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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

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

Aula 09

Montador Relocável

Professores:
Anarosa A. F. Brandão (PCS 2302)
Jaime Simão Sichman (PCS 2302)
Reginaldo Arakaki (PCS 2024)
Ricardo Luís de Azevedo da Rocha (PCS 2024)

Monitores: Diego Queiroz e Tiago Matos

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha
© 2009

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

Roteiro

1. Necessidade de programas relocáveis
2. Implicações na linguagem simbólica
   - Novas pseudo-instruções
   - Novo formato de instrução
3. Montador relocável
   - Diagrama de classes
   - Exemplo de funcionamento
4. Parte Experimental
   - Implementação de um montador relocável para o simulador MVN

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha
© 2009

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

3

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
  - 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
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha
© 2009

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

4

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!

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

5

Implicações na linguagem simbólica

- Para que se possa exprimir um programa relocável e 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

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

6

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
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

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 não-resolvido (endereços **externos** são não-resolvidos, endereços locais não-resolvidos são erros!)
  - Absoluto ou relocável (somente para endereços **locais**, para endereços externos designa-se como absoluto)
  - Local ou público (em relação à localidade do endereço referenciado (operando), todos os endereços **externos** ao módulo (**importados**) e os **exportados** pelo módulo são operandos **públicos**, os demais devem ser **locais**.

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

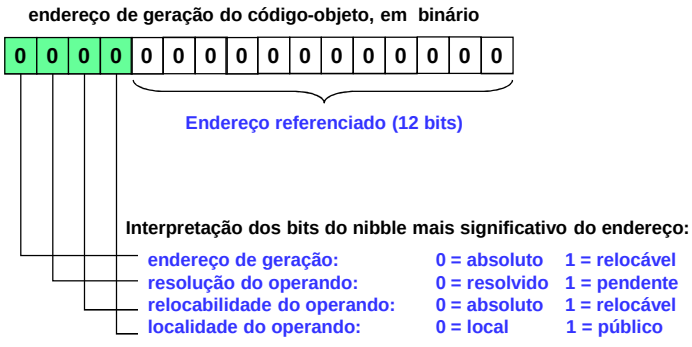
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

Formatos no programa-objeto relocável

- Cada código gerado incorpora duas componentes de endereço:
  - Sobre o Endereço onde deve ser gerado (absoluto/relocável)
  - Operando referenciado (resolvido/não, absoluto/relocável, local/público)
- 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
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha
© 2009

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

9

Combinações possíveis

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

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha
© 2009

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

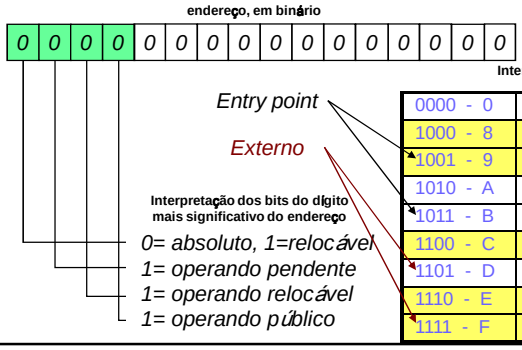
Revisores:

v. 1.4 ago 2009

10

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**)



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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

11

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 de entrada (entry point)
    - **<** – Endereço simbólico externo (external)
  - Geração de código-objeto no novo formato:
    - Origem absoluta e relocável
    - Operandos absoluto e relocável

