

INSTITUTO TECNOLÓGICO DE NUEVO LEÓN



Lenguajes y Autómatas II

UNIDAD # 3

Nombre de la Unidad:

Optimización

Catedrático: Juan Pablo Rosas

Nombre: Alondra Sarahi Sanchez Saucedo

Matrícula: 14480840

Guadalupe N.L. a 15 de 04 del 2018

Introducción

En el siguiente documento se podrá mostrar la información referente a la unidad tres de optimización de la materia Lenguajes y Autómatas II en el cual se hará referencia a los temas de la unidad, así mismo se describirá una pequeña información acerca de los diferentes subtemas que tienen en si nos viene hablando sobre la optimización de los programas y sus diferentes tipos, sus costos, algunos criterios para mejorar el código y las herramientas del flujo de datos

Esta información nos ayudara a entender mejor los conceptos para después aplicarlos en la programación, pero como siempre es bueno tener un poco de teoría antes para entender mejor con lo que estaremos trabajando en la unidad tres; es muy importante analizar la información en caso de alguna duda que se tiene para así tener un mayor concepto de la información y poder trabajar mejor a la hora de generar códigos de programación

Al igual encontraran algunos conceptos claves los cuales decidí fuesen importantes por que muchas veces ves las palabras en la lectura, pero no sabes el significado entonces en este documento cuenta con algunas palabras que puede ser importante saber que significa y entender mejor su concepto.

Capítulo 1: Tipos de optimización

La optimización busca mejorar la forma en que un programa utiliza los recursos. Las optimizaciones se realizan en base al alcance ofrecido por el compilador. La optimización va a depender del lenguaje de programación y es directamente proporcional al tiempo de compilación; es decir, entre más optimización mayor tiempo de compilación.

○ Sección 1.1: Locales

La optimización local se realiza sobre módulos del programa. En la mayoría de las ocasiones a través de funciones, métodos, procedimientos, clases, etc.

Las características 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: la E/S, la concurrencia, la rapidez y confiabilidad de un conjunto de instrucciones

○ Sección 1.2: 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.

El problema de la optimización en ciclos y en general radica en que es muy difícil saber el uso exacto de algunas instrucciones. Así que no todo código de proceso puede ser optimizado. Otro uso de la optimización puede ser el mejoramiento de consultas en SQL o en aplicaciones remotas (sockets, E/S, etc.)

○ Sección 1.3: 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

○ Sección 1.4: 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.

Ideas básicas:

Se recorre el código buscando combinaciones de instrucciones que pueden ser reemplazadas por otras equivalentes más eficientes.

Se utiliza una ventana de n instrucciones y un conjunto de patrones de transformación (patrón, secuencias, remplazan).

Las nuevas instrucciones son reconsideradas para las futuras optimizaciones.

Ejemplos:

Eliminación de cargas innecesarias

Reducción de potencia

Eliminación de cadenas de saltos

Capítulo 2: Costos

○ Sección 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 del microprocesador son elementos que se deben optimizar para tener un mercado potencial más amplio

Memoria

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

Registros

Los registros del programador se emplean para controlar instrucciones en ejecución, maneja direccionamiento de memoria y proporcionar capacidad aritmética los registros son específicos 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

Registro de segmento

Registro de apuntador de instrucciones

Registro de apuntadores

Registro de propósito general

Registros índices

Registro de banderas

Pilas

La aparición de lenguajes con estructura de bloqueo 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 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 se 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 de una vez terminada esta 4dado que durante la ejecución de un programa es habitual que unos procedimientos llamen a otros y estos a otros, sucesivamente, se crea cadena jerárquica de llamadas procedimiento.

Dado que la cadena de llamadas esta organizada jerárquicamente, los distintos registros de activación asociado a cada procedimiento o función se colocan en una pila en la que entran cuando comience la ejecución del procedimiento y saldrán al terminar el mismo

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.

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 des ambladores.

La optimización al igual que la programación es un arte y no se ha podido sistematizar del todo

Conclusión

Cómo se puede observar en el documento se estuvo hablado de los temas los cuales aportaba la unidad tres de optimización, para esto se hizo un pequeño resumen de lo más importante de cada uno de los subtemas en si el tema nos viene mostrando la optimización y sus diferentes tipos de engloba

la optimización y sus tipos son esos procesos que nos ayudan a realizar la ejecución de código más rápido y eficiente esto es bueno ya que podrás ahorrar tiempo al momento de estarlo realizado optimizas las tareas o algunas de las diferentes acciones que se estén realizando o llevando a cabo dentro del programa

