

# Sistemas Operacionais

## Montadores, ligadores e carregadores



2ª edição

Revisão: Fev/2003

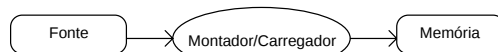
Anexo A

## Visão geral de montadores, ligadores e carregadores

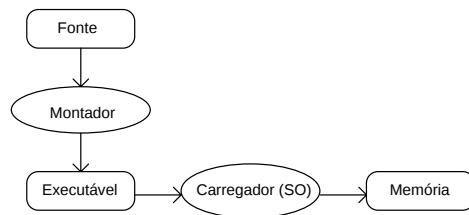
### ■ Como um programa é construído até ser executado?

- Montadores (*assemblers*): responsáveis por traduzir programas escritos em linguagem de montagem (*assembly*) para linguagem de máquina
  - Compiladores: traduzem programa escrito em linguagem de programação
- Ligadores (*linkers*): une diferentes partes de um mesmo programa
  - Permite desenvolvimento de forma modular
- Carregadores (*loaders*): transfere um programa para a memória principal para que seja executado

## Montador, ligador e carregador (1)

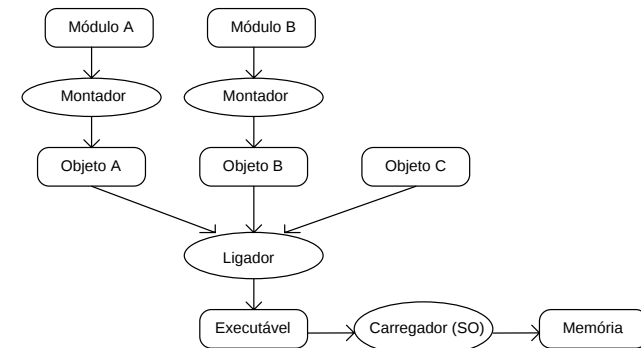


Arquitetura 1 - Montador e carregador integrados



Arquitetura 2 - Montador e carregador independentes

## Montador, ligador e carregador (2)



Arquitetura 3 - Montador, ligador e carregador independentes

## Descrição de uma máquina hipotética

### Arquitetura:

- 1 acumulador (acc) de 16 bits
- 1 apontador de instruções (*program counter*) de 16 bits
- Memória de  $2^{16}$  palavras de 16 bits
- 3 formatos de instruções

|            |                               |
|------------|-------------------------------|
| Formato 1: | Opcode                        |
| Formato 2: | Opcode                        |
|            | Endereço do operando          |
| Formato 3: | Opcode                        |
|            | Endereço do primeiro operando |
|            | Endereço do segundo operando  |

## Conjunto de instruções da máquina hipotética

| Opcode Simbólico | Opcode Numérico | Tamanho | Número de Operandos | Ação                                  |
|------------------|-----------------|---------|---------------------|---------------------------------------|
| ADD              | 01              | 2       | 1                   | $ACC \leftarrow ACC + mem(OP)$        |
| SUB              | 02              | 2       | 1                   | $ACC \leftarrow ACC - mem(OP)$        |
| MUL              | 03              | 2       | 1                   | $ACC \leftarrow ACC \times mem(OP)$   |
| DIV              | 04              | 2       | 1                   | $ACC \leftarrow ACC \div mem(OP)$     |
| JMP              | 05              | 2       | 1                   | $PC \leftarrow OP$                    |
| JMPN             | 06              | 2       | 1                   | se $ACC < 0$ então $PC \leftarrow OP$ |
| JMPP             | 07              | 2       | 1                   | se $ACC > 0$ então $PC \leftarrow OP$ |
| JMPZ             | 08              | 2       | 1                   | se $ACC = 0$ então $PC \leftarrow OP$ |
| COPY             | 09              | 3       | 2                   | $mem(OP2) \leftarrow mem(OP1)$        |
| LOAD             | 10              | 2       | 1                   | $ACC \leftarrow mem(OP)$              |
| STORE            | 11              | 2       | 1                   | $mem(OP) \leftarrow ACC$              |
| INPUT            | 12              | 2       | 1                   | $mem(OP) \leftarrow entrada$          |
| OUTPUT           | 13              | 2       | 1                   | $saída \leftarrow mem(OP)$            |
| STOP             | 14              | 1       | 0                   | Suspende execução                     |

