





 <p>DEPARTAMENTO DE ENGENHARIA DE COMUNICAÇÃO E SISTEMAS DIGITAIS</p> <p>PCS 2302/2024 Laboratório de Fundamentos da Eng.de Computação</p> <p>Aula 02:</p> <p>Introdução Máq. von Neumann</p> <p>Autores:</p> <p>Anna H. R. Costa Jaime S. Sichman João José Neto Paulo S. Muniz Silva Ricardo L. A. Rocha</p> <p>Reestruturação: Paulo S. Muniz Silva</p> <p>v.1.0 ago. 2012</p> <p>1</p>	<p align="center">PCS-2302 / PCS-2024</p> <p align="center">Lab. de Fundamentos de Eng. de Computação</p>
	<p align="center">Aula 02</p>
	<p align="center">Exercícios Turma 4</p> <p align="center">Professores:</p> <p align="center">Marcos A. Simplício Junior Paulo Sergio Muniz Silva</p>

   <p>DEPARTAMENTO DE ENGENHARIA DE COMUNICAÇÃO E SISTEMAS DIGITAIS</p> <p>PCS 2302/2024 Laboratório de Fundamentos da Eng.de Computação</p> <p>Aula 02:</p> <p>Introdução Máq. von Neumann</p> <p>Autores:</p> <p>Anna H. R. Costa Jaime S. Sichman João José Neto Paulo S. Muniz Silva Ricardo L. A. Rocha</p> <p>Reestruturação: Paulo S. Muniz Silva</p> <p>v.1.0 ago. 2012</p> <p>2</p>	<p align="center">Exercícios</p>
	<ul style="list-style-type: none"> • Vamos desenvolver algumas primitivas para a biblioteca básica do simulador MVN na “linguagem de montagem” do simulador MVN, na forma de sub-rotinas • Objetivos: <ul style="list-style-type: none"> – Explorar os conceitos discutidos na aula anterior – Exercitar a linguagem de montagem apresentada nesta aula – Criar rotinas que serão usadas em aulas futuras desta disciplina • Para cada sub-rotina, construa um programa “main” correspondente para testá-la

DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS



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

Aula 02:
Introdução
Máq. von Neumann

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.1.0 ago. 2012

Exercícios (1)

TYGXXA02E01.asm (Entrega obrigatória)



- **PACK:** Compõe uma word a partir de seus bytes. O resultado está no acumulador.

Parâmetros: os dois endereços que contêm os bytes.
Retorno (acumulador): a palavra.


- Se os endereços que contêm os bytes (valores binários) forem denotados por B1 e B2, sendo:
 B1 = 00XY e B2 = 00ZT,
 a word resultante será XYZT (valor binário).

Endereço de início do programa principal: 0000
Endereço das words de entrada: 0002 e 0004
Endereço da word de saída: 0006

v.1.0 ago. 2012
3

DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS



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

Aula 02:
Introdução
Máq. von Neumann

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.1.0 ago. 2012

Exercícios (2)

TYGXXA02E02.asm (Entrega obrigatória)



- **UNPACK:** Extrai os bytes de uma word contida no acumulador colocando-os em dois endereços da memória.

Parâmetros: os dois endereços em que os bytes serão armazenados.
Retorno (acumulador): não.

- Se a word for denotada por XYZT (valores binários) e os endereços que conterão seus bytes forem denotados por B1 e B2, devemos ter:
 B1 = 00XY e B2 = 00ZT

Endereço de início do programa principal: 0000
Endereço da word de entrada: 0002
Endereço das words de saída: 0004 e 0006

v.1.0 ago. 2012
4

DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS

PCS

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

Aula 02:
Introdução
Máq. von Neumann

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.1.0 ago. 2012

Exercícios (3)

TYGXXA02E03.asm (Entrega obrigatória)

- **MEMCPY**: Copia uma sequência de tamanho arbitrário de bytes da memória de uma posição em outra. Retorna 0000 em caso de sucesso e FFFF em caso de erro (ex.: endereço de destino inválido).
Parâmetros: o número de caracteres a ser copiado, o endereço da sequência de origem e o endereço da sequência de destino.
Retorno (acumulador): 0000 em caso de sucesso; FFFF caso contrário.

Endereço de início do programa principal: 0000



Endereço do número de bytes a ser copiado: 0002

Endereço inicial da sequência de origem: 0004

Endereço inicial da sequência de destino: 0006

Endereço reservado para sequências de origem e destino: 0008..0028

5

DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS

PCS

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

Aula 02:
Introdução
Máq. von Neumann

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.1.0 ago. 2012

Exercícios (4a)

TYGXXA02E04.asm (Entrega opcional)

- **STRCMP**: Compara duas strings (sequências de caracteres), retornando o tamanho do prefixo em comum entre elas, ou seja, o número de caracteres mais à esquerda que são idênticos em ambas, sem contar o indicador de final de string.
Parâmetros: os endereços da primeira e da segunda strings.
Retorno (acumulador): o tamanho do prefixo em comum.

- Uma string segue a seguinte convenção: seu final é indicado por uma word que denota o valor nulo – seus bytes têm valor 0 (binário).

- Duas situações podem ocorrer: a string tem um número par ou um número ímpar de caracteres. Usaremos a seguinte convenção:

Número par: "abcd"



Número ímpar: "abc"

61	62	63	64	00	00
----	----	----	----	----	----

61	62	63	00	00	00
----	----	----	----	----	----

6

3

DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS

PCS

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

Aula 02:
Introdução
Máq. von Neumann

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.1.0 ago. 2012

Exercícios (4b)

TYGXXA02E04.asm (Entrega opcional)



- Para efeitos de teste, considere que as strings têm até 16 caracteres (8 words) cada.

