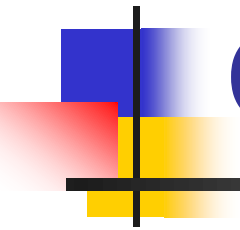


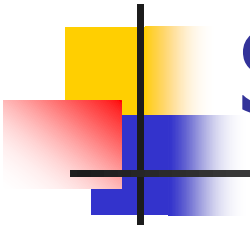
SCC0221 - Introdução à Ciência de Computação I

Prof.: Dr. Rudinei Goularte

(rudinei@icmc.usp.br)

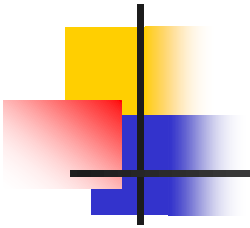
Aula 2 - Visão geral: computador e programação

Instituto de Ciências Matemáticas e de Computação - ICMC
Sala 4-229



Sumário

- 1. Sistemas de Numeração
- 2. Tipos de dados e informação
- 3. O Computador
- 4. Programação e Níveis de Linguagem



1. Sistemas de Numeração

- O sistema de numeração usual, o sistema decimal, é um sistema posicional.
- Usa-se dez dígitos 0, 1, 2, ..., 9, sendo que um número maior que 9 é representado usando uma convenção que atribui significado à posição ou lugar ocupado por um dígito.

1. Sistemas de Numeração

- No sistema decimal, o número **2562** tem a seguinte interpretação:

$2 \times 1000\ (10^3) = 2000$

$5 \times 100\ (10^2) = 500$

$6 \times 10\ (10^1) = 60$

$2 \times 1\ (10^0) = 2$

1. Sistemas de Numeração

- Sistema decimal
  - 10 dígitos: 0 1 2 3 4 5 6 7 8 9
  - Casas decimais: ... 10^3 10^2 10 1
- Sistema binário
  - 2 dígitos: 0 1
  - Casas binárias: .... 2^3 2^2 2 1
- Sistema Octal
  - 8 dígitos: 0 1 2 3 4 5 6 7
  - Casas octais: .... 8^3 8^2 8 1
- Sistema hexadecimal
  - 16 dígitos : 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - Casas hexadecimais: ... 16^3 16^2 16 1

1. Sistemas de Numeração

decimal	*octal*	*hexadecimal*	*binário*
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	10	8	1000
9	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111

Conversão binário-decimal

- No sistema binário de numeração a base é 2.
- Assim o valor do número 110101 é:

$1 \times 32\ (2^5) = 32$

$1 \times 16\ (2^4) = 16$

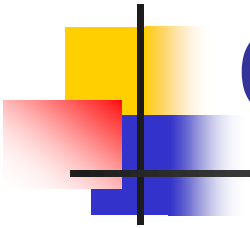
$0 \times 8\ \ (2^3) = 0$

$1 \times 4\ \ (2^2) = 4$

$0 \times 2\ \ (2^1) = 0$

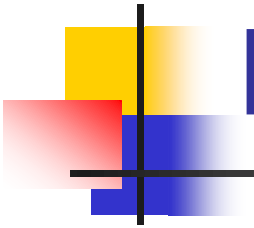
$1 \times 1\ \ (2^0) = 1$

$= 32 + 16 + 0 + 4 + 0 + 1 = 53$



Conversão decimal-binário

- Converta o número 53 (decimal) para seu respectivo valor binário.



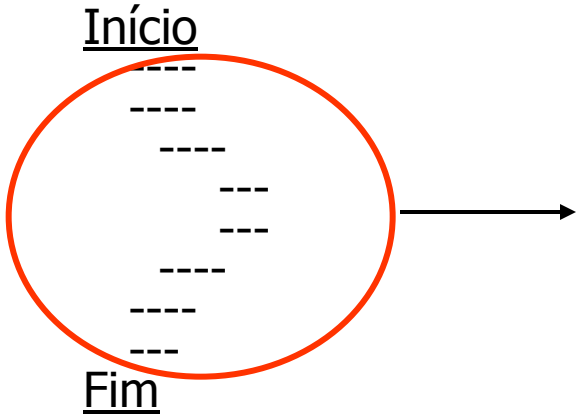
Para Pensar...