## Montador (*assembler*)

- Responsável por traduzir programas escritos em linguagem de montagem (*assembly*) para linguagem de máquina

| End. | Opcode | Oper |           |
|------|--------|------|-----------|
| 0    | 12     | 13   | INPUT N1  |
| 2    | 12     | 14   | INPUT N2  |
| 4    | 10     | 13   | LOAD N1   |
| 6    | 01     | 14   | ADD N2    |
| 8    | 11     | 15   | STORE N3  |
| 10   | 13     | 15   | OUTPUT N3 |
| 12   | 14     |      | STOP      |
| 13   | ??     | N1:  | SPACE     |
| 14   | ??     | N2:  | SPACE     |
| 15   | ??     | N3:  | SPACE     |

## Conceitos básicos

- Sintaxe típica de programas de montagem
  - [rótulo:] [operação] [operando1][operando2] [;comentários]
- Rótulo:
  - Símbolo empregado para representar uma posição de memória
- Operação:
  - Opcodes simbólicos: representam instruções
  - Pseudo-instruções(diretivas): operações que orientam a atuação do montador
- Operandos:
  - Posições de memória, registradores, constantes, etc...
- Comentário

## Funções básicas de um montador

- Substituir opcodes simbólicos por opcodes numéricos
- Substituir endereços simbólicos por endereços numéricos
- Reservar espaço para dados
- Gerar constantes em memória

## Diretivas básicas de montadores

- Demarcação de início/fim do programa fonte
  - ┆ begin/end
- Reserva de espaço para variáveis
  - ┆ e.g.: space, db, dw, dd
- Geração de constantes
  - ┆ e.g.: const
- Contador de posições
  - ┆ e.g.: org, origin
- Definições de sinônimos
  - ┆ e.g.: equ, set

## Algoritmo clássico de duas passagens

- Realiza duas leituras do arquivo fonte
- Primeira passagem:
  - ┆ Reconhece todos os símbolos definidos pelo programador
    - Armazena símbolos e o respectivo valor em uma tabela
- Segunda passagem:
  - ┆ Geração do código objeto
- Estruturas de dados empregadas:
  - ┆ Tabela de instruções
  - ┆ Tabela de diretivas
  - ┆ Tabela de símbolos
  - ┆ Contador de posições

## Primeira passagem

```
00 09 ?? ??    COPY    ZERO, OLDE
03 09 ?? ??    COPY    ONE, OLD
06 12 ??       INPUT    LIMIT
08 13 ??       OUTPUT   OLD
10 10 ??       FRONT:  LOAD  OLDER
12 01 ??       ADD      OLD
14 11 ??       STORE    NEW
16 02 ??       SUB      LIMIT
18 07 ??       JMPP     FINAL
20 13 ??       OUTPUT   NEW
22 09 ?? ??    COPY    OLD, OLDER
25 09 ?? ??    COPY    NEW, OLD
28 05 10       JMP      FRONT
30 13 ??       FINAL:  OUTPUT  LIMIT
32 14          STOP
33 00          ZERO:   CONST   0
34 01          ONE:    CONST   1
35 ??          OLDER:  SPACE
36 ??          OLD:    SPACE
37 ??          NEW:    SPACE
38 ??          LIMIT:  SPACE
```

Tabela de Símbolos

| <u>Símbolo</u> | <u>Valor</u> |
|----------------|--------------|
| FRONT          | 10           |
| FINAL          | 30           |
| ZERO           | 33           |
| ONE            | 34           |
| OLDER          | 35           |
| OLD            | 36           |
| NEW            | 37           |
| LIMIT          | 38           |

