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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Construção de Programas em Linguagem de Máquina (1)

- Escrever um programa usando diretamente codificação binária não é uma tarefa simples, e tampouco agradável.
- Entretanto, foi este tipo de codificação que permitiu a construção dos primeiros programas do curso.
- Naturalmente, se um programa é muito grande ou se lida com diversas estruturas complexas (listas, etc.), a sua codificação se torna ainda mais difícil e complexa.

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Construção de Programas em Linguagem de Máquina (2)

- Por conta disso, torna-se imprescindível construir alguma **abstração** que facilite a programação e a verificação dos programas.
- A primeira idéia, mais natural, é utilizar o modelo de máquina existente e, a partir dele, definir nomes (mnemônicos) para cada instrução da máquina.
- Posteriormente, verifica-se que somente isso não basta, pois é necessário lidar com os endereços dentro de um programa (rótulos, operandos, sub-rotinas), com a reserva de espaço para tabelas, com valores constantes.
- Enfim, é necessário definir uma **linguagem simbólica**.

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Linguagem Simbólica

- Uma instrução de máquina tem usualmente o aspecto seguinte em sua imagem mnemônica:

0012 JZ 042 ; 1042 0012=rótulo **J**Z=mnemônico 042=operando numérico

- A mesma instrução, em linguagem simbólica, pode ser escrita com ou sem um rótulo simbólico, e pode também referenciar um operando através de um rótulo simbólico ou numérico:

Q JZ R ; Q=rótulo JZ=mnemônico R=operando simbólico
JZ R ; rótulo omitido JZ=mnemônico R=operando simbólico
Q JZ 042 ; Q=rótulo JZ=mnemônico 042=operando numérico
JZ 042 ; rótulo omitido JZ=mnemônico 042=operando numérico

- Convenciona-se que sempre o primeiro elemento da linha é um rótulo; caso o rótulo for omitido deverá haver uma instrução
- Entre os elementos de uma linha deve haver ao menos um espaço
- Cada linha deve conter uma instrução/pseudo-instrução completa
- À direita de um ponto-e-vírgula, todo texto é ignorado (=comentário)
- Mnemônicos e significado das pseudo-instruções:
  - @ (Operando numérico: define endereço da instrução seguinte)
  - **$** (Reserva de área de dados)
  - # (Final físico do texto-fonte. Operando=endereço de execução)
  - **K** (Constante. Operando numérico = valor da constante, em hexadecimal)

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Exemplo de programa em linguagem simbólica

