PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

PCS-2302 / PCS-2024
Lab. de Fundamentos de Eng. de Computação

Aula 03

Montador Relocável
Ligador e relocador

Professores:
Anarosa Alves Franco Brandão (PCS 2302)
Marcos A. Simplício Junior (PCS 2302/2024)
Ricardo Luis de Azevedo da Rocha (PCS2024)

Monitores: Felipe Leno e Michel Bieleveld

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Roteiro

1. Necessidade de programas relocáveis
2. Visão geral
3. Implicações na linguagem simbólica
   - Novas pseudo-instruções
   - Novo formato de instrução
4. Montador relocável
5. Ligador e Relocador
6. Importância de estruturação do código.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Necessidade de Programas Relocáveis (1)

- Programas absolutos são executáveis estritamente nas posições de memória em que foram criados.
- Tornam difícil a manutenção e o trabalho em equipe, pois:
  - Exigem gerência cuidadosa das áreas de memória ocupadas e dos endereços de cada parte do programa;
  - Toda vez que um programa é modificado, pode ser necessário recodificá-lo parcial ou totalmente;
  - Se a área ocupada pelo novo código for maior que a antiga, é preciso alojar o programa em outra parte da memória.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Necessidade de Programas Relocáveis (2)

- Programas relocáveis permitem sua execução em qualquer posição de memória:
  - As referências à memória devem ser previamente ajustadas;
  - Um gerenciador da ocupação da memória deve ser utilizado.
- Tornam possível utilizar partes de código projetadas externamente:
  - Uso de bibliotecas;
  - Exigem que se possa montar parcialmente um programa, sem todos os endereços resolvidos!

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

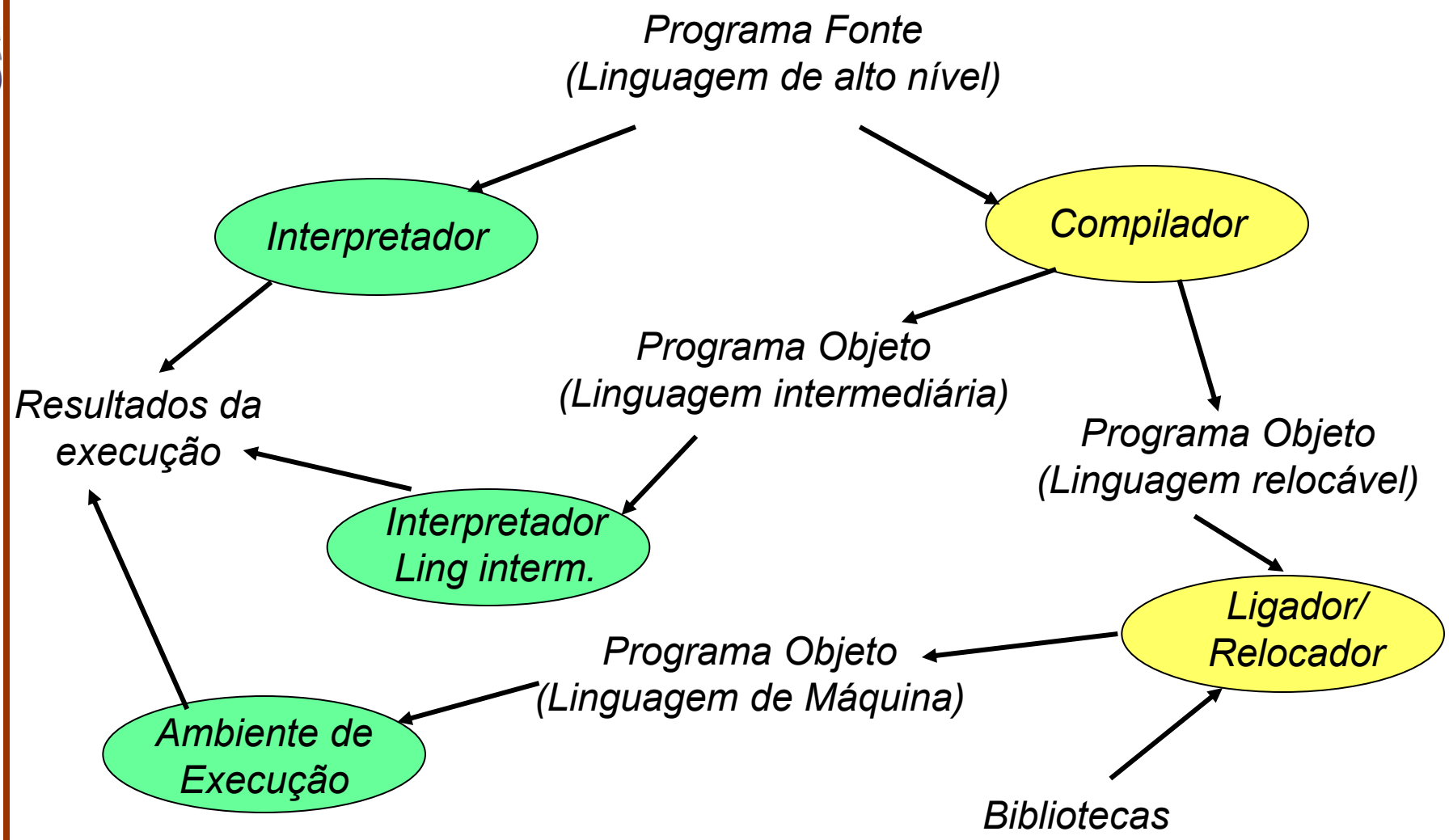
Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Visão geral