- E como fazer as conversões entre os outros sistemas?

2. Tipos de dados e informação

- Um algoritmo computacional deve ser uma seqüência de instruções manipulando dados.
  - **Instruções**: comandos que definem integralmente uma operação a ser executada. Determinam a forma pela qual os dados devem ser tratados.
  - **Dados**: elementos conhecidos de um problema. Podem ser recolhidos/fornecidos por diversos meios e serão processados pelo computador através das instruções.
  - **Informação:** Um conjunto **estruturado** de dados, transmitindo conhecimento.
  - Exemplos: média de notas e previsão do tempo.

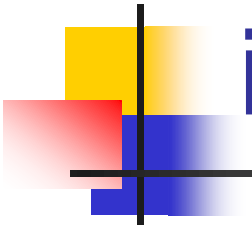
2. Tipos de dados e informação



Instruções manipulando dados. = Declarações Comandos Expressões.

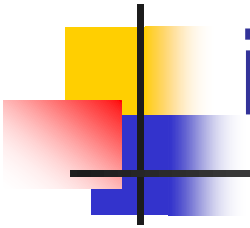
- Como representar os dados computacionalmente?

2. Tipos de dados e informação



- Os dados são computacionalmente representados através de constantes e variáveis, as quais possuem um tipo de dado associado.
  - Dizemos que uma determinada constante ou variável é de um determinado tipo.

2. Tipos de dados e informação

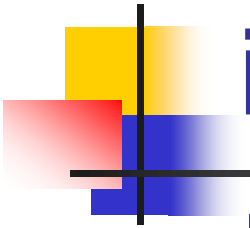


- Definição:

  Tipos de dados são métodos para interpretar o conteúdo da memória do computador. Definem o conjunto de valores que uma variável pode assumir.

- Um tipo de dado especifica:
  - A quantidade de memória que deve ser reservada para uma constante ou variável.
  - Como o dado representado na memória deve ser interpretado (o que significa a cadeia de bits).

2. Tipos de dados e informação



- As linguagens de programação conseguem manipular um conjunto de tipos de dados.
- Dentre eles, os tipos primitivos de dados (básicos) são classificados em dados literais (não numéricos), numéricos e lógicos.

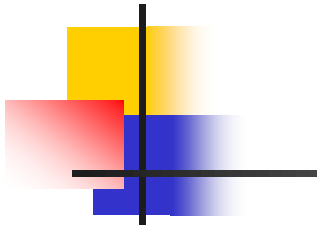
  - Como são organizados na memória do computador?

2. Tipos de dados e informação

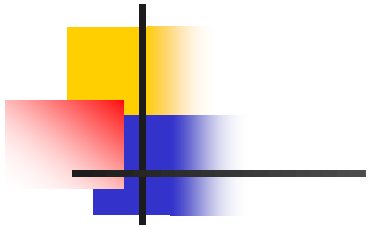
- Dados são organizados na memória através de sequências binárias, chamadas palavras.
- Uma palavra é uma unidade de informação de comprimento fixo n.
- Cada dígito binário é chamado de bit (**bi**nary digi**t**). É o componente básico da representação de dados.
  - O conjunto de 8 bits é chamado de byte.

2. Tipos de dados e informação

- Dados literias
  - São usados dois códigos padrões de caracteres de 8 bits:
    - EBCDIC (*Extended Binary Coded Decimal Interchange Code*) desenvolvido pela IBM, e
    - ASCII (*American Standard Code for Information Interchange*).