## Segunda passagem

|    |    |    |        |        |            |
|----|----|----|--------|--------|------------|
| 00 | 09 | 33 | 35     | COPY   | ZERO, OLDE |
| 03 | 09 | 34 | 36     | COPY   | ONE, OLD   |
| 06 | 12 | 38 |        | INPUT  | LIMIT      |
| 08 | 13 | 36 |        | OUTPUT | OLD        |
| 10 | 10 | 35 | FRONT: | LOAD   | OLDER      |
| 12 | 01 | 36 |        | ADD    | OLD        |
| 14 | 11 | 37 |        | STORE  | NEW        |
| 16 | 02 | 38 |        | SUB    | LIMIT      |
| 18 | 07 | 30 |        | JMPP   | FINAL      |
| 20 | 13 | 37 |        | OUTPUT | NEW        |
| 22 | 09 | 36 | 35     | COPY   | OLD, OLDER |
| 25 | 09 | 37 | 36     | COPY   | NEW, OLD   |
| 28 | 05 | 10 |        | JMP    | FRONT      |
| 30 | 13 | 38 | FINAL: | OUTPUT | LIMIT      |
| 32 | 14 |    |        | STOP   |            |
| 33 | 00 |    | ZERO:  | CONST  | 0          |
| 34 | 01 |    | ONE:   | CONST  | 1          |
| 35 | ?? |    | OLDER: | SPACE  |            |
| 36 | ?? |    | OLD:   | SPACE  |            |
| 37 | ?? |    | NEW:   | SPACE  |            |
| 38 | ?? |    | LIMIT: | SPACE  |            |

### Tabela de Símbolos

| Símbolo | Valor |
|---------|-------|
| FRONT   | 10    |
| FINAL   | 30    |
| ZERO    | 33    |
| ONE 34  |       |
| OLDER   | 35    |
| OLD 36  |       |
| NEW 37  |       |
| LIMIT   | 38    |

13

## Valor absoluto x relativo

- Valor absoluto são aqueles cujo valor final independem da posição onde o programa será carregado na memória
  - e.g.: opcodes, constantes
- Valor relativo são aqueles cujo valor final dependem da posição onde o programa será carregado na memória
  - e.g: labels, variáveis
- Necessidade de corrigir parte do código (valores relativos) para ficar coerente com o endereço de carga do programa
  - Procedimento de relocação

Sistemas Operacionais

14

## Exemplo valor absoluto versus relativo

|      |        |     | end. | Opcode | operando |
|------|--------|-----|------|--------|----------|
| TIP: | SET    | 6   | 0:   |        |          |
|      | INPUT  | N1  | 0:   | 12a    | 15r      |
|      | INPUT  | N2  | 2:   | 12a    | 16r      |
|      | LOAD   | N3  | 4:   | 10a    | 17r      |
|      | ADD    | N1  | 6:   | 01a    | 15r      |
|      | ADD    | N2  | 8:   | 01a    | 16r      |
|      | STORE  | N3  | 10:  | 11a    | 17r      |
|      | OUTPUT | N3  | 12:  | 13a    | 17r      |
|      | STOP   |     | 14:  | 14a    |          |
| N1:  | SPACE  | 1   | 15:  | ??a    |          |
| N2:  | SPACE  | 1   | 16:  | ??a    |          |
| N3:  | CONST  | TIP | 17:  | 06a    |          |

15

## Atributos dos símbolos

- Dois atributos básicos:
  - Nome e valor
- Outros atributos:
  - Valor absoluto ou valor relativo
  - Tamanho da variável

Sistemas Operacionais

16

## Tratamento de erros

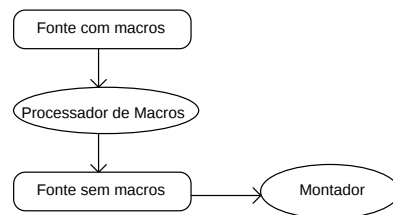
- Instrução não identificada
- Número inválido de operandos
- Tipo inválido de operandos
- Símbolo redefinido
- Símbolo indefinido

## Montador de uma passagem

- Objetivo é economizar uma segunda passagem
- Geração de uma lista de pendências para conter as referências a símbolos usados antes de serem definidos
  - Na tabela de símbolos, para uma referência ainda não definida, o valor da entrada é um ponteiro para uma lista de posições onde o símbolo é utilizado

## Macros

- Empregado para simplificar a construção de programas
- Construções semelhantes a subrotinas
  - Associação de nomes a trecho de códigos repetidos muitas vezes

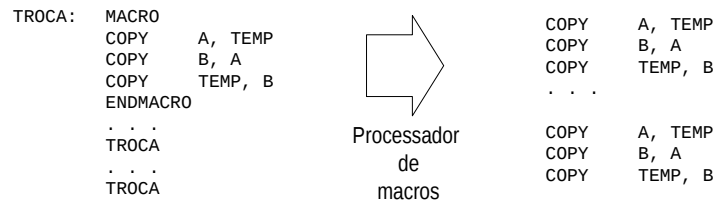


## Sub-rotinas versus macros

- Sub-rotinas
  - Código aparece na memória do computador apenas uma vez
  - Chamada desvia a execução do programa para a sub-rotina
  - Implementada por instruções de máquina
- Macros
  - Código aparece na memória do computador quantas vezes a macro for chamada
  - Chamada insere no programa o código da macro
  - Implementada pelo montador
- Ambas admitem parâmetros