hay diferentes tipos como ya los mencione cada uno de ellos engloba diferentes puntos, y otros son más rápidos etc. va dependiendo que optimización sea pero es recomendable hacer la optimización ya que muchas veces te ayuda a identificar los posibles errores que se tiene o te ayuda eliminar líneas que son innecesarias otro de los puntos son los costos de ejecución y uno de los más importantes es la memoria ya que muchas veces los programas utilizan demasiada memoria y eso no es bueno para el equipo pues muchas veces este se traba o deja de funcionar es por eso que la optimización también lo beneficia mucho al momento de hacer programas demasiado pesados y no contar con un buen equipo en el cual se pueda correr correctamente y sin tener alguna falla en si tenemos que tener en cuenta siempre que generemos algún código de programación contar con los requerimientos necesarios para que este funcione correctamente, y no olvidemos uno de los puntos más importantes que es la optimización está siempre nos va beneficiar al momento de estar generando o desarrollando programas nos permitirá trabajar mejor en los tiempos y nos ayuda a tener un código más limpio y poder tener un mejor manejo

hoy en día existen diferentes programas que te ayudan en la optimización de tu código estas herramientas te permiten hacer mas ligero tu código y eliminar todo aquello que no necesitas así te facilitan el uso y mejor el aspecto de este

Conceptos

Compilador: Un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio (bytecode), o simplemente texto. Este proceso de traducción se conoce como compilación

Módulos: es una parte autónoma de un programa de ordenador

Bloque de programa: El bloque de código es una sección de código con una o más declaraciones y sentencias.

Concurrencia: Reunión en un mismo lugar de personas, sucesos o cosas, Acción de contribuir con otros a un resultado.

Ciclos: Serie de fases o estados por las que pasa un acontecimiento o fenómeno y que se suceden en el mismo orden hasta llegar a una fase o estado a partir de los cuales vuelven a repetirse en el mismo orden

Instrucciones: referencia a la acción de instruir (enseñar, adoctrinar, comunicar conocimientos, dar a conocer el estado de algo). La instrucción es el caudal de conocimientos adquiridos y el curso que sigue un proceso que se está instruyendo.

Bifurcación: División de una cosa en dos ramales, brazos o puntas.

implícitos: Que está incluido en una cosa, sin que esta lo diga o lo especifique.

gestor de memoria: es el acto de gestionar la memoria de un dispositivo informático. El proceso de asignación de memoria a los programas que la solicitan

Registro de segmento: se utiliza para alinear en un límite de párrafo o dicho de otra forma codifica la dirección de inicio de cada segmento y su dirección en un registro de segmento supone cuatro bits 0 a su derecha.

Registro de apuntador de instrucciones: El registro apuntador de instrucciones (IP) de 16 bits contiene el desplazamiento de dirección de la siguiente instrucción que se ejecuta.

El registro IP está asociado con el registro CS en el sentido de que el IP indica la instrucción actual dentro del segmento de código que se está ejecutando actualmente.

Registro de apuntadores: Los registros apuntadores están asociados con el registro SS y permiten al procesador acceso a datos en el segmento de pila. Los registros apuntadores son dos:

El registro SP El registro BP

Registros índices: Los registros SI y DI están disponibles para direccionamientos indexados y para sumas y restas. Que son la operación de puntero.

Registro de banderas: Los registros de banderas sirven para indicar el estado actual de la máquina y el resultado del procesamiento. Cuando algunas instrucciones piden comparaciones o cálculos aritméticos cambian el estado de las banderas.

Las banderas están en el registro de banderas en las siguientes posiciones:

bits 15 14 13 12 11 10 9 8 7 6 5

4 3 2 1

Ofuscadores: se refiere a encubrir el significado de una comunicación haciéndola más confusa y complicada de interpretar.

Bibliografía o Referencia

<http://gaferz.blogspot.mx/2013/11/tipos-de-optimizacion.html>

Gabriela Fernández Espinoza. (Noviembre 13,2013). *TIPOS DE OPTIMIZACION*. Abril 15,2018, de Blogger Sitio web: view-source:<http://gaferz.blogspot.mx/2013/11/tipos-de-optimizacion.html>

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

Sin autor. (2013). *Unidad III*. Noviembre 13, 2018, de S/N Sitio web:

<https://prezi.com/m-ft53psccpy/321-costo-de-ejecucion-memoria-registros-pilas/>

Juan Jose Sanchez . (enero 19,2014). *Costo de Ejecución* . abril 15,2018, de Prezi Sitio web: <https://prezi.com/m-ft53psccpy/321-costo-de-ejecucion-memoria-registros-pilas/>

http://dsc.itmorelia.edu.mx/~jcolivares/courses/ps207a/ps2_u7.pdf

Juan Carlos Rojas. (Febrero 7,2014). *Optimización*. Abril15,2018, de itmorelia.edu.mx Sitio web: http://dsc.itmorelia.edu.mx/~jcolivares/courses/ps207a/ps2_u7.pdf