

Programación Estructurada

INTRODUCCION

La computadora automática debe su derecho a existir, su utilidad, precisamente a su capacidad de efectuar vastos cálculos que no pueden realizar los seres humanos. Deseamos que la computadora efectúe lo que nunca podríamos hacer nosotros, y la potencia de las máquinas actuales es tal, que inclusive los cálculos pequeños, por su tamaño, escapan al poder de nuestra imaginación limitada.

Sin embargo debemos organizar el cálculo de manera tal que nuestros limitados poderes sean suficientes para asegurar que se establecerá el efecto deseado. Esta organización incluye la composición de los programas.

Los avances en la tecnología siempre van parejos con progresos en los lenguajes de programación y con nuevas ayudas para simplificar el uso del computador, con lo cual un número mayor de usuarios se beneficia del. Pero la necesidad de hacer programas para resolver problemas específicos quizás nunca desaparecerá.

CONCEPTO DE PROGRAMACION ESTRUCTURADA

EL creciente empleo de los computadores ha conducido a buscar un abaratamiento del desarrollo de software, paralelo a la reducción del costo del hardware obtenido gracias a los avances tecnológicos. Los altos costos del mantenimiento de las aplicaciones en producción normal también han urgido la necesidad de mejorar la productividad del personal de programación.

En la década del sesenta salieron a la luz pública los principios de lo que más tarde se llamó Programación Estructurada, posteriormente se liberó el conjunto de las llamadas "Técnicas para mejoramiento de la productividad en programación" (en inglés Improved Programming Technologies, abreviado IPTs), siendo la Programación Estructurada una de ellas.

Los programas computarizados pueden ser escritos con un alto grado de estructuración, lo cual les permite ser más fácilmente comprensibles en actividades tales como pruebas, mantenimiento y modificación de los mismos. Mediante la programación Estructurada todas las bifurcaciones de control de un programa se encuentran estandarizadas, de forma tal que es posible leer la codificación del mismo desde su inicio hasta su terminación en forma continua, sin tener que saltar de un lugar a otro del programa siguiendo el rastro de la lógica establecida por el programador, como es la situación habitual con codificaciones desarrolladas bajo otras técnicas.

EN programación Estructurada los programadores deben profundizar más que lo usual al procederá realizar el diseño original del programa, pero el resultado final es más fácil de leer y comprender, el objetivo de un programador profesional al escribir programas de una manera estructurada, es realizarlos utilizando solamente un número de bifurcaciones de control estandarizados.

EL resultado de aplicar la sistemática y disciplinada manera de elaboración de programas establecida por la Programación Estructurada es una programación de alta precisión como nunca antes había sido lograda. Las pruebas de los programas, desarrollados utilizando este método, se acoplan más rápidamente y el resultado final con programas que pueden ser leídos, mantenidos y modificados por otros programadores con mucho mayor facilidad.

DEFINICIONES

Programación Estructurada es una técnica en la cual la estructura de un programa, esto es, la interrelación de

sus partes realiza tan claramente como es posible mediante el uso de tres estructuras lógicas de control:

- Secuencia: Sucesión simple de dos o mas operaciones.
- Selección: bifurcación condicional de una o mas operaciones.
- Interacción: Repetición de una operación mientras se cumple una condición.

Estos tres tipos de estructuras lógicas de control pueden ser combinados para producir programas que manejen cualquier tarea de procesamiento de información.

Un programa estructurado esta compuesto de segmentos, los cuales puedan estar constituidos por unas pocas instrucciones o por una pagina o más de codificación. Cada segmento tiene solamente una entrada y una salida, estos segmentos, asumiendo que no poseen lazos infinitos y no tienen instrucciones que jamas se ejecuten, se denominan programas propios. Cuando varios programas propios se combinan utilizando las tres estructuras básicas de control mencionadas anteriormente, el resultado es también un programa propio.

La programación Estructurada esta basada en el Teorema de la Estructura, el cual establece que cualquier programa propio (un programa con una entrada y una salida exclusivamente) es equivalente a un programa que contiene solamente las estructuras lógicas mencionadas anteriormente.

Una característica importante en un programa estructurado es que puede ser leído en secuencia, desde el comienzo hasta el final sin perder la continuidad de la tarea que cumple el programa, lo contrario de lo que ocurre con otros estilos de programación. Esto es importante debido a que, es mucho más fácil comprender completamente el trabajo que realiza una función determinada, si todas las instrucciones que influyen en

su acción están físicamente contiguas y encerradas por un bloque. La facilidad de lectura, de comienzo a fin, es una consecuencia de utilizar solamente tres estructuras de control y de eliminar la instrucción de desvío de flujo de control, excepto en circunstancias muy especiales tales como la simulación de una estructura lógica de control en un lenguaje de programación que no la posea.