Processo de Execução de Linguagens de Alto Nível



Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

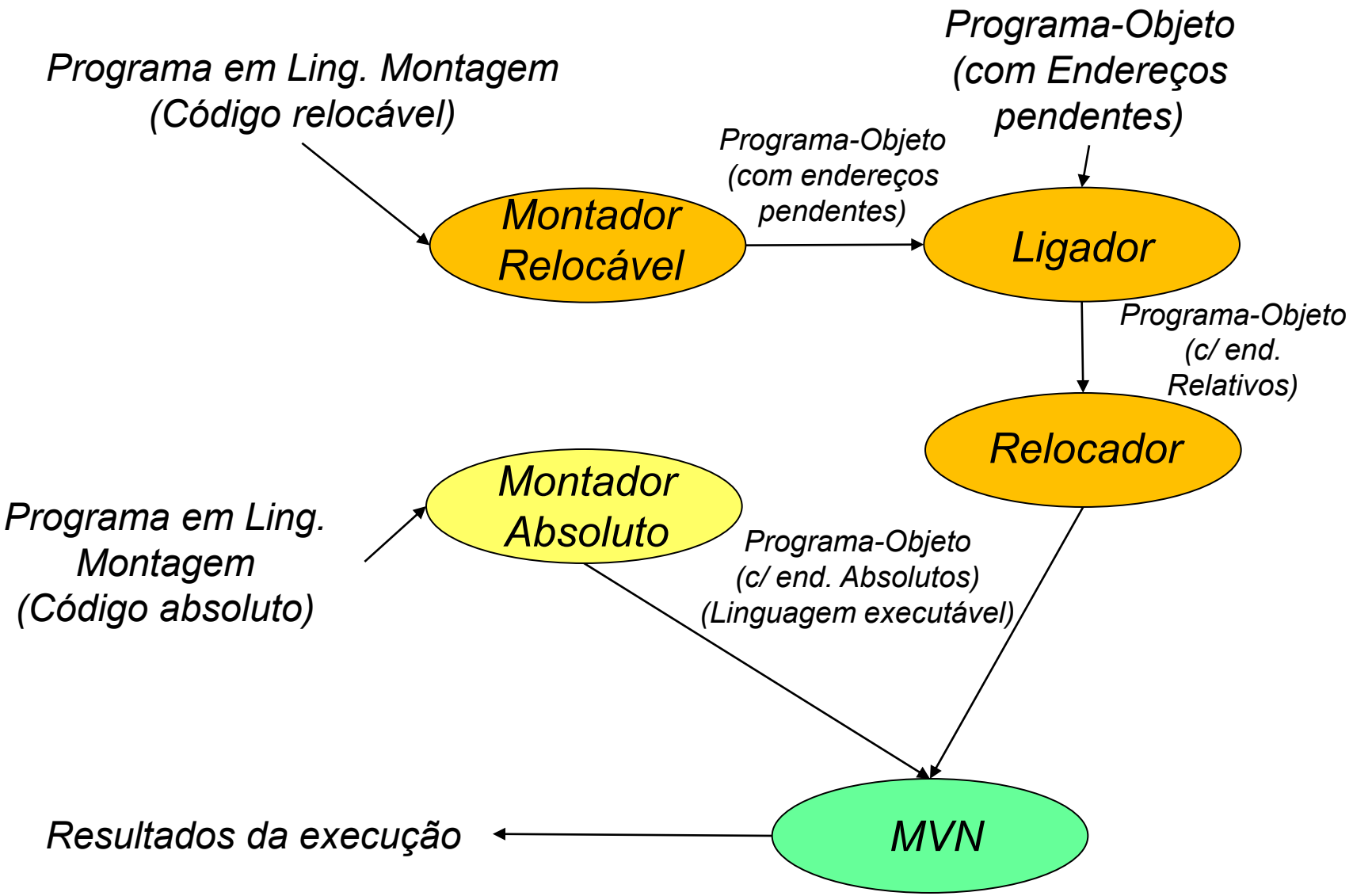
Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Visão Geral



PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Implicações na linguagem simbólica

- Para que se possa exprimir um programa relocável com possibilidade de construção em módulos, separadamente desenvolvidos, é necessário que:
  - Haja a possibilidade de representar e identificar endereços *absolutos* e endereços *relativos*;
  - Um programa possa ser montado sem que os seus endereços simbólicos estejam todos *resolvidos*;
  - Seja possível *identificar*, em um módulo, *símbolos* que possam ser *referenciados* simbolicamente em outros módulos.

