INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Ing. Sistemas computacionales

"Lenguajes Autómatas 2"

Profesor: Juan Pablo Rosas Baldazo

Tarea: Reporte de unidad

Unidad 4

Presenta.

Mario Humberto Uriegas de León

No. De control: 14480514

Cd. Guadalupe; Nuevo León; a 03 de Mayo de 2018

Introducción

En el siguiente reporte se observara un resumen relacionado con el tema “generación de código objeto” con sus respectivos subtemas como registros con una explicación más amplia hablando de su distribución y su asignación, del lenguaje ensamblador observando sus características y su clasificación, en conjunto con el lenguaje maquina con sus características a la vez que ventajas y desventajas y por último se analizara administración de memoria.

Generación de código objeto

La fase final de un compilador es la generación de código objeto, que por lo general consiste en código de máquina re localizable o código ensamblador. Las posiciones de memoria se seleccionan para cada una de las variables usadas por el programa. Después, cada una de las instrucciones intermedias se traduce a una secuencia de instrucciones de máquina que ejecuta la misma tarea. El generador de código objeto puede considerarse como la penúltima fase de un compilador, la cual se encarga de tomar como entrada el código intermedio generado por el front-end, y producir código objeto de la arquitectura target.

Capítulo 1: Registros

Los registros son la memoria principal de la computadora. Existen diversos registros de propósito general y otros de uso exclusivo. Algunos registros de propósito general son utilizados para cierto tipo de funciones.
Los registros son espacios físicos dentro del microprocesador con capacidad de 4 bits hasta 64bits dependiendo del microprocesador que se emplee.

Sección 1.1: **Distribución**

La distribución es el proceso en el que el programa generado puede ejecutarse en otras máquinas. Con respecto al ensamblador, la mayoría del direccionamiento se hace relativo para que el programa sea re localizable por un programa llamado cargador.

En el caso de programas compilados se necesitan de las librerías, si son estáticas se incluyen en el ejecutable por lo que el programa se hace gráfico, si son dinámicas no, pero el programa es más pequeño.

El uso de registros puede dividirse en dos subproblemas:

- Durante la reserva de registros (allocation), se seleccionan el conjunto de variables que viviría en registros en un punto del programa.
- Durante la (posterior) asignación de registros (assignation), se elige el registro específico para cada variable.

Sección 1.2: **Asignación**

El ensamblador se encarga de ajustar el tamaño de los segmentos tomando como base el número de bytes que necesita cada instrucción que va ensamblando, ya que sería un desperdicio de memoria utilizar los segmentos completos. Por ejemplos, si un programa únicamente necesita 10kb para almacenar los datos, el segmento de datos únicamente será de 10kb y no de los 64kb que puede manejar.

Capítulo 2: Lenguaje ensamblador

El ensamblador es un traductor de un código de bajo nivel a un código, ejecutable directamente por la máquina para la que se ha generado.
Fue la primera abstracción de un lenguaje de programación, posteriormente aparecieron los compiladores.

Sección 2.1: **Características**

El programa lee un archivo escrito en lenguaje ensamblador y sustituye cada uno de los códigos mnemotécnicos por su equivalente código máquina.
Los programas se hacen fácilmente portables de máquina a máquina y el cálculo de bifurcaciones se hace de manera fácil.

Clasificación

- **Ensambladores básicos:** son de muy bajo nivel, y su tarea consiste básicamente, en ofrecer nombres simbólicos a las distintas instrucciones, parámetros y cosas tales como los modos de direccionamiento.
- **Ensambladores modulares, o macro ensambladores:** Descendientes de los ensambladores básicos, fueron muy populares en las décadas de los 50 y los 60, fueron antes de la generalización de los lenguajes de alto nivel. Un macroinstrucción es el equivalente a una función en un lenguaje de alto nivel.

Sección 2.2: **Almacenamiento**

Una de las principales ventajas del uso del ensamblador, es que se encarga de administrar de manera transparente para el usuario la creación de memoria, las bifurcaciones y el paso de parámetros. Además nos permite acceder directamente a los recursos de la máquina para un mejor desempeño.

Capítulo 3: Lenguaje máquina
Este proporciona poca o ninguna abstracción del microprocesador de un ordenador. El lenguaje máquina sólo es entendible por las computadoras. Se basa en una lógica binaria de 0 y 1, generalmente implementada por mecanismos eléctricos. En general el lenguaje máquina es difícil de entender para los humanos por este motivo hacemos uso de lenguajes más parecidos a los lenguajes naturales.

Sección 3.1: **Características**
El lenguaje máquina realiza un conjunto de operaciones predeterminadas llamadas micro operaciones.
Las micro operaciones sólo realizan operaciones del tipo aritmética (+,-,*,/) lógicas (AND, OR, NOT) y de control (secuencial, de control y repetitiva).
Algunos microprocesadores implementan más funcionalidades llamado **CISC**, pero son más lentos que los **RISC** ya que estos tienen registros más grandes.
Ventajas

- Mayor adaptación al equipo.
- Máxima velocidad con mínimo uso de memoria.

Desventajas

- Imposibilidad de escribir código independiente de la máquina.
- Mayor dificulta en la programación y en la comprensión de los programas.
- El programador debe conocer más de un centenar de instrucciones.
- Es necesario conocer en detalle la arquitectura de la máquina.

