bloque trajo consigo la necesidad de técnicas de alojamiento en memoria más flexibles, que pudieran adaptarse a las demandas de memoria durante la ejecución del programa.
En general los compiladores, la asignación de memoria de variables locales se hace de una forma más flexible, atendiendo al hecho de que solamente necesitan memoria asignada desde el momento que comienza la ejecución de la función hasta el momento en que esta finaliza.
Así, cada vez que comienza la ejecución de un procedimiento (o función) se crea un registro de activación para contener los objetos necesarios para su ejecución, eliminándolo una vez terminada esta.
Dado que durante la ejecución, un programa es habitual que unos procedimientos llamen a otros y estos a otros, sucesivamente, se crea cadena jerárquica de llamadas a procedimiento.
Dado que la cadena de llamadas está organizada jerárquicamente, los distintos registros de activación asociados a cada procedimiento (o función) se colocaran en una pila en la que entraran cuando comience la ejecución del procedimiento y saldrán al terminar el mismo.

Costos de ejecución

los costos de ejecución son aquellos que vienen implicictos al ejecutar los programas en algunos programas,por lo que el espacio y la velocidad del microprocesador son elementos que se deben optimizar para tener un mercado potencial mas alto Otro tipo de aplicaciones que deben optimizarse son las aplicaciones para dispositivos móviles los dispositivos móviles tienen recursos mas limitados que un dispositivo de computo convencional razón para mejor uso de memoria y otros recursos de hadware

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 mas legible
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 civil y código para dispositivos móviles

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.

1. Los usuarios y otras personas de la empresa que forman parte del proceso bajo estudio comprenden con facilidad anotaciones sencillas. Por consiguiente, los analistas pueden trabajar con los usuarios y lograr que participen en el estudio de los diagramas de flujo de datos.

2. Los usuarios pueden hacer sugerencias para modificar los diagramas con la finalidad de describir la actividad con mayor exactitud. Asimismo pueden examinar las gráficas y reconocer con rapidez problemas

3. El análisis de flujo de datos permite a los analistas aislar áreas de interés en la organización y estudiarlas al examinar los datos que entran en el proceso

4. A medida que los analistas reúnen hechos y detalles, comprenden mejor el proceso

Conclusiones

pudimos conocer los tipos de optimizaciones para nuestro código de programación, algo importante a resaltar aquí es que nosotros debemos tomar en cuenta la información de cada uno, hay una optimizaciones que son rápidas pero carece de encontrar y arreglar errores, como la de tardarse en compilar el código pero es la más eficiente al de momento de la compilación.
los tipos de optimizadores ya sean normales cíclicos etc como poder evaluar algunos criterios ya sea el código las herramientas de flujo de datos que es mas fácil gestionar los recursos de un celular móvil que los de una cpu etc Otra parte importante que debemos tomar en cuenta serían los costos de ejecución para nuestra maquina o incluso nuestro mismo código, ya que debemos de tomar en cuenta varios factores para que, tanto nuestra maquina funcione en condiciones óptimas como la de que nuestro código sea compilado de manera más rápida y sin necesidad de utilizar tantos recursos que nos podremos ahorrar si podemos optimizar el mismo código.

Reporte

Lo que entendí en cuestión a la investigación fue que es un optimizador y lo que dice de los optimizadores que es un proceso para maximizar el rendimiento y ahorrar espacio y tiempo hay diferentes tipos de optimizadores y uno de ellos es el optimizador local el cual puede guardar tus variables de entrada y utilizar una sola salida al momento de compilarlo.
Otro tipo de optimizador es el bucle en el cual se utilizan acciones repetitivas y el cual se puede utilizar el numero de veces que tu gustes otos son los ciclos el problema de los ciclos es que se le tienen que dar opciones muy especificas al momento de implementarlo ya que es muy difícil saber el uso de sus instrucciones los optimizadores globales son los que ayudan a desempeñar los problemas generales y la mayoría depende de una arquitectura de computadora la optimización de mirilla busca la manera de optimizar el flujo de programas los costos de ejecución son aquellos los cuales sirven para ejecutar el mínimo espacio y microprocesadores los cuales son para optimizar al mercado potencial

Conceptos

Desensamblador: Es un programa de computador que traduce el lenguaje de maquina a lenguaje ensamblador
Código: Conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecutar dicho programa
Flujo de datos: Se utiliza para hacer varias cosas entre ellas trabajos y tareas. Es una representación gráfica del flujo de datos a través de un sistema de información
Análisis: Etapa del ciclo de vida de un sistema informático, esta etapa los analistas se encargan de analizar los requisitos del sistema
Memoria: Dispositivo que retiene, memoriza o almacena datos informáticos durante algún periodo de tiempo
Analizador lexicográfico: Es la primera fase de un compilador consistente en un programa que recibe como entrada el código fuente de otro programa y produce una salida compuesta de tokens
Generador de código: Es una de las fases mediante el cual un compilador convierte un programa sintácticamente correcto en una serie de instrucciones a ser interpretadas por una maquina
Lenguaje de programación: Lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por maquinas como las computadoras
Compilación: Es el proceso por el cual se traducen las instrucciones escritas en un determinado lenguaje de programación a lenguaje máquina.
Sockets: Designa un concepto abstracto por el cual dos programas pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada
Método de puntos: Consiste en asignar una cantidad de puntos a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos
Bifurcación: Cuando se aplica en el contexto de un lenguaje de programación o un sistema operativo, hace referencia a la creación de una copia de sí mismo por parte de un programa, que entonces actúa como un proceso hijo del proceso originario, ahora llamado “padre”

Biblografia

Bibliografía

Beck. Software de Sistemas, Introducción a la programación de Sistemas. Addison-Wesley Iberoamericana.
Guerra Crespo. Héctor. Compiladores. Ed. Tecnológica didáctica.
Vázquez Gaudioso. Elena, García Saiz. Tomas. Introducción a la teoría de autómatas, gramática y lenguajes. Ed. Universitaria Ramón Areces