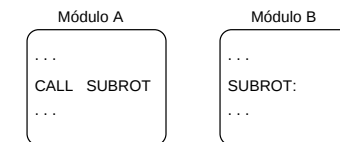
## Princípio de funcionamento

- Introdução de duas novas diretivas: MACRO e ENDMACRO
  - Delimitação do corpo da macro
- Chamada a macro corresponde a usar o seu nome como operação
- Operação de expansão



## Ligadores

- Tarefas básicas:
  - Juntar diversos arquivos objetos em um único arquivo executável
  - Resolver referências cruzadas
    - Quando o programador usa, em um módulo, um símbolo que foi definido em outro módulo

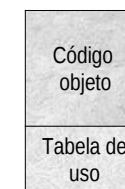


## Características do procedimento de montagem

- Módulos são montados de forma completamente independente
  - Módulos podem ser bibliotecas
    - Código fonte não conhecido
- Símbolos não conhecidos/definidos dentro de um módulo são denominados de referência cruzada ou referência não resolvida
  - Montador necessita “informar” essa situação
    - Tabela de uso

## Tabela de uso

- Inserida pelo montador no código objeto
- Indica quais símbolos externos ao módulo foram utilizados e onde, no código objeto, eles foram referenciados
- Ligador utiliza informações da *tabela de uso* para realizar as correções necessárias
- Formato do arquivo objeto (saída do montador):
  - Ligador deve “conhecer” o formato



## Problemas com o algoritmo básico de montagem

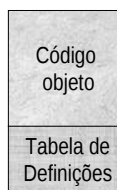
- No algoritmo original do montador, uma referência não resolvida (símbolo) é considerado erro de programação
- Agora existe a hipótese de que o símbolo é definido em outro módulo
- Como agir ?
  - Opção 1: considerar toda referência não resolvida como símbolo externo
    - Desvantagem: erro de montagem é detectado apenas na fase de ligação
  - Opção 2: incluir diretivas de montagem que obriguem o programador a declarar explicitamente que está utilizando um símbolo externo

## Diretiva EXTERN

- Programador informa quais referências (símbolos) empregadas são definidas em outros módulos
  - Diferenciação entre erro “símbolo não-definido” e “símbolo não-conhecido” no momento da montagem
- Símbolos declarados como EXTERN fazem parte da *tabela de uso*
- Noção de símbolos locais e símbolos globais
  - Símbolo é local ao módulo ao qual foi definido
  - Símbolo é global quando for definido em um módulo e utilizado por outro
    - Necessário indicar isso (outra diretiva: PUBLIC)

## Diretiva PUBLIC

- Programador informa quais símbolos serão exportados por um módulo, podendo ser referenciados por outros módulos
- Símbolo é definido localmente porém tem um escopo global
- Símbolos PUBLIC formam a *tabela de definições*
  - Sub-conjunto da tabela de símbolos
  - Incluída no arquivo objeto



## Resumo

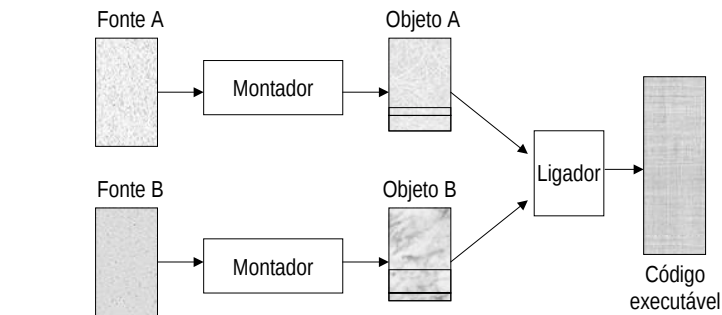
- Diretiva EXTERN permite informar quais símbolos são definidos externamente (importação)
- Diretiva PUBLIC indica que um símbolo pode ser utilizado em outros módulos (exportação)
- Tabela de uso mantém quais símbolos externos foram empregados em um módulo e onde, no código, foram utilizados
- Tabela de definições reúne todos os símbolos (e respectivos atributos) tornados públicos