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

Implicações no montador

- No montador, tornam-se necessários:
  - **endereços relativos** – uma pseudo-instrução especial deve indicar que se trata de origem relativa;
  - **importar símbolos** – para que um símbolo **X** de outro programa possa ser referenciado no programa;
  - **exportar símbolos** – para que um ponto **X** do programa possa ser referenciado em outros programas;
  - anexar, ao final da montagem, todos os **símbolos não-resolvidos** ao programa-objeto, para que essa informação possa ser passada posteriormente ao *programa ligador* (*linker*);
  - Gerar **código-objeto no formato compatível** com o *loader* hexadecimal (função **P** do simulador MVN).

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Alterações no Montador

- A inserção das seguintes modificações no montador absoluto são necessárias:
  - Inclusão e tratamento das novas pseudo instruções, para declarar:
    - **&** – Origem relocável
    - **>** – Endereço simbólico *interno para exportar* (*entry point*)
    - **<** – Endereço simbólico *externo para importar* (external)
  - Geração de código-objeto no novo formato:
    - Origem absoluta e relocável
    - Operando absoluto e relocável

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplos

- **&** – Origem relocável

```
ABC & /01AC  ; ASSOCIA A ORIGEM CORRENTE AO SÍMBOLO ABC
             ; NOVA ORIGEM (RELOCÁVEL) É /01AC + BASE DE RELOCAÇÃO
    & /0000  ; INICIA A ORIGEM (RELOCÁVEL) EM 0
XYZ ...      ; XYZ FICA ASSOCIADO AO ENDEREÇO RELOCÁVEL 0
```

- **>** – Endereço simbólico de entrada (*entry point*)

```
ABC >    ; DECLARA QUE O SÍMBOLO INTERNO ABC ESTÁ SENDO EXPORTADO
```

- **<** – Endereço simbólico externo (*external*)

```
ABC <    ; DECLARA QUE O SÍMBOLO EXTERNO ABC ESTÁ SENDO IMPORTADO
```

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Tipos de endereços no programa-objeto

- Há dois aspectos a considerar:
  - o endereço onde será gerado o código;
  - os endereços referenciados pelo código.
- Endereço onde o código deve ser gerado:
  - Absoluto ou relocável.
- Endereço referenciado pelo código:
  - Resolvido ou pendente (não-resolvido): endereços **externos** são não-resolvidos, endereços internos não-resolvidos são erros!
  - Absoluto ou relocável: somente para endereços **internos;** para endereços externos designa-se como absoluto;
  - Interno ou externo (em relação à localidade do endereço referenciado (operando), todos os endereços **importados (<)** no módulo são considerados **externos**, os demais (exportáveis e rótulos locais) são considerados **internos (>)**.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

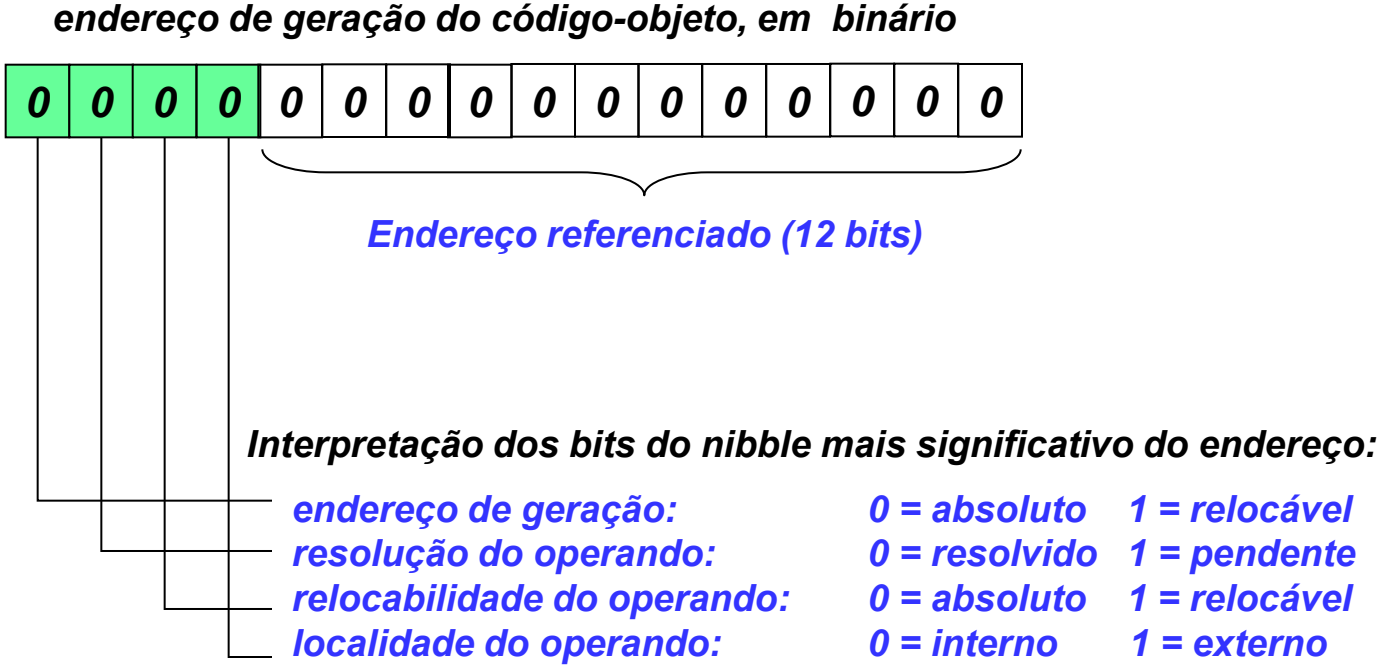
Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Formatos no programa-objeto relocável