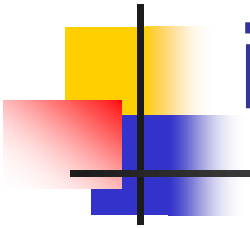


Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□



Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

2. Tipos de dados e informação



- Dados numéricos
  - Podem ser basicamente de dois tipos:
    - Inteiros: não possuem partes decimais. Ex.: 27 e 456
    - Reais: possuem partes decimais. Ex: 213,53 e 8,5

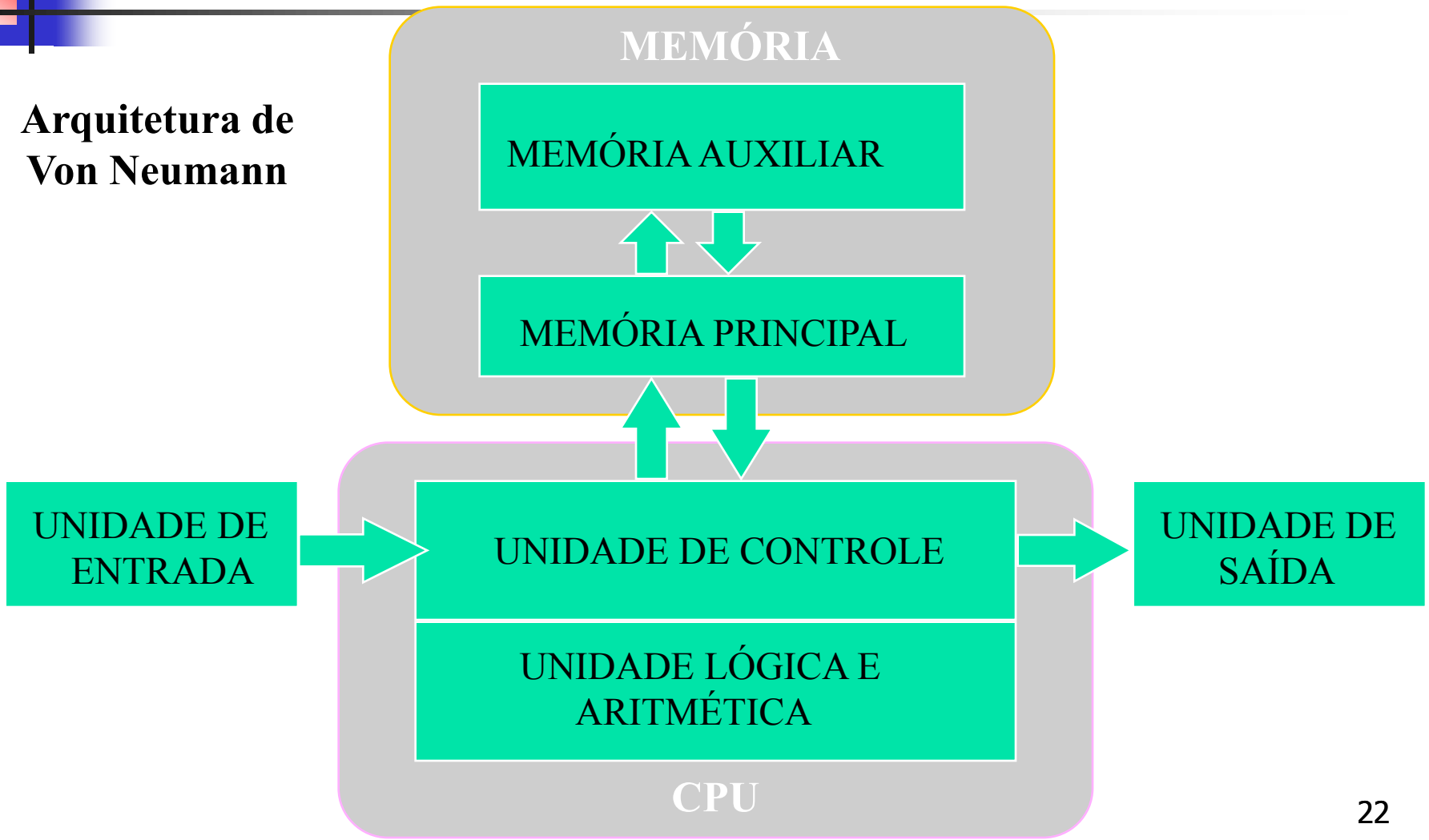
2. Tipos de dados e informação

- Dados lógicos
  - Utilizados para representar dois valores únicos: **verdadeiro** ou **falso** (**true** ou **false**)
    - Tipo de dado: lógico (boolean).
    - Também conhecido como tipo booleano, em referência à álgebra de Boole.