VENTAJAS POTENCIALES

Un programa escrito de acuerdo a estos principios no solamente tendrá una estructura, sino también una excelente presentación.

Un programa escrito de esta forma tiende a ser mucho más fácil de comprender que programas escritos en otros estilos.

La facilidad de comprensión del contenido de un programa puede facilitar el chequeo de la codificación y reducir el tiempo de prueba y depuración de programas. Esto ultimo es cierto parcialmente, debido a que la programación estructurada concentra los errores en uno de los factores más generador de fallas en programación: la lógica.

Un programa que es fácil para leer y el cual esta compuesto de segmentos bien definidos tiende a ser simple, rápido y menos expuesto a mantenimiento. Estos beneficios derivan en parte del hecho que, aunque el programa tenga una extensión significativa, en documentación tiende siempre a estar al día, esto no suele suceder con los métodos convencionales de programación.

La programación estructurada ofrece estos beneficios, pero no se la debe considerar como una panacea ya que el desarrollo de programas es, principalmente, una tarea de dedicación, esfuerzo y creatividad.

TEOREMA DE LA ESTRUCTURA

El teorema de la estructura establece que un programa propio puede ser escrito utilizando solamente las siguientes estructuras lógicas de control: secuencia, selección e iteración.

Un programa se define como propio si cumple con los dos requerimientos siguientes:

- Tiene exactamente una entrada y una salida para control del programa.
- Existen caminos seguíbles desde la entrada hasta la salida que conducen por cada parte del programa, es decir, no existen lazos infinitos ni instrucciones que no se ejecutan.

ETIQUETAS E INSTRUCCIÓN *GOTO*:

Ocasionalmente se habla de la programación estructurada como una técnica de programación que no utiliza *GOTO* (instrucción de desvío del flujo de control en forma incondicional); si bien es cierto que un programa bien estructurado tiene, o bien ninguna o muy pocas instrucciones *GOTO*, asumiendo que estamos empleando un lenguaje de programación adecuado, la ausencia de instrucciones *GOTO* puede ser mal interpretada. Es conveniente que aclaremos este aspecto en este momento.

Un programa bien estructurado gana una parte importante de su fácil comprensión del hecho que puede ser leído en forma secuencial sin desvíos en el flujo de control desde una parte del programa a otra. Esta característica es consecuencia de usar exclusivamente las estructuras lógicas de control estándar (el *GOTO* no es una de ellas), esta secuencialidad o lectura *TOPDOWN* es beneficiosa debido a que hay un límite definido para muchos detalles que la mente humana puede abarcar de una vez. Se hace relativamente fácil y rápida la comprensión de la tarea que realiza una instrucción si su función puede ser entendida en términos de unas pocas instrucciones más, físicamente contiguas y delimitadas.

El problema con la instrucción *GOTO* es que generalmente aleja al programa realizado de los propósitos descritos y en casos extremos puede hacer que un programa sea esencialmente incomprensible.

No se requieren esfuerzos especiales para eliminar de un programa los *GOTO*, los cuales han sido, algunas veces, malentendidos como enemigos de la programación estructurada, existen buenas y fundadas razones para no querer usarlos pero no se necesita que se realice un trabajo arduo para eliminarlos; ellos no aparecerán, en general, cuando se utilicen las estructuras lógicas de control, descritas anteriormente. Naturalmente, si escogemos para programar un lenguaje de computación que no posea las estructuras lógicas de control fundamentales, entonces, tendremos que simularlas y seguramente ello implicará el uso de la instrucción *GOTO*; pero este uso puede hacerse en forma cuidadosamente controlada.

Existen situaciones poco comunes en las cuales el uso de *GOTO* puede tener ventajas comparado con otras maneras de expresar un proceso; estos casos son excepcionales y usualmente no ocurren en la programación realizada diariamente.

Se deben analizar cuidadosamente las consecuencias de emplear el *GOTO*, antes de su uso.

SEGMENTACION

Para la comprensión de un programa se haga en forma fácil y rápida es necesario que, al revisarlo, uno no tenga que hojear una gran cantidad de páginas para entender cuáles el trabajo que realiza. Una regla práctica para lograr estos fines es establecer que cada segmento del programa no exceda, en longitud, a una página de codificación, o sea, alrededor de 50 líneas (el significado que se asigna al término segmento, en este trabajo, no tiene ninguna relación con su significado en relación a las funciones de sistemas operativos o sistemas manejadores de Bases de Datos).

