

# Programación de Sistemas

## Unidad 5. Ligador

# Contenido

# Introducción

- Cuando en un programa existen varias secciones de control, estas pueden ser ensambladas juntas o por separado
- En cualquier caso, se obtendrá un código objeto por cada una de las secciones
- El cargador no necesita ni tiene manera de saber que secciones de control fueron ensambladas al mismo tiempo
- El cargador solo ve secciones de control que deben ser ligadas, relocalizadas y cargadas

# Consideraciones Especiales

- En ocasiones es necesario añadir a la instrucción la localidad de memoria en donde inicia la memoria
- En aquellas instrucciones donde se tiene solo un símbolo local con referencias externas, se debe añadir o restar (dependiendo de la instrucción) el valor de la localidad de memoria en donde el programa comienza

# Consideraciones

- Se tienen tres programas ensamblados de manera separada, cada uno con una sola sección de control
- Cada programa contiene una lista de elementos LISTA, LISTB, LISTC; el final de estas listas esta indicado por las etiquetas ENDA, ENDB y ENDC
- Cada programa contiene exactamente el mismo el mismo conjunto de referencias a estos símbolos externos
- Tres de estos son operandos de instrucción REF1 a REF3 y el resto son valores para las palabras REF4 a REF8

# Programa A

```
0000      PROGA      START      0
                        EXTDEF    LISTA,ENDA
                        EXTREF    LISTB,ENDB,LISTC,ENDC
                        .
                        .
                        .
0020      REF1      LDA         LISTA          00201D
0023      REF2      +LDT       LISTB+4       74100004
0027      REF3      LDX        #ENDA-LISTA   050014
                        .
                        .
                        .
0040      LISTA     EQU        *
                        .
                        .
0054      ENDA     EQU        *
0054      REF4     WORD       ENDA-LISTA+LISTC 000014
0057      REF5     WORD       ENDC-LISTC-10   FFFFF6
005A      REF6     WORD       ENDC-LISTC+LISTA-1 00003F
005D      REF7     WORD       ENDA-LISTA-(ENDB-LISTB) 000014
0060      REF8     WORD       LISTB-LISTA    FFFFC0
                        END      REF1
```

# Registros de Modificación del Programa A

- REF1:El operando LISTA se calcula directamente y vale 1D (por el direccionamiento relativo a PC)
- REF2:Se coloca el valor de 4 y se genera un registro de modificación para la referencia externa LISTB
  - M00002405+LISTB
- REF3:Se puede calcular de manera directa y vale 14
- REF4:Se puede calcular de manera directa la operación ENDA-LISTA, pero se debe generar un registro de modificación para la referencia externa LISTC
  - M00005406+LISTC

# Programa B

```
0000      PROGB      START      0
                        EXTDEF    LISTB,ENDB
                        EXTREF    LISTA,ENDA,LISTC,ENDC
                        .
                        .
                        .
0036      REF1      +LDA      LISTA      00100000
003A      REF2      LDT       LISTB+4    742027
003D      REF3      +LDX      #ENDA-LISTA 05100000
                        .
                        .
                        .
0060      LISTB     EQU       *
                        .
                        .
0070      ENDB     EQU       *
0070      REF4     WORD      ENDA-LISTA+LISTC 000000
0073      REF5     WORD      ENDC-LISTC-10  FFFFFFF6
0076      REF6     WORD      ENDC-LISTC+LISTA-1  FFFFFFFF
0079      REF7     WORD      ENDA-LISTA-(ENDB-LISTB)  FFFFF0
007C      REF8     WORD      LISTB-LISTA  000060
                        END
```

# Registros de Modificación del Programa B

- REF1:El operando LISTA necesita un registro de modificación por ser una referencia externa
  - M00003705+LISTA
- REF2:Se puede calcular el valor directamente y es 27 (por el ajuste del direccionamiento relativo a PC)
- REF3:Se coloca la dirección en 0 y se requieren dos registros de modificación para las referencias externas
  - M00003E05+ENDA
  - M00003E05-LISTA
- REF4:Se coloca la dirección en 0, y se requieren tres registros de modificación para las referencias externas
  - M00007006+ENDA
  - M00007006-LISTA
  - M00007006-LISTC

# Programa C

```
0000      PROGC      START      0
          EXTDEF    LISTC,ENDC
          EXTREF    LISTA,ENDA,LISTB,ENDB
          .
          .
          .
0018      REF1      +LDA      LISTA      00100000
001C      REF2      +LDT      LISTB+4    74100004
0020      REF3      +LDX      #ENDA-LISTA 05100000
          .
          .
          .
0030      LISTC     EQU      *
          .
          .
0042      ENDC     EQU      *
0042      REF4      WORD     ENDA-LISTA+LISTC 000030
0045      REF5      WORD     ENDC-LISTC-10 000008
0048      REF6      WORD     ENDC-LISTC+LISTA-1 000011
004B      REF7      WORD     ENDA-LISTA-(ENDB-LISTB) 000000
004E      REF8      WORD     LISTB-LISTA 000000
          END
```