## Alterações no algoritmo básico do montador

- Tabela de símbolos deve informar se símbolo é local ou externo
- Incluir tratamento da diretiva EXTERN e PUBLIC
  - Primeira passagem insere respectivo rótulo na tabela de uso (extern)
  - Primeira passagem insere operando na tabela de definições (public)
- Tabela de uso e tabela de definições devem, após a montagem, ser incluídas no arquivo objeto gerado
  - Atributos da tabela de símbolo são copiados para a tabela de definições
- Tratamento de erros:
  - Redefinição de símbolos (locais e globais devem ter nomes distintos)
  - Símbolo exportado mas não definido
  - Símbolo externo foi exportado

## Exemplo de geração de módulos

- Supor um programa que é composto por dois módulos A e B



## Módulo A

|                    |                                                                               |    |   |   |
|--------------------|-------------------------------------------------------------------------------|----|---|---|
| MOD_A: BEGIN       | Tabela de Símbolos: <u>Símbolo</u> <u>Valor</u> <u>Rel/Abs</u> <u>Externo</u> |    |   |   |
| X: EXTERN          | MOD_A                                                                         | 0  | r | n |
| Y: EXTERN          | X                                                                             | 0  | a | s |
| MOD_B: EXTERN      | Y                                                                             | 0  | a | s |
| PUBLIC VAL         | MOD_B                                                                         | 0  | a | s |
| PUBLIC L1          | TAM                                                                           | 10 | a | n |
| TAM: EQU 10        | L1                                                                            | 10 | r | n |
| INPUT Y            | VAL                                                                           | 11 | r | n |
| LOAD VAL           | Tabela de Definições: <u>Símbolo</u> <u>Valor</u> <u>Rel/Abs</u>              |    |   |   |
| ADD Y              | VAL                                                                           | 11 | r |   |
| STORE Y + 2        | L1                                                                            | 10 | r |   |
| JMP MOD_B          | Tabela de Uso: <u>Símbolo</u> <u>End.</u> <u>Operação</u>                     |    |   |   |
| L1: STOP           | Y                                                                             | 1  | + |   |
| VAL: CONST TAM - X | Y                                                                             | 5  | + |   |
| END                | Y                                                                             | 7  | + |   |
|                    | MOD_B                                                                         | 9  | + |   |
|                    | X                                                                             | 11 | - |   |

Atenção: valores em decimal

## Módulo B

|               |                                                                               |    |   |   |
|---------------|-------------------------------------------------------------------------------|----|---|---|
| MOD_B: BEGIN  | Tabela de Símbolos: <u>Símbolo</u> <u>Valor</u> <u>Rel/Abs</u> <u>Externo</u> |    |   |   |
| PUBLIC X      | MOD_B                                                                         | 0  | r | n |
| PUBLIC Y      | VAL                                                                           | 0  | a | s |
| PUBLIC MOD_B  | L1                                                                            | 0  | a | s |
| VAL: EXTERN   | X                                                                             | 5  | a | n |
| L1: EXTERN    | Y                                                                             | 8  | r | n |
| X: EQU 5      | TAM                                                                           | 11 | r | n |
| OUTPUT Y      | Tabela de Definições: <u>Símbolo</u> <u>Valor</u> <u>Rel/Abs</u>              |    |   |   |
| OUTPUT VAL    | X                                                                             | 5  | a |   |
| OUTPUT TAM    | Y                                                                             | 8  | r |   |
| JMP L1        | MOD_B                                                                         | 0  | r |   |
| Y: SPACE 3    | Tabela de Uso: <u>Símbolo</u> <u>Endereço</u> <u>Operação</u>                 |    |   |   |
| TAM: CONST 10 | VAL                                                                           | 3  | + |   |
| END           | L1                                                                            | 7  | + |   |

Atenção: valores em decimal

## Algoritmo básico de ligação (estática)

- Efetua duas passagens sobre os arquivos objeto
- Primeira passagem:
  - Recolhe as informações contidas nas tabelas de definições e compõem uma tabela global de definições
- Segunda passagem:
  - Cópia cada código objeto para o arquivo executável
  - Consulta a tabela de uso de cada arquivo objeto e resolve a referência consultando a tabela global de definições
- Necessidade de realizar relocação
  - Módulos são montados considerando endereço de origem como zero

