

Unidad IV: Generación de código objeto

4.1 Registro

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. Existen registros acumuladores, puntero de instrucción, de pila, etc.

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 relocalizable 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.

Debido a la complejidad del software actual se necesitan de asistentes para poder instalar y ejecutar un programa.

Operar sobre registros es más rápido y eficiente que operar sobre memoria. Por ello, la adjudicación eficiente de registros tiene un gran impacto en la performance.

El uso de registros puede dividirse en dos subproblemas:

- Durante la reserva de registros (allocation), se seleccionan el conjunto de variables que vivirá en registros en un punto del programa.

– Durante la (posterior) asignación de registros (assignment), se elige el registro específico para cada variable.

4.2 Lenguaje ensamblador

El ensamblador (del inglés assembler) 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.

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.

Ensambladores

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, antes de la generalización de los lenguajes de alto nivel. Un macroinstrucción es el equivalente a una función en un lenguaje de alto nivel.

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.

4.3 Lenguaje máquina

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.

Características

- El lenguaje máquina realiza un conjunto de operaciones predeterminadas llamadas microoperaciones.
- Las microoperaciones sólo realizan operaciones del tipo aritmética (+, -, *, /), lógicas (AND, OR, NOT) y de control (secuencial, decisión, repetitiva).
- El lenguaje máquina es dependiente del tipo de arquitectura. Así un programa máquina para una arquitectura Intel x86 no se ejecutará en una arquitectura Power PC de IBM (al menos de manera nativa).
- Algunos microprocesadores implementan más funcionalidades llamado CISC, pero son más lentos que los RISC ya que estos tienen registros más grandes.

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 e indirecto.

El direccionamiento directo también recibe el nombre de direccionamiento absoluto y el acceso a las direcciones se hace de manera directa. El

direccionamiento indirecto también recibe el nombre de direccionamiento relativo y se basa a partir de una dirección genérica, generalmente el inicio del programa.

Para acceder a una dirección relativa se suma a la dirección base el número de espacios de memorias necesarias.

El direccionamiento relativo hace a los programas relocalizables e independientes. Si la dirección base es el inicio de la memoria fija el direccionamiento pasa a ser un variante de direccionamiento absoluto.

4.4 Administración de memoria

Consiste en determinar la posición de memoria en la que los diferentes símbolos del programa almacenan la información

Depende de la estrategia utilizada para la gestión de memoria, el mecanismo puede variar.

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 punteros no estaba vigilado por lo que se tienen muchos problemas con 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.