# Registros de Modificación del Programa C

- REF1:El operando LISTA necesita un registro de modificación por ser una referencia externa
  - M00001905+LISTA
- REF2:Se coloca el valor de 4 y se genera un registro de modificación para la referencia externa LISTB
  - M00001D05+LISTB
- REF3:Se coloca la dirección en 0 y se requieren dos registros de modificación para las referencias externas
  - M00002105+ENDA
  - M00002105-LISTA
- REF4:Se coloca el valor de LISTC, pero se deben generar registros de modificación para las referencias externas de la operación ENDA-LISTA
  - M00004206+ENDA
  - M00004206-LISTA
  - M00004206+PROGC

# Referencias Adelantadas

- Es posible y muy común que una sección de control haga una referencia a un símbolo externo cuya definición no aparecerá sino hasta mas adelante
- En tal caso la operación de ligado no puede ser realizada hasta que se conozca la dirección asignada al símbolo externo (es decir, hasta que se lea la correspondiente sección de control)
- Debido a esto, al igual que ocurre con el ensamblador, el cargador-ligador realiza dos pasadas sobre la entrada

# Tabla de Símbolos Externos (ESTAB)

- Es análoga a la tabla de símbolos (SYMTAB) en el algoritmo del ensamblador
- Es utilizada para almacenar el nombre y dirección de cada símbolo externo en el conjunto de secciones de control que esta siendo cargado
- También indica en que sección de control esta definido el símbolo
- La organización de la tabla normalmente se basa en una tabla hash

# Estructuras Auxiliares

- PROGADDR
- Dirección de inicio en memoria donde el programa ligado será cargado. Su valor es proporcionado al cargador por el sistema operativo
- CSADDR
- Dirección de inicio asignada a la sección de control que esta siendo revisada por el cargador

# Primer Paso del Ligado

- Durante el primer paso el cargador se enfoca solamente en los registros de Encabezado (H) y de Definición (D) en las secciones de control
- La dirección inicial de carga (PROGADDR) es obtenida del sistema operativo
- Esta se convierte en la dirección de inicio (CSADDR) para la primera sección de control

# Generación de la Tabla de Símbolos Externos

- El nombre de la sección de control del registro de Encabezado se introduce en la tabla de símbolos externos (ESTAB) con un valor dado por CSADDR
- Todos los símbolos externos que aparecen en el registro de Definición de la sección de control son también introducidos en la tabla
- Sus direcciones son obtenidas añadiendo el valor especificado en el registro de Definición al valor de CSADDR
- Cuando se encuentra el registro de Fin (E), la longitud de sección de control (CSLTH) que se obtuvo del registro de encabezado se añade al valor de SADDR
- Esta operación proporciona la dirección de inicio para la siguiente sección de control

# Primer Paso del Ligador

1. Obtener PROGADDR del sistema operativo
2. Hacer CSADDR = PROGADDR
3. MIENTRAS no se llegue al final del archivo
  1. Leer el registro de entrada (Encabezado)
  2. Hacer CSLTH = Longitud de la sección
  3. Buscar en ESTAB el nombre de la sección
  4. SI encontrado, ERROR (Símbolo duplicado)
  5. OTRO, Insertar el símbolo en ESTAB con el valor de CSADDR
  6. MIENTRAS el tipo de registro no sea E
    1. Leer siguiente registro
    2. SI el tipo de registro es D
    3. PARA cada símbolo en el registro
    4. Busca en ESTAB el símbolo
    5. SI encontrado, ERROR (Símbolo duplicado)
    6. OTRO, Insertar el símbolo en ESTAB con el valor de CSADDR + dirección especificada
  7. Suma CSLTH a CSADDR (obtener la dirección de la siguiente sección)

# Ejemplo

- Dados los siguientes tres códigos objeto, generar la ESTAB
- Suponer que el programa se carga en la dirección 4000H

# Programa A

HPROGA 000000000063

DLISTA 000040ENDA 000054

RLISTB ENDB LISTC ENDC

T0000200A00201D74100004050014

T0000540F000014FFFFFF600003F000014FFFFC0

M00002405+LISTB

M00005406+LISTC

M00005706+ENDC

M00005706-LISTC

M00005A06+ENDC

M00005A06-LISTC

M00005D06-ENDB

M00005D06+LISTB

E000020

# Programa B

HPROGB 00000000007F

DLISTB 000060ENDB 000070

RLISTA ENDA LISTC ENDC

T0000360B0010000074202705100000

T0000700F000000FFFFFF6FFFFFFFFFFFF0000060

M00003705+LISTA

M00003E05+ENDA

M00003E05-LISTA

M00007006+ENDA

M00007006-LISTA

M00007006+LISTC

M00007306+ENDC

M00007306-LISTC

M00007606+ENDC

M00007606-LISTC

M00007606+LISTA

M00007906+ENDA

M00007906-LISTA

M00007C06-LISTA

# Programa C

HPROGC 000000000051

DLISTC 000030ENDC 000042

RLISTA ENDA LISTB ENDB

T0000180C001000007410000405100000

T0000420F000030000008000011000000000000

M00001905+LISTA

M00001D05+LISTB

M00002105+ENDA

M00002105-LISTA

M00004206+ENDA

M00004206-LISTA

M00004806+LISTA

M00004B06+ENDA

M00004B06-LISTA

M00004B06-ENDB

M00004B06+LISTB

M00004E06+LISTB

M00004E06-LISTA

E

# ESTAB Generada

Sección de control	Símbolo	Dirección	Longitud
PROGA		4000	0063
	LISTA	4040	
	ENDA	4054	
PROGB		4063	007F
	LISTB	40C3	
	ENDB	40D3	
PROGC		40E2	0051
	LISTC	4112	
	ENDC	4124	