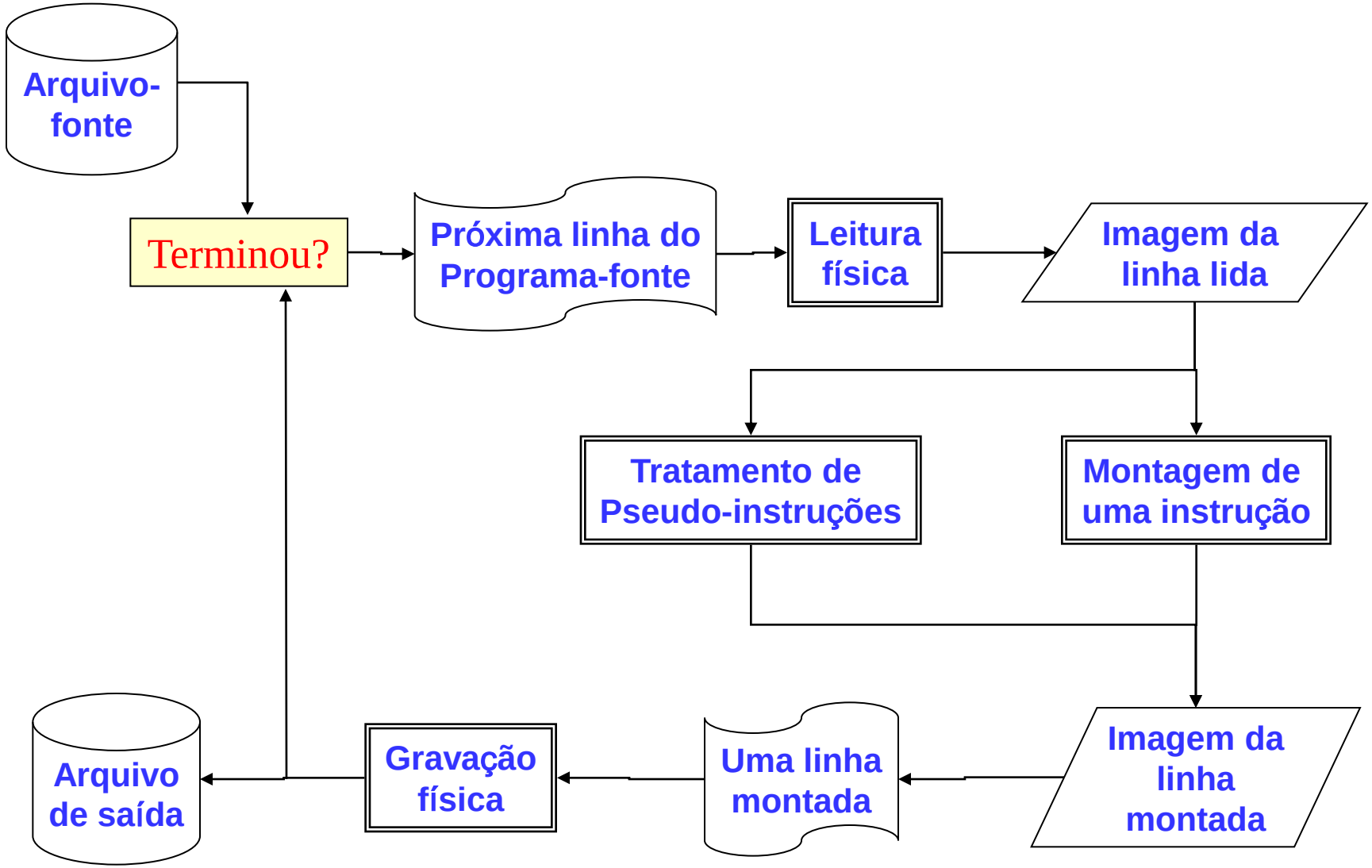
O programa abaixo, que foi dado como exemplo na aula 2:

```
0100          8F00   Obtém o endereço para onde se deseja mover o dado
0102          4F02   Compõe o endereço com o código de operação Move
0104          9106   Guarda instrução montada para executar em seguida
0106          9000   Executa a instrução recém-montada
0108          ....   Provavelmente, o código seguinte altera o conteúdo de 0F00
....
015C          0100   Volta a repetir o procedimento, para outro endereço.
....
0F00          034C   Endereço (34C) para onde se deseja mover o dado
0F02          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
```

Esquema geral de um montador



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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Construção de um Montador

- Para a construção de um montador pressupõe-se que sejam tratadas as seguintes questões:
  - **definição das instruções**: determinar os mnemônicos que as representam simbolicamente;
  - **definição das pseudo-instruções**: determinar os mnemônicos que as representam, bem como sua função para o montador
- Durante a execução de um montador, pressupõe-se que sejam resolvidos os seguintes problemas:
  - **alocação dos rótulos**: determinar qual será o endereço efetivo de um nome encontrado;
  - **geração de código:** gerar um arquivo com o código correspondente em linguagem de máquina
- Para cumprir esta tarefa é necessário completar, em primeiro lugar, as definições dos mnemônicos (instruções e pseudo-instruções), para se pensar posteriormente, nos algoritmos.