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

12

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 >       ; ASSOCIA A ORIGEM CORRENTE AO ENTRY-POINT ABC
```

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

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

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

13

Alterações complementares

- Para atingir toda a sua funcionalidade, as seguintes adições posteriores serão necessárias:
  - Geração de código-objeto no novo formato, incluindo:
    - Operando simbólico
    - Endereços simbólicos de entrada e externos
    - Outras referências simbólicas não-resolvidas
  - Alteração do dumper hexadecimal: incluir referências simbólicas
  - Algoritmo de relocação a partir de uma base estabelecida
  - Alteração do loader hexadecimal: incluir relocação

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

14

Novas pseudo-instruções

Em adição às pseudo-instruções já utilizadas:

- @ (define uma ORIGEM ABSOLUTA para o código a ser gerado)
  - Exemplo: @ /50 ;indica /050 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 /10 ; Gera /10 na posição correspondente a XYZ
- $ (define um BLOCO DE MEMÓRIA com número especificado de bytes)
  - Exemplo: XYZ $ =30 ; reserva 30 bytes, e o primeiro chama-se XYZ (Operando = número de bytes a serem reservados para o bloco)

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

- & (define uma ORIGEM RELOCÁVEL para o código a ser gerado)
  - Exemplo: & /50 ;indica que o próximo código se localizará no endereço /050, relativo à origem do código corrente.
- > (define um endereço simbólico local como entry-point do programa)
  - Exemplo: ABC > ; indica que o símbolo ABC está sendo exportado
- < (define um endereço simbólico que referencia um entry-point externo)
  - Exemplo: ABC < ; indica que ABC é um símbolo importado

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha
© 2009

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

15

Exemplo: programa em linguagem simbólica

O programa abaixo, que foi dado como exemplo na aula 5:

```
A100        8F00   Obtém o endereço para onde se deseja mover o dado
A102        4F02   Compõe o endereço com o código de operação Move
A104        9106   Guarda instrução montada para executar em seguida
8106        9000   Executa a instrução recém-montada
A108        ....   Provavelmente, o código seguinte altera o conteúdo de 0F00
....
A15C        0100   Volta a repetir o procedimento, para outro endereço.
....
AF00        034C   Endereço (34C) para onde se deseja mover o dado
8F02        9000   Código de operação Move, com operando 000
```

codificado em linguagem simbólica, fica com o seguinte aspecto:

```
  & /0100     ; &=origem do código 0100=posição de memória (em hexadecimal)
P LD E        ; P=rótulo LD=load E=endereço simbólico da constante 034C
  +  M        ; +=add M=rótulo de onde está uma instrução Move 0000
  MM X        ; MM=move X=endereço da instrução seguinte
X MM 0        ; reservado para guardar a instrução recém-montada
  ...
  JP P        ; JP=jump (desvio) P=rótulo da primeira instrução deste programa
  ...
E K  034C     ; E=rótulo K=constante 034C=operando numérico, em hexadecimal
M MM /0000    ; M=rótulo MM=move 0000=operando zero
  # P         ; #=final físico P=rótulo da primeira instrução a ser executada
```

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha
© 2009

Aula 9:

Montador Relocável

Autores:

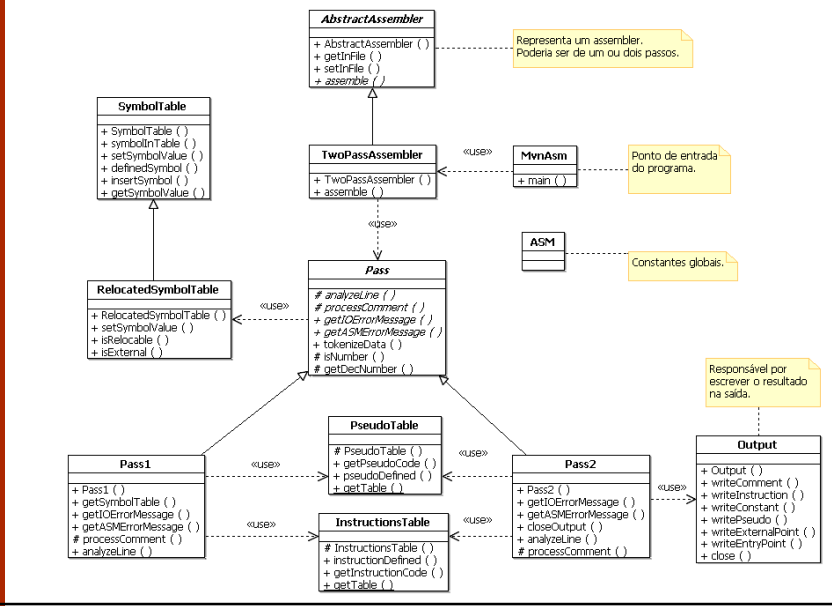
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

16

Diagrama de Classes do Montador relocável



Classes do Montador Relocável

- O montador é definido a partir da classe abstrata (*AbstractAssembler*). O montador construído para esta disciplina é de dois passos (*TwoPassAssembler*). As demais classes (oriundas do montador absoluto) são:
  - **Tabela de instruções** (*InstructionsTable*): define as instruções válidas (símbolo e valor).
  - **Tabela de pseudo-instruções** (*PseudoTable*): define as pseudo-instruções válidas (símbolo e valor).
  - **Tabela de símbolos** (*SymbolTable*): permite armazenar e recuperar os rótulos (símbolo e endereço real).
  - **Passo** (*Pass*): define a estrutura dos passos, que são derivados desta classe (*Pass1* e *Pass2*).
  - **Saída** (*Output*): responsável por toda saída de dados para os arquivos.
  - **Constantes** (*ASM*): define as constantes utilizadas no montador.
  - **Ponto de entrada** (*MvnAsm*): contém o aplicativo que inicia o montador.
- A nova classe é:
  - **Tabela de símbolos relocáveis** (*RelocatedSymbolTable*): armazena e recupera os símbolos relocáveis.
- Nas classes existentes devem ser introduzidas mudanças.

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 >
ENTRADA1 >
ENTRADA2 >
SAIDA <

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

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

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

Exemplo: Somador

- Tabela de símbolos

	isRelocable	isExternal	Endereco (hexa)
SOMADOR	true	false	0004
ENTRADA1	true	false	0000
ENTRADA2	true	false	0002
SAIDA	?	true	?

- Código

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