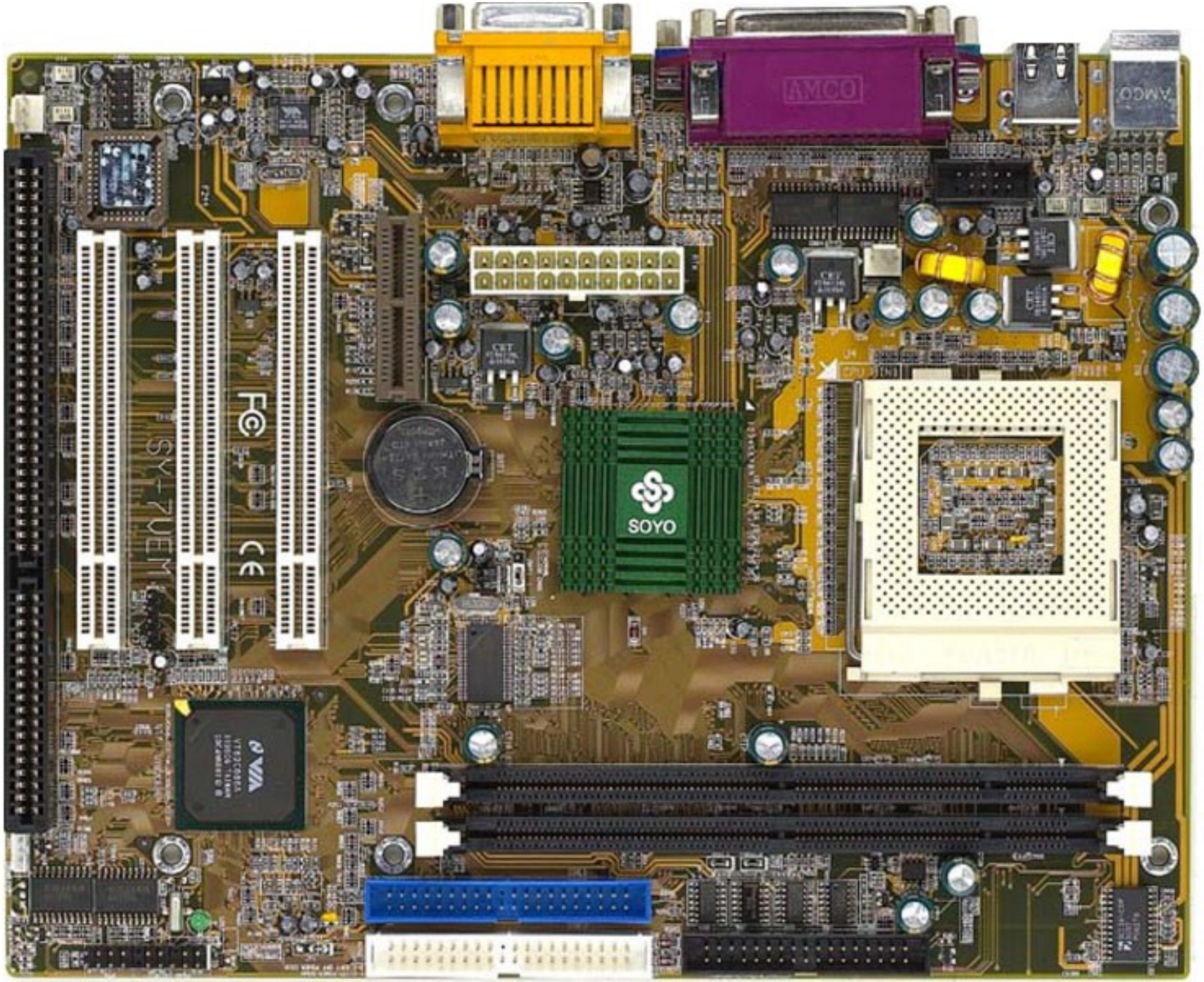
3. O Computador

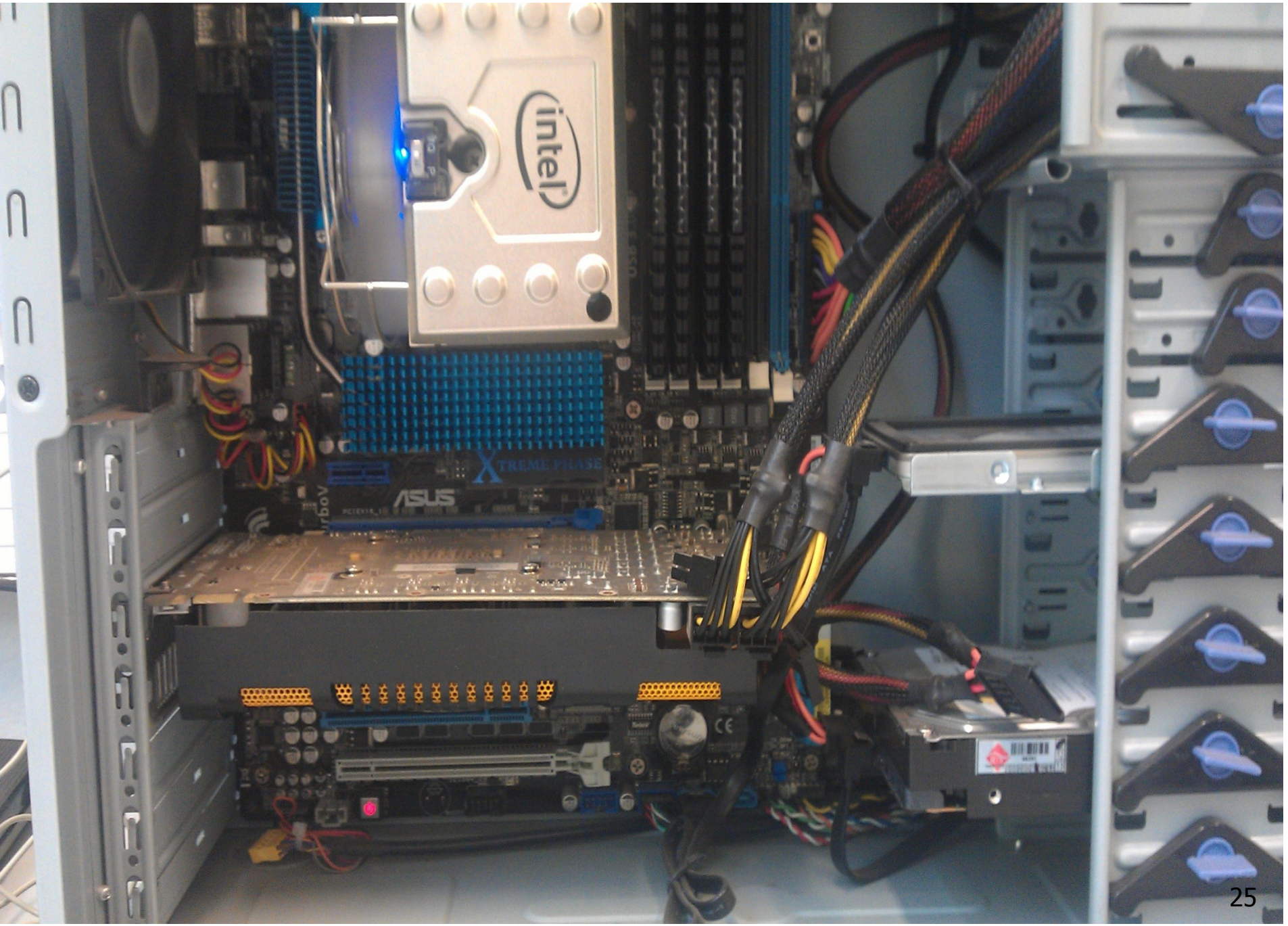
3. O Computador

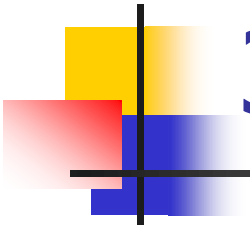
Arquitetura de Von Neumann



3. O Computador

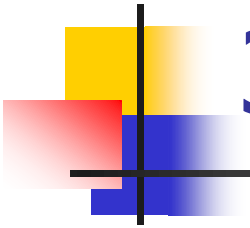

