Programación de Sistemas

Unidad 4. Cargador

Contenido

- Introducción
- Cargador
- Características Dependientes de la Máquina
- Cargador de Arranque

Introducción

Código Objeto

- Un programa en código objeto es aquel que contiene instrucciones ensambladas y valores de datos generados a partir del programa fuente

- Además especifica direcciones de memoria en donde serán cargadas

Procesos para Ejecución

- Antes de que un programa se ejecute se requieren tres procesos:
  - Cargado
    - Coloca el programa objeto en memoria para su ejecución.
  - Relocalización
    - Modifica el programa objeto de tal manera que pueda ser cargado en una dirección diferente de la especificada originalmente
  - Ligado
    - Combina dos o mas programas objeto separados y proporciona la información necesaria para permitir referencias entre ellos

Cargador

Cargador

- Un cargador es un programa de sistema que realiza la operación de cargado
- La función principal de un cargador es traer el programa objeto a la memoria para su ejecución
- También pueden realizar las operaciones de relocalización y ligado
- Algunos sistemas tienen un ligador que realiza la operación de ligado y un cargador por separado para el manejo de la relocalización y la carga

Cargador Absoluto

- Un cargador absoluto es aquel que no realiza operaciones de relocalización

- Todas las operaciones que realiza se pueden ejecutar en un solo paso

Funcionamiento

- El registro de Encabezado (H) es revisado para asegurar que el programa correcto ha sido cargado y que cabe en la memoria disponible
- Mientras existan registros de Texto (T), el código objeto de cada instrucción es colocado en la dirección de memoria indicada
- Cuando se encuentra el registro de FIN, el cargador salta a la dirección de inicio para comenzar la ejecución

Consideraciones Especiales

- En el código objeto generado por el ensamblador, cada digito es representado en su forma de carácter (por esa razón se tiene un byte por columna)

- Por ejemplo el código de operación para la instrucción STL se representa con el par de caracteres 1 y 4

- Cuando estos son ledos por el cargador, ocuparan dos bytes en memoria, sin embargo cuando la instrucción sea leída para su ejecución ocupara un solo byte con su código hexadecimal 14

Algoritmo del Cargador Absoluto

1. Leer registro de Encabezado
2. Verificar el nombre y longitud del programa
3. Leer el primer registro de Texto
4. MIENTRAS el tipo de registro 6 ≠ E
    1. SI el código objeto esta en forma de caracter, se convierte a su representación interna
    2. Mover el código objeto a la localidad especificada en memoria
    3. Leer el siguiente registro del código objeto
5. Saltar a la dirección especificada en el registro
6. FIN

Ejemplo

- Colocar en memoria el siguiente código

HCOPY 00100000107A

T0010001E14103348203900103628103030101548206 13C100300102A0C103900102D

T00101E150C10364820610810334C0000454F46000003000000

T0020391E041030001030E0205D30203FD8205D2810303020576490392C205E38203F

T0020571C1010364C0000F1001000041030E02079302064509039DC20792C1036

T002073073820644C000005

E001000

Programa en Memoria

0FF0	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx
1000	14103348	20390010	36281030	30101548
1010	20613C10	0300102A	0C103900	102D0C10
1020	36482061	0810334C	0000454F	46000003
1030	000000xx	xxxxxxxx	xxxxxxxx	xxxxxxxx
2030	xxxxxxxx	xxxxxxxx	xx041030	001030E0
2040	205D3020	3FD8205D	28103030	20576490
2050	392C205E	38203F10	10364C00	00F10010
2060	00041030	E0207930	20645090	39DC2079
2070	2C103638	20644C00	0005xxxx	xxxxxxxx
2080	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

Características Dependientes de la Máquina

Relocalización

- El cargador absoluto a pesar de que es eficiente y sencillo presenta ciertos inconvenientes
- El principal es la necesidad del programador de especificar la dirección en la que el programa será cargado en memoria
- Este problema se soluciona con el uso de una relocalización
- Al igual que en los ensambladores, la manera en que se implementa la relocalización en un cargador depende de las características de la maquina

Cargador Relativo

- Los cargadores que permiten la relocalización son llamados cargadores relocalizadores o cargadores relativos

Proceso de Cargardo con Relocalización (Registros M)

- Se coloca la información de los Registros de Texto en la dirección disponible
- El hacer esto dependerá del valor real en donde se coloque el programa, no necesariamente el de la dirección indicada
- Tras colocar la información de los registros de Texto, se e sumará a las direcciones indicadas en los registros de Modificación el valor real en el que se colocó el programa

Proceso de Cargado con Relocalización (Máscaras de Bits)

- Se coloca un Registro de Texto en la dirección disponible
- El hacer esto dependerá del valor real en donde se coloque el programa, no necesariamente el de la dirección indicada
- Tras colocarlo, se le sumará la dirección real a cada instrucción a la que le corresponda un 1 en su respectivo bit de la Máscara de Bits del registro de Texto

Relocalización por Hardware

- Existen computadoras que proporcionan una capacidad de relocalización mediante hardware que elimina algunas de las necesidades del cargador de realizar una relocalización

