INSTITUTO MAUÁ DE TECNOLOGIA

MAUÁ

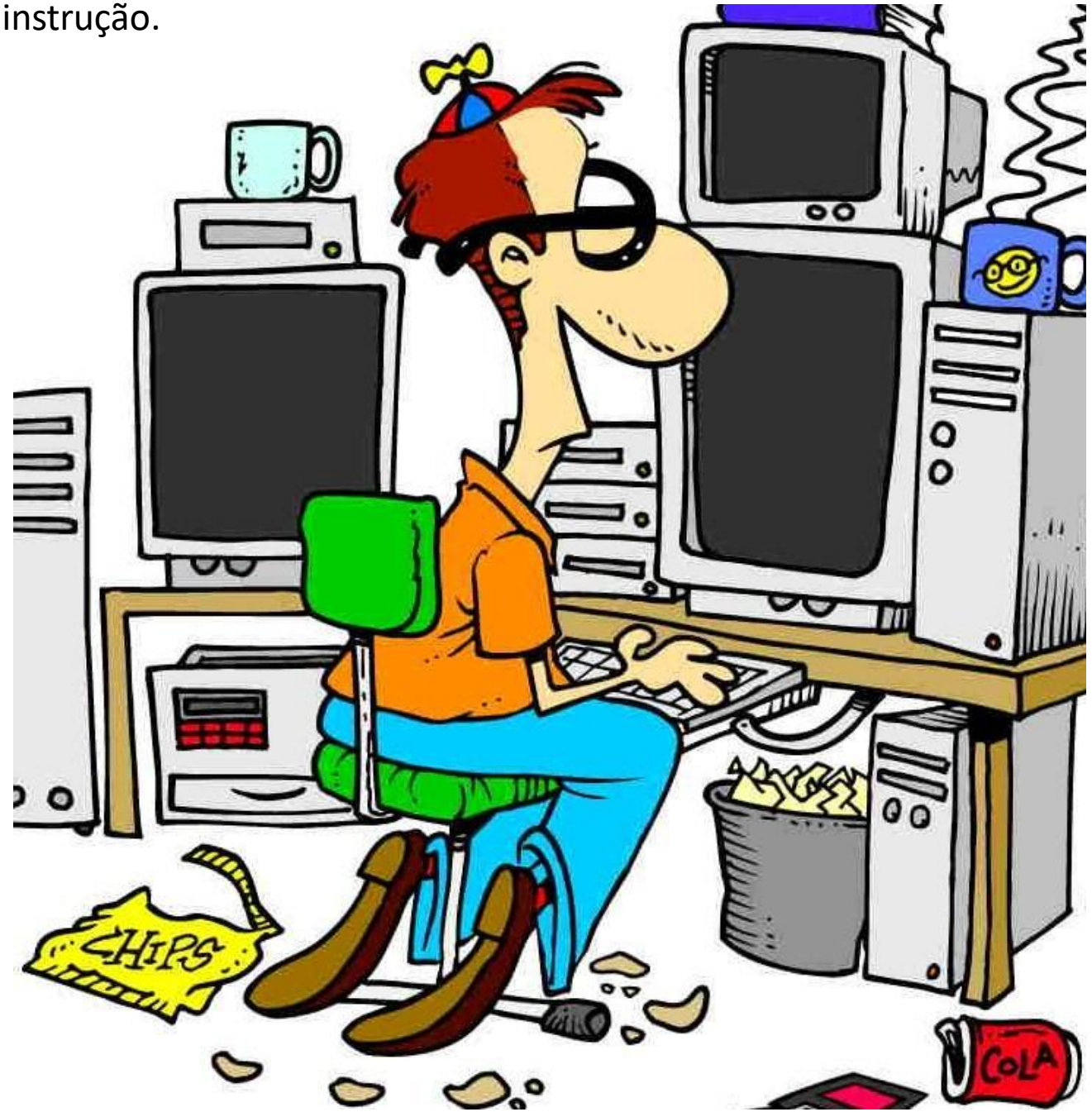
Engenharia de Computação

Arquitetura e Organização de Computadores
ECM245
2017

Professor Dr. João Carlos Lopes Fernandes

E-mail: jlopesf@maua.br

MARIE: Processamento de instrução.