- Cada código gerado incorpora duas componentes de endereço:
  - O Endereço onde deve ser gerada a instrução (absoluto/relocável)
  - Operando referenciado (resolvido/pendente, absoluto/relocável, interno/externo)
- Pode-se codificar esses atributos nos quatro bits mais significativos do endereço onde o código deve ser gerado (até aqui, esses bits sempre foram nulos), já que o endereço ocupa apenas 12 bits



PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Novas pseudo-instruções

Em adição às pseudo-instruções já utilizadas:

- @ (define uma ORIGEM ABSOLUTA para o código a ser gerado)
  - Exemplo: @ /0050 ;indica /0050 como origem do código seguinte .
- # (define o FIM físico do programa)
  - Exemplo: # X ; indica que X é o endereço de execução do programa.
- K (define uma área preenchida por uma CONSTANTE de 2 bytes)
  - Exemplo: XYZ K /0010 ; Gera /0010 na posição correspondente a XYZ.
- $ (define um BLOCO DE MEMÓRIA com número especificado de words)
  - Exemplo: XYZ $ =30 ; reserva 30 words a partir do endereço simbólico XYZ (Operando = número de words a serem reservadas para o bloco)

incluir-se-ão as seguintes novas pseudo-instruções:

- & (define uma ORIGEM RELOCÁVEL para o código a ser gerado)
  - Exemplo: & /0050 ;indica que o próximo código se localizará no endereço /0050, relativo à origem do código corrente.
- > (define um endereço simbólico local como entry-point do programa)
  - Exemplo: ABC > ; indica que o símbolo interno ABC está sendo exportado
- < (define um endereço simbólico que referencia um entry-point externo)
  - Exemplo: ABC < ; indica que o símbolo externo ABC está sendo importado

Exemplo: Somador

- Programa somador.asm

```
; Somador
; *******
; Somador que recebe duas entradas, nas posições
; ENTRADA1 e ENTRADA2, e coloca o resultado da
; soma na posição SAIDA (externa).

SOMADOR >                   ; interno
ENTRADA1 >                  ; interno
ENTRADA2 >                  ; interno
SAIDA <                     ; externo

& /0000                     ; Origem relocável
; Entradas do programa.
ENTRADA1 K /0000
ENTRADA2 K /0000

; Programa
SOMADOR  JP /000            ; Ponto de entrada da subrotina
INICIO   LD ENTRADA1
         +  ENTRADA2
         MM SAIDA           ; Colocando na saída
         RS SOMADOR         ; Retornando
# INICIO
```

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

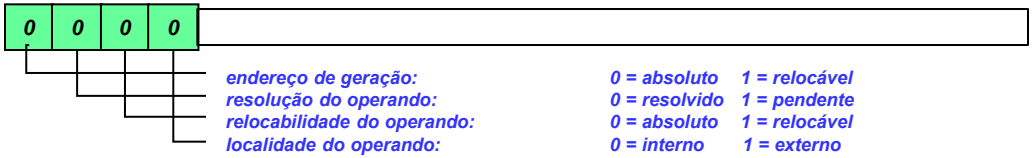
Exemplo: Somador

- Código

	Endereço de geração	Resolução do operando	Relocabilidade do operando	Localidade do operando
SOMADOR >	0	0	1	0
ENTRADA1 >	0	0	1	0
ENTRADA2 >	0	0	1	0
SAIDA <	0	1	0	**0**
& /0000				
ENTRADA1 K /0000	1	0	0	0
ENTRADA2 K /0000	1	0	0	0
SOMADOR JP /000	1	0	0	0
INICIO LD ENTRADA1	1	0	1	0
+ ENTRADA2	1	0	1	0
MM SAIDA	1	1	0	1
RS SOMADOR	1	0	1	0

```
2004 0000 ; "SOMADOR>"
2000 0000 ; "ENTRADA1>"
2002 0000 ; "ENTRADA2>"
4000 0000 ; "SAIDA<"

8000 0000
8002 0000
8004 0000
a006 8000
a008 4002
d00a 9000
a00c b004
```



Interpretação dos bits do nibble mais significativo do endereço:

Exemplo: Somador

- Programa principal.asm

```
; Principal
; *********
; Programa principal que chama o somador.

SOMADOR <                   ; externo
ENTRADA1 <                  ; externo
ENTRADA2 <                  ; externo
SAIDA >                     ; interno

& /0000
          JP INICIO
VALOR1    K =50             ; valor 1 a somar
VALOR2    K #101101         ; valor 2 a somar
SAIDA     K /0000           ; valor de retorno

INICIO    LD VALOR1         ; passando as variáveis
          MM ENTRADA1
          LD VALOR2
          MM ENTRADA2
          SC SOMADOR        ; chamando o somador
          HM /00
# INICIO
```

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

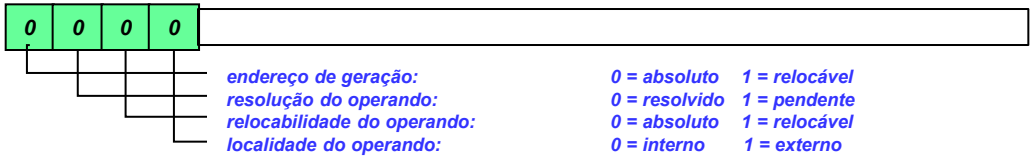
Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo: Somador

- Código
  - Programa principal



Interpretação dos bits do nibble mais significativo do endereço:

	Endereço de geração	Resolução do operando	Relocabilidade do operando	Localidade do operando
SOMADOR <	0	1	0	0
ENTRADA1 <	0	1	0	0
ENTRADA2 <	0	1	0	0
SAIDA >	0	0	0	0
& /0000				
JP INICIO	0	0	0	0
VALOR1 K =50	0	0	0	0
VALOR2 K #101101	0	0	0	0
SAIDA K /0000	0	0	0	0
INICIO LD VALOR1	0	0	0	0
MM ENTRADA1	0	1	0	1
LD VALOR2	0	0	0	0
MM ENTRADA2	0	1	0	1
SC SOMADOR	0	1	0	1
HM /00	0	0	0	0

```
4000 0000 ; "SOMADOR<"
4002 0000 ; "ENTRADA1<"
4004 0000 ; "ENTRADA2<"
0006 0000 ; "SAIDA>"

0000 0008
0002 0032
0004 002d
0006 0000
0008 8002
500a 9002
000c 8004
500e 9004
5010 a000
0012 c000
```

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Ligador e Relocador

- Os programas de sistema **Ligador** e **Relocador** são responsáveis pela junção de todas as partes de um programa que pode ser desenvolvido em módulos, e posteriormente unido.
- O ligador junta todos os módulos e gera uma única saída contendo todas as partes.
- O relocador é responsável por estabelecer o endereço de carga e execução final do programa.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Ligador

- No programa objeto relocável surgiram dois novos tipos de endereços simbólicos:
  - ***Entry points*** – correspondem a rótulos do módulo corrente que devem ser visíveis a partir de outros módulos, e para isso tais módulos devem declará-los como ***externals***.
  - ***Externals*** – correspondem a rótulos declarados como ***entry points*** em outros módulos, e que serão utilizados pelo módulo corrente.
  - Para que um programa formado por diversos módulos fique completo, todos os símbolos declarados como *externals* em algum módulo deverão figurar também como *entry points* em algum dos outros módulos.
  - Cabe ao ligador efetuar essa associação entre *externals* e *entry points*, juntando diversos módulos em um só.
  - Símbolos que permanecem não-resolvidos são mantidos como *externals* ou como *entry points* conforme for o caso.

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

Exemplo de Operação do Ligador (1)

- Neste exemplo, admita-se que se deseje ligar três módulos independentemente montados usando o montador relocável (ver figura no *slide* seguinte):
  - Módulo 1. Contém um programa principal A, e uma função B, e faz referência aos subprogramas C e D. Este módulo ocupa 500 bytes, e os *entry points* A e B correspondem aos endereços relativos 100 e 200, respectivamente.
  - Módulo 2. Contém o subprograma C, que faz referência ao programa A e ao subprograma D. Este módulo ocupa 200 bytes, e o *entry point* C corresponde ao endereço relativo 120.
  - Módulo 3. Contém o subprograma D, que referencia o programa A e os subprogramas B, C e D. Este módulo ocupa 150 bytes, e o *entry point* D corresponde ao endereço relativo 50.
- Com essas hipóteses, o ligador vai receber os três módulos na ordem apresentada.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Ligador (2)

Módulos produzidos pelo montador relocável

Módulo 1: (500 bytes)	*Módulo 2: (200 bytes)*	*Módulo 3: (150 bytes)*
& /0	*& /0*	*& /0*
C < ; subprograma	*A <; programa*	*A < ; programa*
D < ; subprograma	*D < ; subprograma*	*B < ; subprograma*
A > ; principal	*C > ; subprograma*	*C < ; subprograma*
***100 A** ; end. relativo de A*	***120 C** ; end. relativo de C*	*D > ; subprograma*
B > ; função		***50 D** ; end. relativo de D*
***200 B** ; end. relativo de B*		

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Ligador (3)

- Inicialmente a base de alocação é zerada.
- Lê-se o Módulo 1, e todas as referências a endereços relocáveis são corrigidas somando-se-lhes a base de alocação. Todas as referências a símbolos ainda não presentes na tabela devem ficar pendentes, como foi feito no montador.
- Todos os *externals* são adicionados (se aí já não estiverem) à tabela de símbolos e marcados como tais.
- Todos os *entry points* são também coletados e marcados como tais. Caso algum deles corresponda a algum dos *externals* contidos na tabela de símbolos, o valor do *location counter* (base de alocação) associado a tal *entry point* resolve esse símbolo, e portanto deve-se resolver as pendências associadas a tal símbolo, do mesmo modo como foi feito no montador.