Endereço de início do programa principal: 0000

Endereço da word de saída: 0002

Endereço da primeira string: 0004

Endereço da segunda string: 0014

DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS

PCS

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

Aula 02:
Introdução
Máq. von Neumann



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.1.0 ago. 2012

Exercícios: observações

- **Entrega:**
 - Em papel (**opcional**): solução em pseudo-linguagem de alto nível (ex.: similar a C) ou fluxograma.
 - Código fonte (**obrigatório**) devidamente comentado.
- **Dicas:**
 - Planejar a **divisão de tarefas** nas duplas: atenção para o tempo!
 - Para números positivos e negativos em complemento de 1:
 - **Multiplicar por 2** equivale a fazer um deslocamento para a **esquerda**;
 - **Dividir por 2** equivale a fazer um deslocamento para a **direita, carregando o bit de sinal**.

PCS
DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS
PCS 2302/2024
Laboratório de
Fundamentos da
Eng.de Computação

Aula 02:

Introdução

Máq. von Neumann

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.1.0 ago. 2012

Critério de correção

- Programas com os **nomes** pedidos e bem **comentados**
- Respeitou os **endereços de memória** pedidos
- Executou corretamente os seguintes **testes**:

Teste	Entrada PACK, Saída UNPACK	Entrada UNPACK, Saída PACK
E1_1, E2_1	/0012, /0034	/1234
E1_2, E2_2	/007F, /00FF	/7FFF
E1_3, E2_3	/0080, /0000	/8000
E1_4, E2_4	/0080, /0001	/8001
E1_5, E2_5	/00FA, /00FF	/FAFF
E1_6, E2_6	/00FA, /0010	/FA10

PCS
DEPARTAMENTO DE ENGENHARIA DE
COMUNICAÇÃO E SISTEMAS DIGITAIS
PCS 2302/2024
Laboratório de
Fundamentos da
Eng.de Computação

Aula 02:

Introdução

Máq. von Neumann

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.1.0 ago. 2012

Critério de correção

Teste	Entrada	Saída
E3_1	/0003, /0008, /0010: 0008: {/0001, /0002, /0003, /0004}, 0010: {/FFFF, /FFFF, /FFFF, /FFFF}	0010: {/0001, /0002, /0003, /FFFF}
E3_2	/0002, /0008, /0012: 0008: {/0001, /0002, /0003, /0004}, 0010: {/FFFF, /FFFF, /FFFF, /FFFF}	0010: {/FFFF, /0001, /0002, /FFFF}
E3_3	/0000: 0008: {/0001, /0002, /0003, /0004}, 0010: {/FFFF, /FFFF, /FFFF, /FFFF}	0010: {/FFFF, /FFFF, /FFFF, /FFFF}
E4_1	{'va, 'ic, 'om, 'fe, /0000} {'va, 'ic, 'om, 'fe, /0000}	0008
E4_2	{'va, 'ic, 'om, /6665, /0000} {'va, 'ic, 'om, /6600, /0000}	0007
E4_3	{'va, 'ic, 'om, 'fe, /0000} {'va, 'ic, 'oa, 'fe, /0000}	0005

  <p>DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E SISTEMAS DIGITAIS</p> <p>PCS 2302/2024 Laboratório de Fundamentos da Eng.de Computação</p> <p>Aula 02: Introdução Máq. von Neumann</p> <p>Autores: Anna H. R. Costa Jaime S. Sichman João José Neto Paulo S. Muniz Silva Ricardo L. A. Rocha</p> <p>Reestruturação: Paulo S. Muniz Silva</p> <p>v.1.0 ago. 2012</p>	Tabela de mnemônicos para as instruções da MVN (de 2 caracteres)			
	Operação 0 Jump Mnemônico JP	Operação 1 Jump if Zero Mnemônico JZ	Operação 2 Jump if Negative Mnemônico JN	Operação 3 Load Value 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 Subroutine Call Mnemônico SC	Operação B Return from Sub. Mnemônico RS
	Operação C Halt Machine Mnemônico HM	Operação D Get Data Mnemônico GD	Operação E Put Data Mnemônico PD	Operação F Operating System Mnemônico OS

  <p>DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E SISTEMAS DIGITAIS</p> <p>PCS 2302/2024 Laboratório de Fundamentos da Eng.de Computação</p> <p>Aula 02: Introdução Máq. von Neumann</p> <p>Autores: Anna H. R. Costa Jaime S. Sichman João José Neto Paulo S. Muniz Silva Ricardo L. A. Rocha</p> <p>Reestruturação: Paulo S. Muniz Silva</p> <p>v.1.0 ago. 2012</p>	Tabela de caracteres ASCII (7 bits. Ex.: "K" = 4b)								
		0	1	2	3	4	5	6	7
	0	NUL		SP	0	@	P	`	p
	1			!	1	A	Q	a	q
	2			"	2	B	R	b	r
	3			#	3	C	S	c	s
	4			\$	4	D	T	d	t
	5			%	5	E	U	e	u
	6			&	6	F	V	f	v
	7	BEL		'	7	G	W	g	w
	8			(8	H	X	h	x
	9)	9	I	Y	i	y
	a	LF		*	:	J	Z	j	z
	b		ESC	+	;	K	[k	{
	c			,	<	L	\	l	
	d	CR		-	=	M]	m	}
	e			.	>	N	^	n	~
	f			/	?	O	_	o	DEL

Lista de Comandos

- Para a execução do montador

- `java -cp MLR.jar montador.MvnAsm [<arquivo asm>]`
- **Exemplo:** `java -cp MLR.jar montador.MvnAsm test.asm`

- Para a execução da MVN

- `java -jar mvn.jar`
- **Obs.:** Se houver problemas com caracteres especiais, use:
 - `java -Dfile.encoding=cp850 -jar mvn.jar`