## Exemplo de ligação módulos A e B

|                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> MOD_A:  BEGIN X:      EXTERN Y:      EXTERN MOD_B:  PUBLIC  VAL         PUBLIC  L1 TAM:    EQU     10         INPUT   Y 12a 0a  LOAD   VAL 10a 11r  ADD    Y 01a 0a  STORE  Y + 2 11a 2a  JMP     MOD_B 07a 0a  L1:    STOP 14a    VAL:   CONST  TAM - X 10a    END             </pre> | <pre> MOD_B:  BEGIN         PUBLIC  X         PUBLIC  Y         PUBLIC  MOD_B VAL:    EXTERN L1:     EXTERN X:      EQU     5 13a 8r  OUTPUT  Y 13a 0r  OUTPUT  VAL 13a 11r  OUTPUT  TAM 05a 0a  JMP     L1 0a 0a 0a Y:  SPACE  3 10a    TAM:  CONST  10         END             </pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Tabela de Definições:

| Símbolo | Valor   | Rel/Abs |
|---------|---------|---------|
| VAL     | 11      | r       |
| L1      | 10      | r       |
| X       | 5       | a       |
| Y       | 8+12=20 | r       |
| MOD_B   | 0+12=12 | r       |

## Ligação relativa ao módulo B (apenas)

Tabela de Definições:

| Símbolo | Valor   | Rel/Abs |
|---------|---------|---------|
| VAL     | 11      | r       |
| L1      | 10      | r       |
| X       | 5       | a       |
| Y       | 8+12=20 | r       |
| MOD_B   | 0+12=12 | r       |

Tabela de Uso (B):

| Símbolo | Endereço (uso) | Operação |
|---------|----------------|----------|
| VAL     | 3 + 12 = 15    | +        |
| L1      | 7 + 12 = 19    | +        |

| <table border="1"> <thead> <tr> <th>End.</th> <th>Opcod</th> <th>Operando</th> </tr> </thead> <tbody> <tr><td>12</td><td>13a</td><td>8r + 12 = 20r</td></tr> <tr><td>14</td><td>13a</td><td>0a</td></tr> <tr><td>16</td><td>13a</td><td>11r + 12 = 23r</td></tr> <tr><td>18</td><td>05a</td><td>0a</td></tr> <tr><td>20</td><td>0a</td><td></td></tr> <tr><td>21</td><td>0a</td><td></td></tr> <tr><td>22</td><td>0a</td><td></td></tr> <tr><td>23</td><td>10a</td><td></td></tr> </tbody> </table> | End.  | Opcod          | Operando | 12 | 13a | 8r + 12 = 20r | 14 | 13a | 0a | 16 | 13a | 11r + 12 = 23r | 18 | 05a | 0a | 20 | 0a |  | 21 | 0a |  | 22 | 0a |  | 23 | 10a |  | <table border="1"> <thead> <tr> <th>End.</th> <th>Opcod</th> <th>Operando</th> </tr> </thead> <tbody> <tr><td>12</td><td>13a</td><td>20r</td></tr> <tr><td>14</td><td>13a</td><td>11r (0a+11r)</td></tr> <tr><td>16</td><td>13a</td><td>23r</td></tr> <tr><td>18</td><td>05a</td><td>10r (0a+10r)</td></tr> <tr><td>20</td><td>0a</td><td></td></tr> <tr><td>21</td><td>0a</td><td></td></tr> <tr><td>22</td><td>0a</td><td></td></tr> <tr><td>23</td><td>10a</td><td></td></tr> </tbody> </table> | End. | Opcod | Operando | 12 | 13a | 20r | 14 | 13a | 11r (0a+11r) | 16 | 13a | 23r | 18 | 05a | 10r (0a+10r) | 20 | 0a |  | 21 | 0a |  | 22 | 0a |  | 23 | 10a |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----------------|----------|----|-----|---------------|----|-----|----|----|-----|----------------|----|-----|----|----|----|--|----|----|--|----|----|--|----|-----|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------|----------|----|-----|-----|----|-----|--------------|----|-----|-----|----|-----|--------------|----|----|--|----|----|--|----|----|--|----|-----|--|
| End.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Opcod | Operando       |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 12                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 13a   | 8r + 12 = 20r  |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 13a   | 0a             |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 13a   | 11r + 12 = 23r |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 18                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 05a   | 0a             |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 20                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0a    |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 21                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0a    |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 22                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0a    |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 23                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 10a   |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| End.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Opcod | Operando       |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 12                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 13a   | 20r            |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 13a   | 11r (0a+11r)   |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 13a   | 23r            |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 18                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 05a   | 10r (0a+10r)   |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 20                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0a    |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 21                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0a    |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 22                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0a    |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |
| 23                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 10a   |                |          |    |     |               |    |     |    |    |     |                |    |     |    |    |    |  |    |    |  |    |    |  |    |     |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |       |          |    |     |     |    |     |              |    |     |     |    |     |              |    |    |  |    |    |  |    |    |  |    |     |  |