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

Exemplo de Operação do Ligador (4)

- O Módulo 1 contém um programa principal A, uma função B, e faz referência aos subprogramas C e D. Este módulo ocupa 500 bytes, e os *entry points* A e B correspondem aos endereços relativos 100 e 200, respectivamente.
- A base de alocação contém o endereço 0.
- Resulta na tabela de símbolos:

  A = endereço relativo 100+0 = 100 (resolvido)

  B = endereço relativo 200+0 = 200 (resolvido)

  C = *external* indefinido

  D = *external* indefinido
- A nova base de alocação será 0+500 = 500

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Ligador (5)

- Se nenhum módulo adicional for apresentado ao ligador, este deve completar sua tarefa gerando como parte do código objeto o conteúdo de todas as pendências para que a operação de ligação possa prosseguir posteriormente.

- No caso do exemplo, deveriam ser geradas todas as pendências referentes aos símbolos declarados como *externals* e ainda não resolvidos, ou seja, os referentes aos símbolos C e D.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Ligador (6)

- O Módulo 2 contém o subprograma C, que faz referência ao programa A e ao subprograma D. Este módulo ocupa 200 bytes e o *entry point* C corresponde ao endereço relativo 120.
- A base de alocação contém o endereço 500.
- Resulta na tabela de símbolos:

  A = endereço relativo 100 (permanece)

  B = endereço relativo 200 (permanece)

  C = endereço relativo 120+500 = 620 (resolvido)

  D = *external* indefinido (permanece)
- A nova base de alocação será 500+200 = 700

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Ligador (7)

- Se nenhum módulo adicional for apresentado ao ligador, este deve completar sua tarefa gerando como parte do código objeto o conteúdo de todas as pendências para que a operação de ligação possa prosseguir posteriormente.

- No caso deste exemplo, deveriam ser geradas todas as pendências referentes aos símbolos declarados como *externals* e ainda não resolvidos, ou seja, os referentes ao símbolo D.

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

Exemplo de Operação do Ligador (8)

- O Módulo 3 contém o subprograma D, que referencia o programa A e os subprogramas B, C e D. Este módulo ocupa 150 bytes, e o *entry point* D corresponde ao endereço relativo 50.
- A base de alocação contém o endereço 700.
- Resulta na tabela de símbolos:

  A = endereço relativo 100 (permanece)

  B = endereço relativo 200 (permanece)

  C = endereço relativo 620 (permanece)

  D = endereço relativo 50+700 = 750 (resolvido)
- A nova base de alocação será 700+150 = 850

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Ligador (9)

- Todos os endereços simbólicos referenciados entre módulos já estão resolvidos, portanto nada mais há para ligar, e o trabalho do ligador se encerra neste ponto.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Ligador (10)

Módulo 1: (500 bytes)	*Módulo 2: (200 bytes)*	*Módulo 3: (150 bytes)*
& /0	*& /0*	*& /0*
C < ; subprograma	*A <; programa*	*A < ; programa*
D < ; subprograma	*D < ; subprograma*	*B < ; subprograma*
A > ; principal	*C > ; subprobrama*	*C < ; subprograma*
***100 A** ; end. relativo de A*	***120 C** ; end. relativo de C*	*D > ; subprograma*
B > ; função		***50 D** ; end. relativo de D*
***200 B** ; end. relativo de B*		

Ligador gera:

Módulos Ligados: (850 bytes)

***100 A** (considera-se base de alocação 0)*

***200 B** (considera-se base de alocação 0)*

***620 C** (considera-se base de alocação 500 = tamanho do módulo 1)*

***750 D** (considera-se base de alocação 700 = soma dos tamanhos dos módulos 1 e 2)*

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Funcionamento do Relocador (1)

- Como produto da execução do ligador, ao final pode ser obtido um único módulo relocável, isento de referências a símbolos externos não-resolvidos.
- Esse módulo integra todos os módulos menores a partir dos quais foi gerado, mas ainda não pode ser executado pois todos os endereços do seu espaço de endereçamento são relativos.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Funcionamento do Relocador (2)

- É preciso eleger uma região de memória onde esse módulo deverá ser executado, e corrigir (**relocar**) no código todas as referências a endereços relativos, adicionando-lhes o endereço-base escolhido. O programa que executa esta tarefa denomina-se **relocador**.
- Como resultado, surge um programa-objeto equivalente, porém absoluto e pronto para a execução, o qual pode ser carregado na memória usando os métodos já conhecidos, empregados na programação absoluta.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Funcionamento do Relocador (3)

- Em algumas instalações, opta-se por fundir em um único programa o relocador e o ligador, obtendo-se o **relocador-ligador**.
- Perde-se a flexibilidade de obter módulos relocáveis intermediários, a partir da resolução parcial dos endereços externos referenciados.
- Em ambientes mais simples, o relocador-ligador pode ser muito útil e prático, evitando a criação de programas-objeto intermediários adicionais.
- Para a MVN serão utilizados programas distintos para o ligador e para o relocador.