```
3004 0000 ; "SOMADOR>"
3000 0000 ; "ENTRADA1>"
3002 0000 ; "ENTRADA2>"
5000 0000 ; "SAIDA<"

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

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

Exemplo: Somador

- Programa principal.asm

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

SOMADOR <
ENTRADA1 <
ENTRADA2 <
SAIDA >

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

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

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

Exemplo: Somador

- Tabela de símbolos

	isRelocable	isExternal	Endereco (hexa)
SOMADOR	?	true	?
ENTRADA1	?	true	?
ENTRADA2	?	true	?
SAIDA	false	false	0006
INICIO	false	false	0008
VALOR1	false	false	0002
VALOR2	false	false	0004

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

Exemplo: Somador

- Código

		Endereço de geração	Resolução do operando	Relocabilidade do operando	Localidade do operando
SOMADOR <			1	?	1
ENTRADA1 <			1	?	1
ENTRADA2 <			1	?	1
SAIDA >			0	0	1
@ /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	?	1
	LD VALOR2	0	0	0	0
	MM ENTRADA2	0	1	?	1
	SC SOMADOR	0	1	?	1
	HM /00	0	0	0	0

```
5000 0000 ; "SOMADOR<"
5001 0000 ; "ENTRADA1<"
5002 0000 ; "ENTRADA2<"
1006 0000 ; "SAIDA>"

0000 0008
0002 0032
0004 002d
0006 0000
0008 8002
500a 9001
000c 8004
500e 9002
5010 a000
0012 c000
```

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

Professores:
Anarosa A. F. Brandão
Jaime S. Sichman
Reginaldo Arakaki
Ricardo L.A. Rocha

Aula 9:

Montador Relocável

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

Revisores:

v. 1.4 ago 2009

Exercício

TXGYYA09E01_09

- A principal tarefa desta aula é implementar o montador relocável completo em Java, usando a lógica ilustrada nas transparências, e completando a especificação. Deve-se manter o padrão utilizado até o momento para o código e a documentação.
- O diagrama de classes do montador é aquele apresentado anteriormente.
- O novo montador relocável deve ser baseado no montador absoluto trabalhado na aula passada para estender a sua funcionalidade.