➔

## Arquivo executável (módulos A e B)

| End.  | Opcod | Operando |
|-------|-------|----------|
| 00    | 12a   | 20r      |
| 02    | 10a   | 11r      |
| 04    | 01a   | 20r      |
| 06    | 11a   | 22r      |
| 08    | 07a   | 12r      |
| 10    | 14a   |          |
| 11    | 5a    |          |
| <hr/> |       |          |
| 12    | 13a   | 20r      |
| 14    | 13a   | 11r      |
| 16    | 13a   | 23r      |
| 18    | 05a   | 10r      |
| 20    | 0a    |          |
| 21    | 0a    |          |
| 22    | 0a    |          |
| 23    | 10a   |          |

- Mapa de relocação somente é inserido no executável quando o carregador é relocador

## Ligação dinâmica (*load time*)

- Módulo aplicativo é carregado na memória
- Referência a um módulo externo faz com que o carregador busque o módulo “faltante” e realize a ligação
- Vantagens:
  - Flexibilidade para incorporar novas versões de módulos
    - Não necessita religar todos módulos para gerar um executável
  - Oferece mecanismo de base para compartilhamento de código
    - Vários executáveis não precisam incorporar um mesmo módulo
  - Facilita desenvolvimento de software por terceiros
    - e.g.: conceito de módulos no linux (comandos *insmod* e *rmmmod*)

## Carregador

- Módulo responsável por copiar um programa executável para a memória e prepará-lo para execução
- Principais tarefas:
  - Verificar se programa existe
  - Dimensionar a quantidade de memória necessária
  - Solicitar ao sistema operacional a memória necessária
  - Copiar código executável do arquivo para a memória
  - Ajustar os endereços do código executável para refletir a posição onde o programa foi carregado
- Tipicamente integrado ao sistema operacional

## Ligação dinâmica (*run time*)

- Ligação é executada durante a execução
  - *Dynamic Linking Libraries* (DLL)
    - Mecanismo baseado em *stubs*
- Referências externas permanecem não resolvidas no módulo já carregado na memória e em execução
- Chamadas a referências não resolvidas fazem o sistema operacional localizar o módulo “faltante”, carregá-lo e ligá-lo
- Vantagem:
  - Economia de memória
    - Mantém em memória porções realmente utilizadas
  - Flexibilidade para incorporar novas versões de módulos
    - Não necessita religar todos módulos para gerar um executável

## Tipos de carregador

- Carregador absoluto
- Carregador relocador
- Carregador dinâmico (*run time*)

## Carregador absoluto

- Considera que um dado módulo é carregado sempre na mesma posição
  - Todas referências são específicas a um endereço
- Referências são definidas pelo programador ou em tempo de compilação (montagem)
- Definidas pelo programador (desvantagens):
  - Programador deve conhecer previamente onde programa será carregado
  - Inclusão/remoção de instruções no módulo implica em redefinir endereços
- Solução: definir referências em tempo de compilação (montagem)
  - Uso de símbolos para representar referências

## Carregador relocador

- Código (carregador) absoluto impõem restrições
- Solução é gerar código com endereços relativos a um ponto bem definido
  - Início do programa
- Carga na posição  $x$  implica em adicionar  $x$  a cada referência de memória do módulo