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Pseudo-Instruções

- Nas aulas anteriores foram determinados os mnemônicos das instruções, nesta aula serão definidas aqueles relativos às pseudo-instruções.
- As pseudo-instruções utilizadas no montador desta aula (montador absoluto) são as seguintes:
  - **@** : Recebe um operando numérico, define o endereço da instrução seguinte;
  - **K** : Constante, o operando numérico tem o valor da constante (em hexadecimal);
  - **$** : Reserva de área de dados, o operando numérico define o tamanho da área a ser reservada;
  - **#** : Final físico do texto fonte.

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Formas de Construção de um Montador

- Há mais de uma forma de se tratar o construir um Montador. Pelo menos duas são imediatas:
  - Montador de um passo:
    - Lê o código fonte uma única vez;
    - Armazena dinamicamente os rótulos não definidos em uma lista de pendências;
    - Gera o código para cada linha de entrada completamente definida;
    - Resolve uma pendência caso a linha de entrada inicie com um rótulo pendente;
    - Ao final, completa as linhas de código que ainda não haviam sido completamente definidas, resolvendo todos os rótulos pendentes.
  - Montador de dois passos:
    - Lê o código fonte da primeira vez;
    - Num primeiro passo, trata todas as linhas apenas para resolver os endereços dos rótulos e tratar as pseudo-instruções;
    - Lê novamente o código fonte num segundo passo para gerar o código correspondente ao programa

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Estruturas de Dados do Montador (1)

- O montador precisará de um conjunto de estruturas de dados que o permitirão conduzir a tarefa. Dentro deste conjunto, há as seguintes estruturas de dados:
  - **locationCounter**: define a localização atual (endereço corrente) de execução.
  - **Tabela de instruções**: define as instruções válidas (símbolo e valor).
  - **Tabela de pseudo-instruções**: define as pseudo-instruções válidas (símbolo e valor).
  - **Tabela de símbolos**: permite armazenar e recuperar os rótulos (símbolo e endereço real).

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Estruturas de Dados do Montador (2)

- Além destas estruturas, o montador utiliza um conjunto de arquivos (um de entrada e pelo menos dois de saída). Pode ser necessário gerar o texto objeto em algum formato específico, para que um programa *loader* possa carregá-lo na memória.
- Pode-se, ainda, armazenar o conteúdo do texto fonte durante o passo 1 para facilitar a execução do passo 2.

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Construção do Montador

- Nesta disciplina foi escolhido realizar um montador de dois passos. Esta escolha nos conduz à definição das ações a serem realizadas em cada um dos dois passos do montador. Assim temos:
  - **Passo1**: O objetivo é definir os símbolos encontrados, sejam eles rótulos encontrados antes das instruções, ou ainda rótulos de destino de alguma instrução. Para isso deve-se:
    - Manter atualizado o endereço de execução corrente, chamado de **locationCounter**.
    - Armazenar os valores dos símbolos (rótulos) na Tabela de Símbolos (**TS**) para uso posterior no passo 2.
    - Processar as pseudo-instruções.
  - **Passo2**: O objetivo é gerar o código objeto e possivelmente um arquivo de listagem contendo além do código objeto, o texto fonte à direita do código objeto. Para isso, este passo deve:
    - Recuperar os valores dos símbolos (da **TS**).
    - Gerar as instruções.
    - Processar as pseudo-instruções.

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

Tabela de mnemônicos para a MVN (de 2 caracteres)

Operação **0** **J**um**p** Mnemônico **JP**	Operação **1** **J**ump if **Z**ero Mnemônico **JZ**	Operação **2** **J**ump if **N**egative Mnemônico **JN**	Operação **3** **L**oad **V**alue Mnemônico **LV**
Operação **4** **Add** Mnemônico **+**	Operação **5** **Subtract** Mnemônico **–**	Operação **6** **Multiply** Mnemônico *****	Operação **7** **Divide** Mnemônico **/**
Operação **8** **Load** Mnemônico **LD**	Operação **9** Move to **Memory** Mnemônico **MM**	Operação **A** **S**ubroutine **C**all Mnemônico **SC**	Operação **B** **R**eturn from **S**ub. Mnemônico **RS**
Operação **C** **H**alt **M**achine Mnemônico **HM**	Operação **D** **G**et **D**ata Mnemônico **GD**	Operação **E** **P**ut **D**ata Mnemônico **PD**	Operação **F** **O**perating **S**ystem Mnemônico **OS**