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

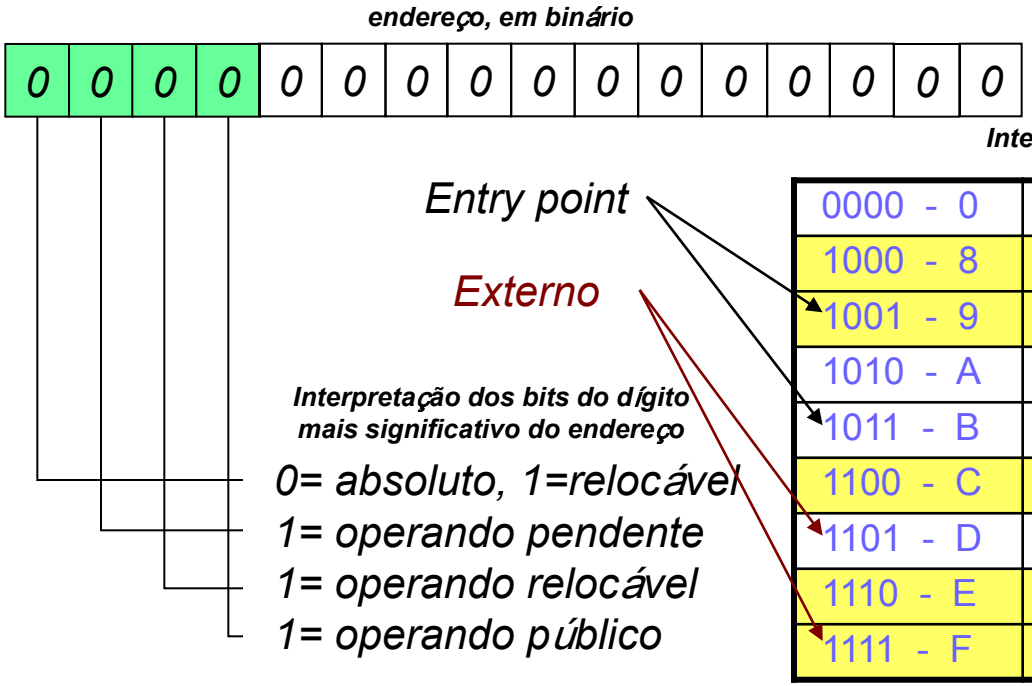
Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

Formatos

- 0134 5423 endereço absoluto 134, opcode 5, operando absoluto 423
- 7134 5002 endereço absoluto 134, opcode 5, operando externo nº 002 (**TS**)
- 9134 5423 endereço relocável 134, opcode 5, operando absol. público 423
- A134 5423 endereço relocável 134, opcode 5, operando relocável 423
- B134 5423 endereço relocável 134, opcode 5, operando relocável público 423
- D134 5002 endereço relocável 134, opcode 5, operando externo nº 002 (**TS**)
- F134 5002 endereço relocável 134, opcode 5, operando externo nº 002 (**TS**)



0000 - 0	End. e oper. absolutos locais
1000 - 8	End. relocável, oper. absoluto local
1001 - 9	End. relocável, oper. abs. público
1010 - A	End. e oper. relocáveis locais
1011 - B	End. relocável oper. relocável público
1100 - C	End. relocável oper. pendente local
1101 - D	End. relocável oper. pendente público
1110 - E	End. relocável oper. pendente reloc. loc
1111 - F	End. relocável oper. pend. reloc. púb.

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Exemplo de Operação do Relocador (2)

Módulos Ligados: (850 bytes)

100 A *(considera-se base de alocação 0)*

200 B *(considera-se base de alocação 0)*

620 C *(considera-se base de alocação 500 = tamanho do módulo 1)*

750 D *(considera-se base de alocação 700 = soma dos tamanhos dos módulos 1 e 2)*

Relocador gera:

Módulos Ligados Absolutos: (850 bytes)

200 A *(base de relocação 100)*

300 B *(base de relocação 100)*

720 C *(base de relocação 100)*

850 D *(base de relocação 100)*

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Estruturação de código

- Criação de programas estruturados
  - Uso de subprogramas
  - Reuso de programas disponíveis em bibliotecas
- Programação estruturada
  - Linguagens de programação estruturadas
    - C, Pascal, Ada, Fortran, Modula