Sección 3.2: **Direccionamiento**
Es la forma en cómo se accede a la memoria. Recordar que un programa no puede ejecutarse sino se encuentra en memoria principal.
La forma de acceder a la memoria depende del microprocesador, pero en general existen dos tipos de direccionamiento:

- Directo: También recibe el nombre de direccionamiento absoluto y el acceso a las direcciones se hace de manera directa.
- Indirecto: También recibe el nombre de direccionamiento relativo y se basa a partir de una dirección genérica, generalmente el inicio del programa.

Capítulo 4: Administración de memoria
La administración de la memoria es un proceso hoy en día muy importante, de tal modo que su mal o buen uso tiene una acción directa sobre el desempeño de memoria. En general un ensamblador tiene un administrador de memoria más limitado que un compilador. En la mayoría de los lenguajes de programación el uso de memoria. Los lenguajes más recientes controlan el uso de punteros y tienen un programa denominado recolector de basura que se encarga de limpiar la memoria no utilizada mejorando el desempeño.

- **Reubicación:** La técnica de multiprogramación requiere que varios programas ocupen la memoria al mismo tiempo. Sin embargo no se sabe con anticipación donde será cargado cada programa por lo que no es práctico usar direccionamiento absoluto de memoria.
- **Organización física:** Debido al costo de una memoria principal rápida, éste se usa en conjunto con una memoria secundaria mucho más lenta (y por consiguiente, barata) a fines de extender su capacidad.
- **Organización lógica:** Aunque la mayor parte de las memorias son organizadas linealmente con un direccionamiento secuencial, esto difícilmente concuerde con el camino seguido por el programa, debido al uso de procedimientos, funciones, **subrutinas**, arreglos, etc.

Conclusiones

En este trabajo se observó la generación del código objeto como un tema introductorio mencionando que es la penúltima fase de un compilador, de los registros con su respectiva distribución dividida en dos subproblemas que son allocation y assignation, en asignación se determina el tamaño de los segmentos como base en el número de bytes, en el tema de lenguaje ensamblador se manejaron sus características y una clasificación con los ensambladores básicos y los modulares, al igual que en el anterior en el lenguaje maquina se vieron características y algunas ventajas y desventajas del mismo finalizando con su direccionamiento, y ya por ultimo en administración de memoria vimos su reubicación, su organización física y su organización lógica.

Conceptos

CISC: Es un modelo de arquitectura de computadores. Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permitir operaciones complejas entre operandos situados en la memoria o en los registros internos, en contraposición a la arquitectura RISC.

RISC: Es un tipo de diseño de CPU generalmente utilizado en microprocesadores o microcontroladores.

Subrutina: como idea general, se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica. Algunos lenguajes de programación, como Visual Basic .NET o Fortran, utilizan el nombre función para referirse a subrutinas que devuelven un valor.

Mnemotécnicos: Es un sistema sencillo utilizado para recordar una secuencia de datos, nombres, números, y en general para recordar listas de ítems que no pueden recordarse fácilmente.

Bibliografía

[1]M. en C. Pérez Pérez Isaías y L.S.C. Monroy Cedillo Jair Jonathan. (2013). Generador de Código objeto. 03/05/2018, de Universidad Autónoma del estado de Hidalgo Sitio web: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro32/autocontenido/autocon/137_generador_de__cdigo_objeto.html

[2] ----. (2012). Generación de código objeto. 03/05/2018, de wordpress.com Sitio web: https://ingarely.files.wordpress.com/2012/11/unidad-viii.pdf

[3] Mark Francisco. (2016). Generación de código objeto. 03/05/2018, de Blogspot Sitio web: http://acaurio.blogspot.mx/2016/11/unidad-4-generacion-de-codigo-objeto.html

[4] ------. (2015). Unidad IV generación de código objeto. 03/05/2018, de ITPN Sitio web: http://itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20IV.pdf

Reporte de aprendizaje

A como yo entendí la generación de código objeto es la fase final de un compilador aunque después dice que es la penúltima, bueno esta fase se encarga de agarra el código que genera el front-end, si el front-end interactúa con nosotros significa que nosotros hicimos el código intermedio, de los registros pues ahora sé que son la memoria de la computadora y que son espacios reales o físicos del microprocesador, la distribución es para el programa sirva en otras máquinas más bien seria como una adaptación en otros equipos, la asignación se encarga no desperdiciar memoria ajustando los segmentos, sobre el lenguaje ensamblador ese ya sabía que era un traductor de bajo nivel, después del ensamblador llegaron los compiladores, de sus características encontré una palabra que nunca había escuchado "mnemotécnicos" que en si es un sistema para recordar secuencias que no son fáciles de recordar, también que hay ensambladores básicos y modulares, el ensamblador nos permite ver los recursos de la maquina o usarlos de una forma directa, del lenguaje maquina se me quedo que es entendible para las computadoras porque eso ya lo sabía también lo de que usa el código binario, lo que desconocía era que era muy rápido con poco uso de memoria, y de la administración de memoria pues ahora se que un ensamblador tiene un administrador de memoria más limitado que un compilador, que los lenguajes mas recientes tienen programas como recolectores de basura que mejoran el desempeño limpiando la memoria no utilizada.