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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

Aula 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

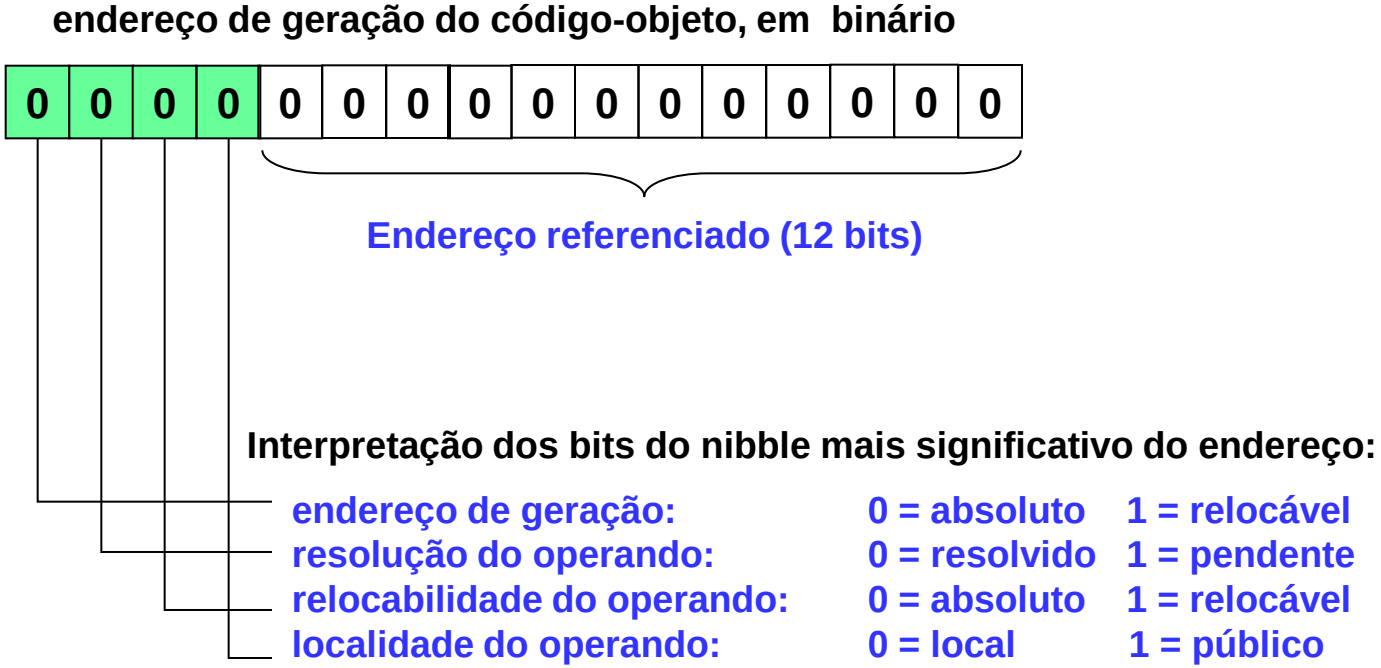
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

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

Aula 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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

Aula 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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

Aula 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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

Aula 8:

Linguagem Simbólica
Montador Absoluto

Autores:

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

Revisores:

v. 1.4 ago 2009

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 8:

Linguagem Simbólica
Montador Absoluto

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
```

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

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 8:

Linguagem Simbólica
Montador Absoluto

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 8:

Linguagem Simbólica
Montador Absoluto

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
```