



INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

LENGUAJES Y AUTOMATAS II

Proyecto Unidad III

DIEGO SALVADOR FERNANDEZ GARCIA  
15480236

ISC

ING. ROSAS BALDAZO JUAN PABLO

15/04/2018

## Índice

Introducción	03
Capítulo 1 tipos de Optimización	04
Capítulo 2 Costos	05
Conceptos	08
Reporte	10
Conclusión	11
Bibliografía	12

## Introducción

El siguiente proyecto constara de una investigación de los tipos de Optimización tales como Locales, Ciclos, Globales, Mirilla y los Costos de ejecución, los criterios y las herramientas, Esto va acompañado de un resumen de los siguientes temas donde explicara cómo es cada una de las optimizaciones.

La optimización depende de un lenguaje de programación y es proporcional al tiempo de compilación.

Se resumirá en conceptos las partes más importantes de cada uno de los temas siguientes donde se explicará cada uno de los conceptos.

## Capítulo 1: Tipos de Optimización

La optimización depende del lenguaje de programación y es proporcional al tiempo de compilación, entre más optimización mayor tiempo de compilación.

Es un proceso que tiene a minimizar o maximizar alguna variable de rendimiento, generalmente tiempo, espacio, procesador, etc.

### Sección 1.1: Locales

Se realiza sobre módulos del programa. En muchas ocasiones a través de funciones, métodos, procedimientos, clases, etc. Su característica 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. Como el espacio de solución es más pequeño la optimización local es más rápida.

### Sección 1.2: Ciclos

Es muy importante ya que son una parte esencial en el rendimiento de un programa dado que realizan acciones repetitivas, y si dichas acciones están mal realizadas, el problema hace N veces más grandes. La mayoría de las optimizaciones sobre los ciclos tratan de encontrar elementos que no deben repetirse en un ciclo.

Su problema es que es difícil de saber el uso exacto de algunas instrucciones. No todo código de proceso puede ser optimizado.

### Sección 1.3: Globales

Optimización entre bloques básicos grafo de flujo de ejecución, antes de realizar una optimización global es necesario crear el grafo de flujo de ejecución.

El grafo de flujo de ejecución representa todos los caminos posibles de ejecución del programa.

### 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.

## Capítulo 2: Costos

Son uno de los factores más importantes al Optimizar, la mejora puede a veces no verse reflejada en el programa final.

La mejor manera de optimizar el código es hacer que los programadores lo optimicen desde el momento de su creación, el programa sería que se tardaría más en codificar esto representa tiempo y esfuerzo que representa un costo extra. Los criterios de optimización siempre están definidos por el compilador.

### 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.

Otro tipo de aplicación que deben optimizar son las aplicaciones para dispositivos móviles.

Los dispositivos móviles tienen recursos más limitados para un dispositivo de computo convencional razón por la cual, el mejor uso de memoria y otros recursos de hardware tienen mayor rendimiento.

#### 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.

Algún lenguaje de programación utiliza 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 borrados cuando el procedimiento termina.

#### 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 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

Segmento, Apuntador de Instrucciones, Apuntadores, Propósito General, Índices, Banderas.

## Pila

La aparición de lenguajes de 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 variable 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 esta finaliza.

## Sección 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 se tendría que codificar más y hacer el 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.

## Sección 2.3: Herramientas para el análisis del flujo de datos

Existen algunas herramientas que permiten el análisis y la correcta optimización del flujo de datos entre las más importantes están:

Depurador, desamblador, diagrama de flujo y diccionario de datos

### Depurador

Es una aplicación que permite correr otros programas, permitiendo al usuario ejercer cierto control sobre los mismos a medida que los estos se ejecuten, y examinar el estado del sistema en el momento en que se presente algún problema.

Permite detener el programa en

Punto determinado mediante un punto de ruptura, bajo ciertas condiciones mediante un punto de ruptura condicional, cuando se cumplan ciertas condiciones , a petición del usuario.