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Anexo

Combinações possíveis no Montador Relocável

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Combinações possíveis – caso geral

	Endereço de geração	Resolução do operando	Relocabilidade do operando	Localidade do operando
0000	absoluto	resolvido	absoluto	interno
0001	absoluto	resolvido	absoluto	externo
0010	absoluto	resolvido	relocável	interno
0011	absoluto	resolvido	relocável	externo
0100	absoluto	pendente	absoluto	interno
0101	absoluto	pendente	absoluto	Externo
0110	absoluto	pendente	relocável	Interno
0111	absoluto	pendente	relocável	externo
1000	relocável	resolvido	absoluto	interno
1001	relocável	resolvido	absoluto	externo
1010	relocável	resolvido	relocável	interno
1011	relocável	resolvido	relocável	externo
1100	relocável	pendente	absoluto	interno
1101	relocável	pendente	absoluto	externo
1110	relocável	pendente	relocável	interno
1111	relocável	pendente	relocável	externo

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

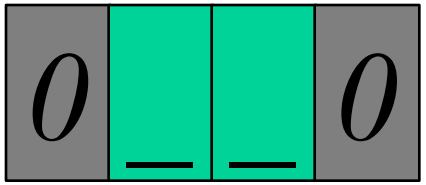
Reestruturação:
Paulo S. Muniz Silva

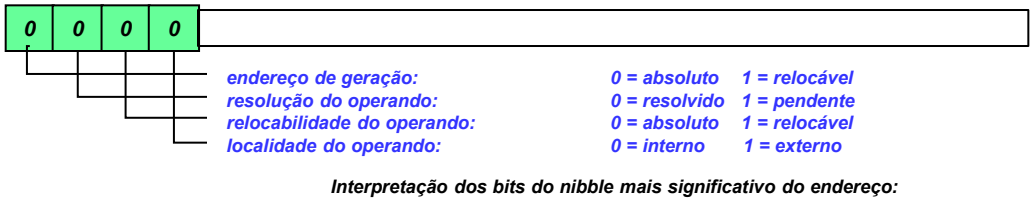
v.1.1 ago 2012

Combinações possíveis no montador

Pseudo-instruções

- **Entry point >**
  - `ABC >`
- **External <**
  - `ABC <`
- Utilização dos bits.





Interpretação dos bits do nibble mais significativo do endereço:

- **Endereço de geração:** sempre zero.
- **Resolução do operando:** zero/um.
- **Relocabilidade do operando:** zero/um.
- **Localidade do operando:** como ele tem o mesmo valor do bit “Resolução”, pode ser considerado como informação redundante. Otimizando, terá sempre zero como valor.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Combinações possíveis no montador
Pseudo-instruções

- 3 combinações possíveis
  - Declaração de variável externa (importada): o segundo bit é igual a um.
  - Declaração de variável interna (exportada) com endereço absoluto: o terceiro bit é igual a zero.
  - Declaração de Variável interna (exportada) com endereço relativo: o terceiro bit é igual a um.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

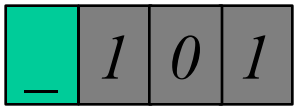
Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Combinações possíveis no montador
Instruções

- Instrução com variáveis externas (*)
  - `SOMADOR <  ;Pseudo-instrução`
  - `MM  SOMADOR  ;  Instrução com variável externa`

- **Endereço de geração:** zero/um.
- **Resolução do operando:** sempre um.
- **Relocabilidade do operando:** sempre zero.
- **Localidade do operando:** sempre um.



(*) Combinações diferentes destas citadas são casos de erros e devem ser corretamente tratadas.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Aula 3:

Montador Relocável

Autores:

Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

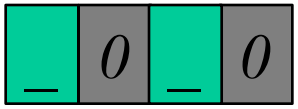
Reestruturação:
Paulo S. Muniz Silva

v.1.1 ago 2012

Combinações possíveis no montador
Instruções

- Instruções com variáveis internas (*)
  - `SAIDA > ; Pseudo-instrução`
  - `LD SAIDA ; instrução com variável interna`

- **Endereço de geração:** zero/um.
- **Resolução do operando:** sempre zero.
- **Relocabilidade do operando:** zero/um.
- **Localidade do operando:** sempre zero.



(*) Combinações diferentes destas citadas são casos de erros e devem ser corretamente tratadas.

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo L. A. Rocha

Autores:

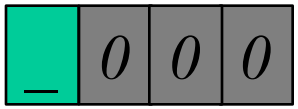
Anna H. R. Costa
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Reestruturação:
Paulo S. Muniz Silva

Combinações possíveis no montador
Instruções

- Declaração de variáveis (*)
  - `SAIDA > ; Pseudo-instrução`
  - `SAIDA K /100 ; Pseudo-instrução`

- **Endereço de geração:** zero/um.
- **Resolução do operando:** sempre zero.
- **Relocabilidade do operando:** sempre zero.
- **Localidade do operando:** sempre zero.



(*) Combinações diferentes destas citadas são casos de erros e devem ser corretamente tratadas.