# Segundo Paso

- El segundo paso del cargador realiza las operaciones de carga, relocalización y ligado del programa
- CSADDR siempre contiene la dirección de inicio de la sección de control que esta siendo cargada
- Otro auxiliar es EXECADDR (dirección de ejecución) que contiene la dirección de la primera instrucción que se ejecuta en el programa

# Dirección de Inicio de Ejecución

- El registro de Fin para cada sección de control contiene la dirección de inicio de la primer instrucción a ser ejecutada en la sección
- Si mas de un código tiene esta dirección el cargador usa de manera arbitraria la ultima que encuentra
- En caso de que no se especifique ninguna se utiliza la dirección de inicio del programa ligado

# Ejemplo

- Obtener los códigos objeto que se colocaran en memoria:

T0000200A00201D74100004050014

T0000540F000014FFFFFF600003F000014FFFFC0

M00002405+LISTB

M00005406+LISTC

M00005706+ENDC

M00005706-LISTC

M00005A06+ENDC

M00005A06-LISTC

M00005D06-ENDB

M00005D06+LISTB

# Solución

- Hay que utilizar la ESTAB para obtener el valor de los símbolos
- Sumar/restar al contenido de la dirección especificada en el campo de dirección del valor de modificación el valor del símbolo
- Repetir estos pasos para cada registro de modificación
- Al analizar se obtendrá el valor final de cada registro de Texto que será el que se distribuya en la memoria

# Efecto de los Registros de Modificación

- Considerar:

T 000020 0A 03201D 77100004

- Se revisara el efecto del registro de Modificación

M 000024 05 +LISTB

- La dirección de memoria 000024 contiene 100004
- El registro M indica que se debe añadir LISTB
- El valor de LISTB de acuerdo a ESTAB es 40C3

$$100004 + 40C3 = 1040C7$$

- Por lo tanto el bloque 77100004 se convierte en 771040C7

# Opciones de Diseño

# Editor de Ligado

- Dado un programa ensamblado o compilado, un editor de ligado produce una versión ligada del programa
- Esta versión es conocida como modulo de carga o imagen ejecutable
- Esta versión es usualmente escrita a un archivo o a una librería para su posterior ejecución

# Funcionamiento

1. El editor de ligado realiza la relocalización de todas las secciones de control relativas al comienzo del programa ligado
2. Todas las instrucciones que necesitan ser modificadas tienen valores que son relativos al inicio del ligado del programa
3. Cuando el programa va a ser ejecutado, un simple cargador-relocalizador puede ser utilizado para cargar el programa en memoria
4. La única modificación a realizar al código objeto es añadirle la dirección de carga actual a los programas relativos dentro del programa

# Ventajas

- Si un programa será ejecutado muchas veces sin tener que ser re-ensamblado, el uso de un editor de ligado reduce de manera sustancial el trabajo requerido
- Un programa ligado generado por un editor de ligado se encuentra en una forma adecuada para ser procesada por un cargador-relocalizador
- El editor de ligado puede reemplazar una subrutina modificada en la versión ligada del programa sin tener que realizar otras acciones con aquellas que no sufrieron modificación

# Ligado Dinámico

- Existe un esquema que pospone la función de ligado hasta el momento de la ejecución
- Una subrutina es cargada y ligada al resto del programa cuando es llamada por primera vez
- Este esquema es llamado ligado dinámico, cargado dinámico o carga en la llamada

# Funcionamiento

1. Se solicita una subrutina, esto se realiza utilizando como parámetro su nombre simbólico
2. El sistema operativo examina sus tablas internas para determinar si la rutina ha sido cargada o no. En caso de ser necesario, la rutina es cargada del sistema de librerías indicado
3. El control es entonces pasado del sistema operativo a la rutina que está siendo llamada
4. Cuando la subrutina completa su procesamiento, el sistema operativo regresa el control al programa que hizo la llamada
5. La memoria reservada para la carga de la subrutina no se libera de manera inmediata en caso que se requirieran otras llamadas

# Ventajas

- El ligado dinámico proporciona la habilidad de cargar las rutinas solo cuando son necesarias
- Si las subrutinas involucradas son grandes o tienen demasiadas referencias externas, el uso del ligado dinámico puede ahorrar grandes cantidades de tiempo y espacio en memoria
- Evita la necesidad de cargar la librería entera para cada ejecución
- Puede hacer innecesario para el programa el tener que conocer el conjunto de instrucciones completo del que se dispone