La segmentación no es solamente particionar un programa en trozos cuya longitud sea de unas 50 líneas; esta

técnica debe cumplir con ciertas características fundamentales:

- La segmentación reflejara la división del programa en partes que se relacionen entre sí en forma jerárquica, formando una estructura de árbol. Esta organización puede ser representada gráficamente por un diagrama de procesos, lo que hace más sencillo comprender la relación existente entre un segmento y el resto del programa. Adicionalmente, podemos indicar que, el segmento en la cumbre de la estructura jerárquica contendrá las funciones de control de mas alto nivel, mientras que los segmentos inferiores en esta organización contendrán funciones detalladas.
- Una segmentación bien diseñada deberá mostrar, claramente, las relaciones existentes entre las distintas funciones de manera que sea fácil comprender lo que debe hacer el programa y asegurar que efectivamente lo realice. Este hecho, garantizara que los cambios que se efectúen a una parte del programa, durante la programación original o su mantenimiento, no afecten al resto del programa que no ha sufrido cambios.
- En una segmentación bien realizada la comunicación entre segmentos se lleva a cabo de una manera cuidadosamente controlada. Algunos autores recomiendan que los segmentos consistan en procedimientos y la única comunicación existente entre ellos sea a través de una lista de parámetros, esto reduce la oportunidad de que interactuen entre ellos de una manera indeseada e inentendible.

IDENTACION

El uso de la identacion es importante debido a que, cuando se es consistente en su utilización, facilita la lectura del programa al mostrar en una forma gráfica las relaciones existentes entre las distintas instrucciones.

La identacion puede ser de gran beneficio, tal como se muestra continuación, donde ambos programas realizan la misma función, pero el de la derecha es más fácil de comprender, verificar y corregir.

DIRECTRICES PAR IDENTAR

Debe comprenderse claramente que las líneas siguientes solo pretenden presentar unas directrices de identacion, sin pretender que estas sean las únicas reglas a seguir en este proceso, cada centro de procesamiento deberá establecer sus propias convenciones, sin que sea motivo de preocupación la diferencia respecto a las sugerencias dadas aquí, lo importante es que se establezcan unas normas y se cumplan de manera consistente.

Las siguientes son sugerencias para el desarrollo de una política de identacion en un centro de procesamiento, la idea fundamental es ayudar a que el lector de un programa le sea fácil comprender las relaciones y las funciones existentes en él:

- En los lenguajes donde se permite el uso de etiquetas, estas deben colocarse lo más externas posibles, por ejemplo comenzando en la columna 2, y deben estar separadas por una línea (siempre que lo permita el lenguaje en uso).
- Se obtiene consistencia si todas las instrucciones se comienzan en una misma columna, por ejemplo en la columna 4 o cualquier otra ubicada a su derecha.
- En los lenguajes en que se hagan declaraciones sobre las variables a utilizar, la información quedara mas claramente representada si los atributos declarados se alinean en forma vertical.
- El uso de líneas en blanco ayuda a mostrar con mas claridad las relaciones existentes entre distintos ítems agrupados en las declaraciones
- Las instrucciones son mucho mas fáciles de localizar y de cambiar si no se escribe mas de una instrucción por línea.
- La vision de control de las estructuras lógicas o de los bloques se clarifica si las instrucciones controladas son idénticas por alguna cantidad constante. Se sugiere una identacion de tres espacios.

VENTAJAS DE LA PROGRAMACION ESTRUCTURADA

Con la programación estructurada elaborar programas de computador sigue siendo un albor que demanda esfuerzo, creatividad, habilidad y cuidado. Sin embargo, con este nuevo estilo podemos obtener las siguientes ventajas:

- – Los programas son más fáciles de entender. Un programa estructurado puede ser leído en secuencia, de arriba hacia abajo, sin necesidad de estar saltando de un sitio a otro en la lógica, lo cual es típico de otros estilos de programación. La estructura del programa es mas clara puesto que las instrucciones están mas ligadas o relacionadas entre sí, por lo que es más fácil comprender lo que hace cada función.
- Reducción del esfuerzo en las pruebas. El programa se puede tener listo para producción normal en un tiempo menor del tradicional; por otro lado, el seguimiento de las fallas("debugging") se facilita debido a la lógica más visible, de tal forma que los errores se pueden detectar y corregir mas fácilmente.
- Reducción de los costos de mantenimiento.
- Programas más sencillos y más rápidos
- Aumento de la productividad del programador
- Se facilita la utilización de las otras técnicas para el mejoramiento de la productividad en programación
- Los programas quedan mejor documentados internamente.