Existem 2 tipos de endereçamento no Marie

Endereçamento direto

- No modo de endereçamento direto, o endereço eficaz do operando é dado no campo de endereço da instrução. A vantagem desse endereçamento é que é necessário apenas um único acesso à memória na busca do operando, e também não há necessidade de cálculos adicionais para encontrar o endereço efetivo. A desvantagem é que o tamanho do número é limitado ao tamanho do endereço.

Exemplo: Add A

- Procura pelo operando na posição “A” da memória
- Adiciona o conteúdo ao acumulador.

Endereçamento indireto

- No modo de endereçamento indireto, o campo de endereço desta vez aponta para uma posição da memória que aponta o endereço do operando. A vantagem desse endereçamento é que para o comprimento de uma palavra N, um espaço de endereço de 2^n (dois elevado à n) pode ser dirigido. A desvantagem, é que a execução acaba sendo mais lenta.

Exemplo: Add A

- Busca em A, encontra o endereço do operando (como Exemplo, B), busca em B pelo operando.
- Adiciona o conteúdo ao acumulador.

Obs.:

O Marie não possui endereçamento Imediato, pois seu programa sempre necessita acessar a memória.

Exemplo: ADD 5.

- Apesar de parecer que o valor cinco será atribuído no acumulador, está errado.
- O Marie buscará na linha 5 da memória o valor, soma ele e o coloca no AC.

Instruções

Processamento de dados: Instruções aritméticas e lógicas.

- Instruções aritméticas – fornecem a capacidade computacional para processamento de dados numéricos.
- Instruções lógicas (booleanas) – operam sobre bits de uma palavra, como bits e não como números.

Armazenamento de dados: Instruções de memória.

- Instruções de memória move dados entre a memória e os registradores.

Movimentação de dados: Instruções de E/S (entrada, saída).

Controle: Instruções de teste e desvio.

- Instruções de teste são usadas para testar o valor de uma palavra de dados ou o estado de uma computação.
- Instruções de desvio são utilizadas para desviar a execução do programa para uma nova instrução.

Sem org.

Linha	
0	load 5
1	add 6
2	store 7
3	output 7
4	halt
5	dec 7
6	dec 9
7	dec 0

Com org. = 100

Linha	
	org 100
100	load 105
101	add 106
102	store 107
103	output 107
104	halt
105	dec 7
106	dec 9
107	dec 0

Código com ênfase nas linhas

No fim do código, é sempre necessário "inicializar" todos os endereços usados no programa, caso contrário não será possível usar o espaço de memória desejado.

```
...
halt
dec 0
hex 1
dec 00011001
```

Inicialização de espaço de memória.

A forma de inicialização do espaço de memória pode ser:

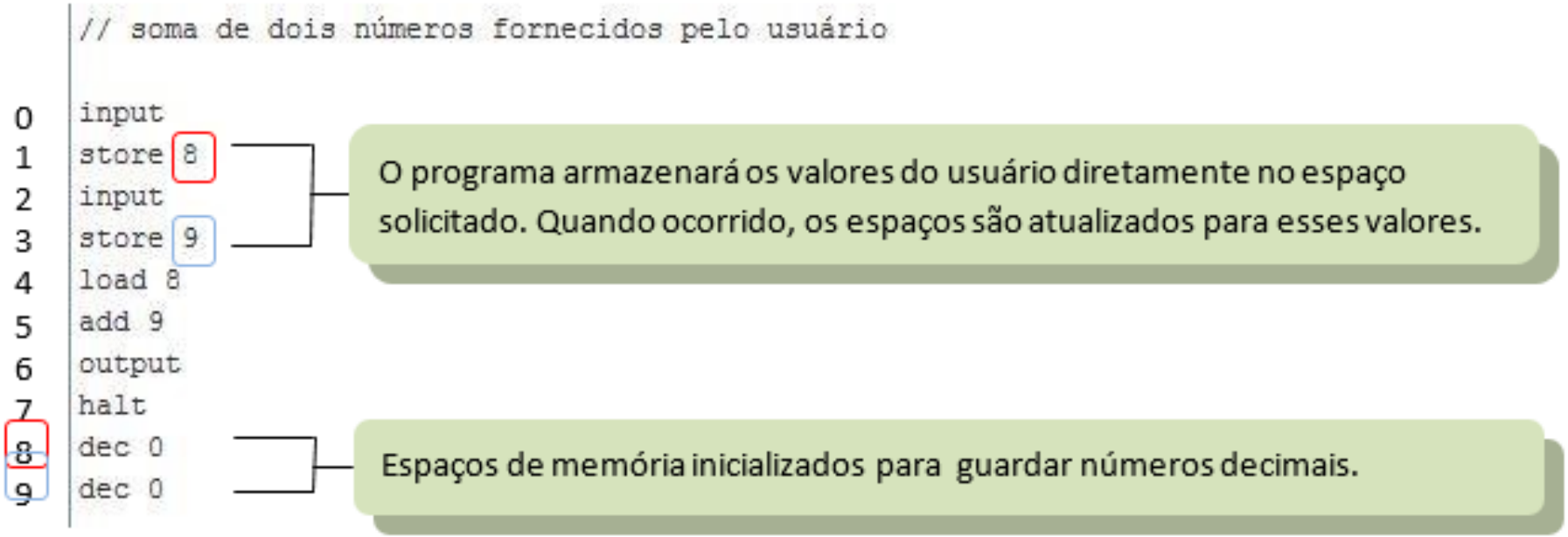
- Decimal: Dec <valor numérico inicial>
- Hexadecimal: Hex <valor numérico inicial>
- ASCII: ASCII <código correspondente em ASCII>

Programas do Marie

- Nos programas do Marie o código do programa e os dados usados são distribuídos na memória de forma sequencial. Por utilizar a linguagem Assembler o Marie não separa programas e os dados na memória, por isso os programas acessam um espaço de memória para pegar um valor desejado.
- Decorrente dessa forma de distribuição na memória existe dois tipos de programas no Marie: ABSOLUTO e REALOCÁVEL.