## Carregador dinâmico (*run time*)

- Carregador relocador não é adequado quando existe *swapping*
  - Movimento de processos entre memória e disco para maximizar uso do processador (típico de sistemas multiprogramados)
  - Processos não retornam necessariamente na mesma posição que executavam anteriormente
    - Relocação já foi feita na primeira carga (Que fazer ?)
- Solução é executar a relocação no momento em que a posição for referenciada
  - Endereços devem ser na forma relativa ao início do módulo
  - Uso de hardware específico
    - e.g.: registradores de base

## Amarração de endereços em tempo de carga

- Tempo de programação
  - Todos os endereços são definidos diretamente pelo programador
- Tempo de compilação ou montagem
  - Endereços são definidos através de símbolos e resolvidos no momento da compilação/montagem
- Tempo de carga
  - Compilador/montador gera endereços relativos aos quais são resolvidos pelo carregador
- Tempo de execução
  - Programa contém endereços relativos os quais são convertidos em tempo de execução (dinamicamente) para end. absolutos por hardware específico

## Formas básicas de operação (1)

---

- Forma 1:
  - Não existe a fase de ligação
  - Endereço de carga é conhecido em tempo de montagem
  - Montador gera o código executável para um determinado endereço de carga
    - Não há necessidade de corrigir endereços
    - Carregador Absoluto
- Forma 2:
  - Não existe a fase de ligação
  - Endereço de carga não é conhecido em tempo de montagem
  - Montador gera código executável considerando endereço zero
    - Necessidade de corrigir endereços
    - Carregador Relocador

## Formas básicas de operação (2)

---

- Forma 3:
  - Existe a fase de ligação
  - Endereço de carga é conhecido em tempo de ligação
  - Ligador gera código executável para o endereço de carga
    - Não há necessidade de corrigir endereços
    - Carregador Absoluto
- Forma 4:
  - Existe a fase de ligação
  - Endereço de carga não é conhecido em tempo de montagem
  - Ligador gera código executável para o endereço zero
    - Necessidade de corrigir endereços
    - Carregador Relocador

## Características gerais de carregador

---

- Carregador independe da existência de ligador
- Tipo do carregador depende apenas do fato do endereço de carga ser conhecido ou não no momento da geração do executável
- Fortemente relacionado com gerência de memória do sistema operacional
  - e.g.: Sistema com paginação empregam carregadores absolutos

## Princípio básico carregador relocador

---

- Necessário corrigir o código
  - Valores absolutos: não necessitam de correção
    - e.g.: códigos de operação, constantes, etc...
  - Valores relativos: necessitam correção
    - e.g.: endereços
- Código gerado pelo montador (compilador) deve incluir informação de quais posições devem ser corrigidas

## Exemplo

```
PROG:  BEGIN
DEZ:    EQU          10
        INPUT        N
        ADD          INCR
        STORE        N+1
        ADD          INCR
        STORE        N+2
        OUTPUT       N+1
        OUTPUT       N+2
        STOP
N:      SPACE        3
INCR:   CONST        DEZ
        END
```

| <u>Símbolo</u> | <u>Valor</u> | <u>Rel./Ab</u> |
|----------------|--------------|----------------|
| PROG           | 0            | r              |
| DEZ            | 10           | a              |
| N              | 15           | r              |
| INCR           | 18           | r              |

Código executável: Carga endereço 200:

|     |     |    |     |
|-----|-----|----|-----|
| 12a | 15r | 12 | 215 |
| 01a | 18r | 01 | 218 |
| 11a | 16a | 11 | 216 |
| 01a | 18r | 01 | 218 |
| 11a | 17a | 11 | 217 |
| 13a | 16r | 13 | 216 |
| 13a | 17r | 13 | 217 |
| 14a |     | 14 |     |
| 00a |     | 00 |     |
| 00a |     | 00 |     |
| 00a |     | 00 |     |
| 10a |     | 10 |     |



## Formato de arquivos

- Imagem de memória
  - Contém apenas o código a ser executado
  - Carga consiste em copiar byte a byte o conteúdo do arquivo para a memória
- Código objeto/executável mais informações adicionais
  - Número mágico
    - Constante arbitrária que identifica o formato do arquivo
  - Mapa de relocação
    - bit-map, lista de endereços
  - Tamanho total do programa

## Leituras complementares

- R. Oliveira, A. Carissimi, S. Toscani; Sistemas Operacionais. Editora Sagra-Luzzato, 2001.
  - Anexo A