SOYO
SY-7VEM


intel
ASUS
X TREME PHASE



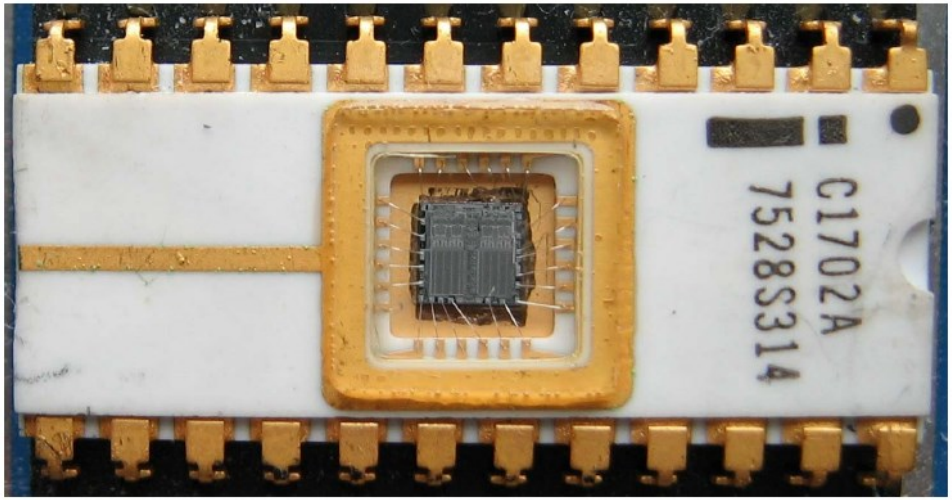
3. O Computador

- CPU (*Central Processing Unit*), ou Unidade de Processamento Central:
  - Responsável pelo gerenciamento de todas as funções do sistema.
  - Capaz de somar grandezas representadas por 0’s e 1’s e comparar grandezas. Para isto trabalha em velocidades altíssimas.
  - CPU (microprocessador): velocidade alcança 50.000 MIPS (Milhões de Instruções por Segundo).