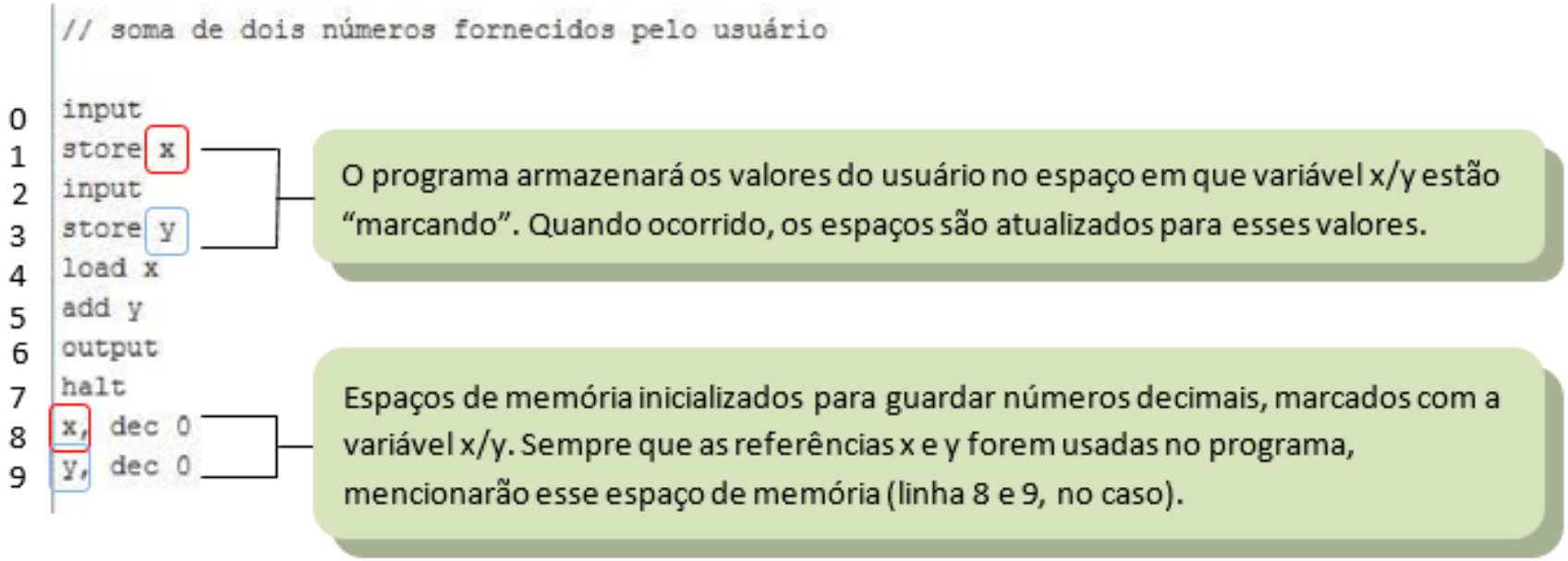
ABSOLUTO

- Programas absolutos são programas estáticos em que os valores são acessados através de um endereço direto da memória.
- Exemplo de programa absoluto:



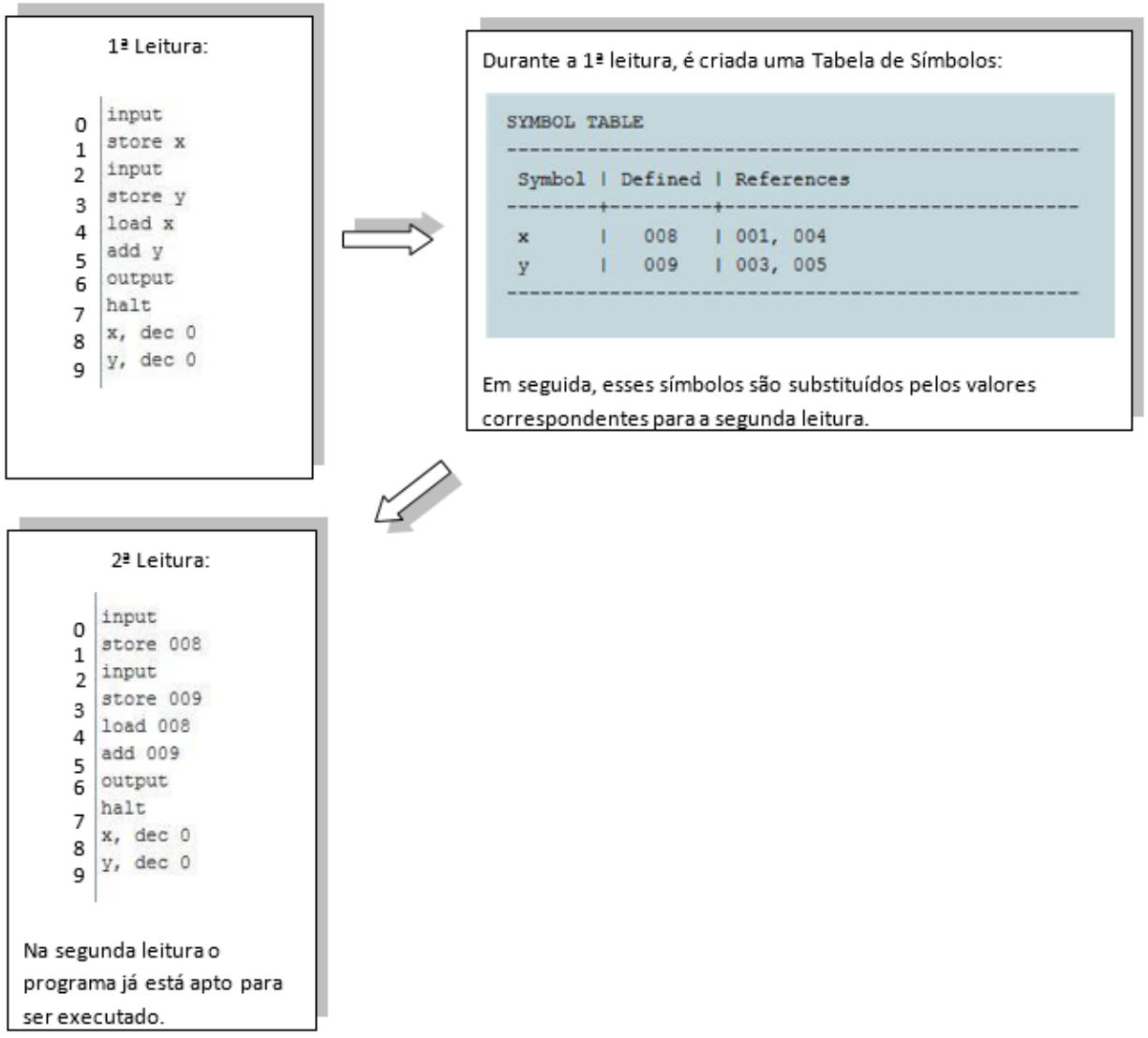
REALOCÁVEL

- No formato realocável não é necessário citar o endereço da memória onde os dados estão ou deverão ser gravados, basta referenciá-lo com variáveis que apontarão para o endereço. O contador de programa percorrerá o algoritmo até encontrar tal variável que, no caso do MARIE, deverá ser declarada no final do programa. Neste formato a programação se torna mais simples.
- Exemplo de programa realocável:



TRADUÇÃO DE NÍVEL 1 E 2

- O conceito de tradução de nível 1 e 2 consiste em quantas vezes é necessário fazer a leitura do programa. Quando o programa é absoluto precisa-se fazer apenas uma leitura e quando é realocável é preciso fazer duas.



MARIE Instruction Set

Type	Instruction	Hex Opcode	Summary
Arithmetic	Add X	3	Adds value in AC at address X into AC, `AC ← AC + X`
	Subt X	4	Subtracts value in AC at address X into AC, `AC ← AC - X`
	AddI X	B	Add Indirect: Use the value at X as the actual address of the data operand to add to AC
	Clear	A	`AC ← 0`
Data Transfer	Load X	1	Loads Contents of Address X into AC
	Store X	2	Stores Contents of AC into Address X
I/O	Input	5	Request user to input a value
	Output	6	Prints value from AC

MARIE Instruction Set

Branch	Jump X	9	Jumps to Address X
	Skipcond (C)	8	Skips the next instruction based on C: if (C) is - 000: Skips if AC < 0 - 400: Skips if AC = 0 - 800: Skips if AC > 0
Subroutine	JnS X	0	Jumps and Store: Stores PC at address X and jumps to X+1
	JumpI X	C	Uses the value at X as the address to jump to
Indirect Addressing	StoreI	E	Stores value in AC at the indirect address. e.g. `StoreI addresspointer` Gets value from addresspointer, stores the AC value into the address
	LoadI	D	Loads value from indirect address into AC e.g. `LoadI addresspointer` Gets address value from addresspointer, loads value at the address into AC
	Halt	7	End the program