- Por ejemplo, algunas maquinas consideran todas las referencias de memoria como relativas al comienzo del área de memoria asignada al usuario

Características Independientes de la Máquina

- La mayoría de los cargadores pueden incorporar de manera automática rutinas de un subprograma que se encuentre en una biblioteca
- En la mayora de los casos existe una biblioteca estándar que se utiliza para éste fin
- Sin embargo otras bibliotecas deben ser especificadas por sentencias de control o por parámetros al cargador
- Esta propiedad permite al programador el uso de subrutinas de una o mas bibliotecas

Búsqueda Automática de Bibliotecas

- Es una propiedad que permite que las subrutinas llamadas por el programa que esta siendo cargado, sean automáticamente extraídas de la librera en la que se encuentran, cargadas y ligadas con el programa principal solo con mencionar sus nombres

Búsqueda de Bibliotecas

- Las bibliotecas a ser buscadas por el cargador normalmente contienen versiones ensambladas o compiladas de las subrutinas
- Es posible buscar estas bibliotecas revisando los registros de Definición para todos los códigos objeto de la biblioteca, sin embargo esto resultara poco eficiente
- Debido a esto, se utiliza una estructura de directorios las subrutinas, el cual contiene el nombre da cada rutina y un apuntador hacia la dirección en el código objeto en el que se encuentra
- De esta manera la búsqueda de una subrutina se reduce a buscarla en el directorio de rutinas y de a partir de ah, ir a la dirección en la que se encuentra

Opciones del Cargador

- Muchos cargadores permiten al usuario especificar opciones que modifiquen el funcionamiento del cargador

- Lenguaje de comandos
  - Un archivo de entrada que contenga estas secuencias de control
  - Inclusión de las declaraciones en el código fuente que será ensamblado

Lenguaje de Comandos

- INCLUDE
  - Permite la selección de fuentes alternativas de entrada

INCLUDE nombre (biblioteca)

  - Indica al cargador el leer programa objeto nombre que se encuentra en biblioteca y tratarlo como si fuera parte de la entrada primaria del cargador

Lenguaje de Comandos

- DELETE
  - Permite eliminar símbolos externos o secciones de control enteras

DELETE nombre

  - Indica al cargador que elimine la sección de control o el símbolo nombre

Lenguaje de Comandos

- CHANGE
  - Permite cambiar las referencias externas dentro de los programas que están siendo cargados y ligados

CHANGE nombre1, nombre2

  - Indica que el símbolo externo nombre1 sea cambiado a nombre2 en cualquier lugar que aparezca en los programas objeto

Lenguaje de Comandos

- LIBRARY
  - Otra opción común del cargador involucra la inclusión de rutinas de bibliotecas para satisfacer referencias externas

LIBRARY nombre

  - Le indica al cargador que incluya la biblioteca nombre para realizar la búsqueda de subrutinas

Lenguaje de Comandos

- NOCALL
  - Permite especificar que no se carguen ciertas bibliotecas

    NOCALL nombre1,nombre2,nombre3

  - Le indica al cargador que las referencias externas nombre1,nombre2,nombre3 permanecerán sin resolver, esto evita que se lleguen a cargar y ligar rutinas que no sean necesarias

Ejemplo

- Consideraciones:
  - Se tiene un programa principal llamado COPY que utiliza dos sub-programas, RDREC y WRREC, cada uno de estos es una sección de control separada
  - Suponer que un conjunto de subrutinas esta disponible en el sistema mediante la biblioteca UTLIB. Dos de estas, READ y WRITE, se diseñan para realizar las mismas funciones que RDREC y WRREC
  - Escribir las instrucciones para que el programa COPY utilice estas rutinas

Solución

- Comandos:
  - INCLUDE READ(UTLIB)
  - INCLUDE WRITE(UTLIB)
  - DELETE RDREC,WRREC
  - CHANGE RDREC,READ
  - CHANGE WRREC, WRITE

Cargador de Arranque

Cargador de Arranque

- Una de las principales preguntas es ¿Cómo se carga el cargador en memoria ?
- Una posibilidad es que sea el sistema operativo quien lo carga
- Si es as, ¿Quién carga al sistema operativo ?

Cargador en Memoria ROM

- En algunas computadoras, se tiene un cargador absoluto que reside de manera permanente en una memoria de solo lectura (ROM)
- Cuando el cargador recibe una señal del hardware, la máquina comienza a ejecutar este programa
- En algunos sistemas este programa es ejecutado en la memoria ROM
- En otras es copiado a la memoria principal para ser ejecutado

Carga vía Hardware

- El problema con el esquema anterior es cuando se tienen equipos que no tienen una memoria de solo lectura
- Una solución a esto es tener una función basada en hardware o una cantidad muy pequeña de memoria ROM, que lea un registro de longitud fija de un dispositivo en memoria en una posición fija
- Después de que la operación de lectura está completa, el control es automáticamente transferido a la dirección en memoria donde el registro fue almacenado
- Este registro contiene instrucciones máquina que cargan el programa