### Desamblador o Desensamblador

Un programa de computadora que traduce el lenguaje de maquina a lenguaje ensamblador, la operación inversa de la que hace el ensamblador.

Un desensamblador se diferencia de un descompilador, en que está dirigido a un lenguaje de alto nivel en vez de al lenguaje ensamblador,

El desensamblador transforma el código binario en instrucciones básicas del pc en la que se ejecuta, el programa no es capaz de distinguir si dichas instrucciones provienen de un “if”, o de un “for” o de un “while”. No conoce los nombres de las variables así que suele inventar nombres como VAR\_0001, VAR\_0002, etc.

#### Diagrama de flujo de datos

Es una herramienta de modelización que permite describir, de un sistema, la transformación de entradas en salidas; el DFD también es conocido con el nombre de Modelo de procesos de Negocios.

#### Diccionario de datos

Es un listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas y rigurosas que le permite al usuario y al proyectista del sistema tener una comprensión de las entradas, de las salidas y también de cálculos intermedios.

## Conceptos más importantes

**Optimización:** El proceso que consiste en encontrar los mínimos y máximos de una función.

**Lenguaje de programación:** Es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

**Compilación:** Es el proceso por el cual se traducen las instrucciones escritas en un determinado lenguaje de programación a lenguaje máquina.

**Procesador:** Programa informático que procesa o somete a una serie de operaciones la información introducida en la computadora.

**Funciones:** Es una sección de un programa que calcula un valor de manera independiente al resto del programa.

**Métodos:** Es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia.

**Ciclos:** Es una sentencia que ejecuta repetidas veces un trozo de código, hasta que la condición asignada a dicho bucle deja de cumplirse.

**Instrucciones:** En informática al conjunto de datos insertados en una secuencia estructurada o específica que el procesador interpreta y ejecuta.

**Grafo de flujo:** Es una representación, en forma de grafo dirigido, de todos los caminos que pueden ser atravesados a través de un programa durante su ejecución.



**Bifurcación:** 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. Los procesos resultantes son idénticos.

**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.

**Bit:** Es un dígito del sistema de numeración binario.

**Ejecución:** Realización de una acción, especialmente en cumplimiento de un proyecto, un encargo o una orden.

## REPORTE

Tipos de Optimización: son las variables donde uno tiene que organizar el tiempo y ver cual lenguaje de programación se puede utilizar.

Local: solamente es necesario utilizarla cuando algún programa es crítico y es realizado solo en los módulos de un programa.

Ciclos: El rendimiento del programa es donde se checan los errores de un programa y trata de corregirlos.

Globales: Son los pasos para la creación de un programa.

Costos: Es la parte fundamental de un programa ya que con ellos se puede tener un control sobre la ejecución y los tiempos.

Ejecución: es la parte donde uno tiene que ver cuanta memoria y capacidad tiene su procesador para poder compilar un programa.

Criterios para mejorar el código: para mejorar el código es necesario que se tenga un tiempo estimado para poder que tu programa quede con tiempo y así no tener problemas.

Herramientas para el análisis del flujo de datos: se tiene que analizar antes de hacer un programa con diagramas, depuraciones para tener una pequeña base del programa.

## Conclusión

En esta unidad de lenguajes y autómatas II con el resumen me quedo más en claro cada uno de los siguientes conceptos, me pareció interesante ver el proceso por el cual se lleva cada una de las optimizaciones empezando por la local por la cual tiene que ser crítico para que se pueda ver reflejado, En los ciclos es la parte donde verifica el problema de un programa.

En los costos se entendió que es la parte más esencial en la optimización empezando los por los costos de ejecución ya que con el proceso se puede apoyar al codificador a tener mayor velocidad en el programa.

La memoria es un recurso importante de una computadora porque dependiendo la aplicación los y los requisitos que se pida la memoria hace su función al momento de almacenar los datos.

Por último, la parte más importante de un programador debería de ser que se optimizara el código desde un principio para que no tenga tantos problemas algunos.

## BIBLIOGRAFIA

Automatas|UnidadII|<http://itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20II.pdf>