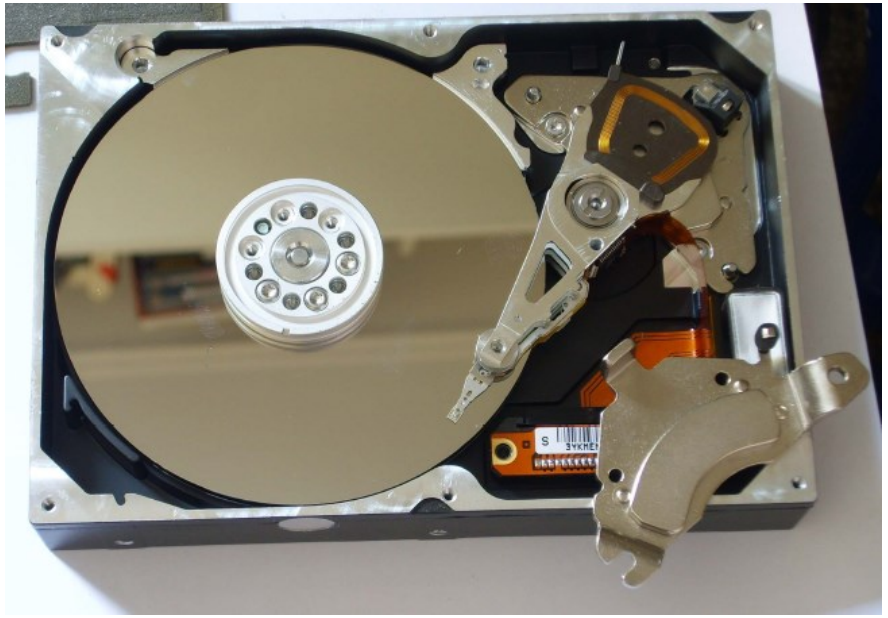


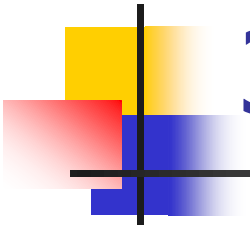
3. O Computador

- Memória
  - Principal
    - RAM – Random Access Memory (volátil)
    - ROM – Read Only Memory
  - Secundária
    - HDs – Hard Disks
    - CDs – Compact Disks
    - DVDs – Digital Versartile Disks
    - ...


C1702A
7528S314


elixir
M2U51264D89HC3G-5T
512MB DDR-400MHz CL3
PC3200U-30331
Warranty Void If Removed
0542.MN06A1107A.5YM0410N08.XX.CN

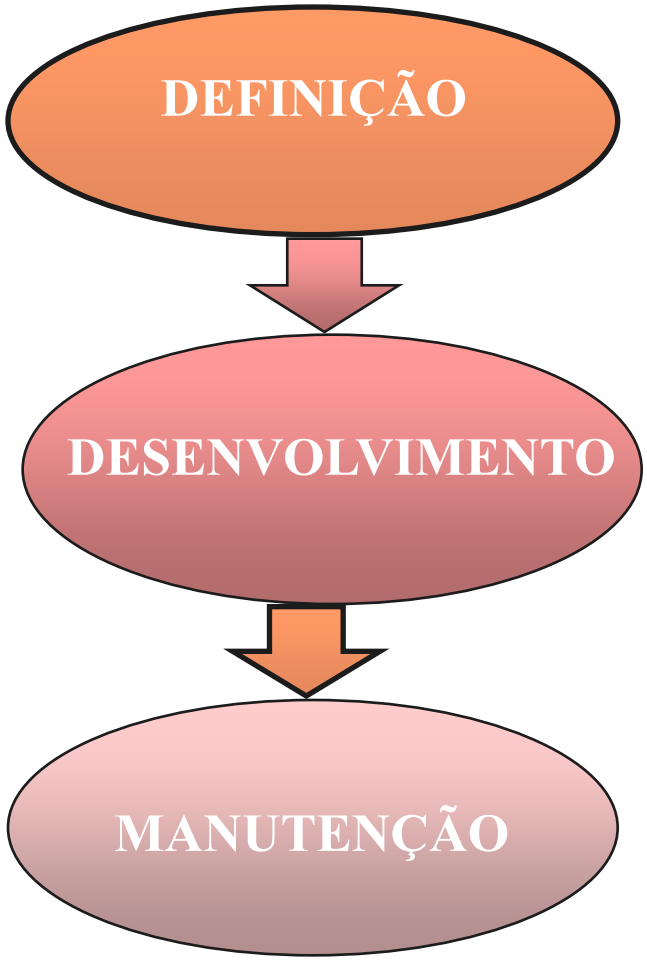




3. O Computador

- Memória – medidas de capacidade
  - K = Kilo (KB = Kilobytes = 2^{10} bytes)
  - M = Mega (MB = Megabytes = 2^{20} bytes)
  - G = Giga (GB = Gigabytes = 2^{30} bytes)
  - T = Tera (TB = Terabytes = 2^{40} bytes)
  - P = Peta (PB = Petabytes = 2^{50} bytes)
  - E = Exa (EB = Exabytes = 2^{60} bytes)
  - Z = Zetta (ZB = Zettabytes = 2^{70} bytes)
  - Y = Yotta (YB = Yottabytes = 2^{80} bytes)

4. Programação e Níveis de Linguagem



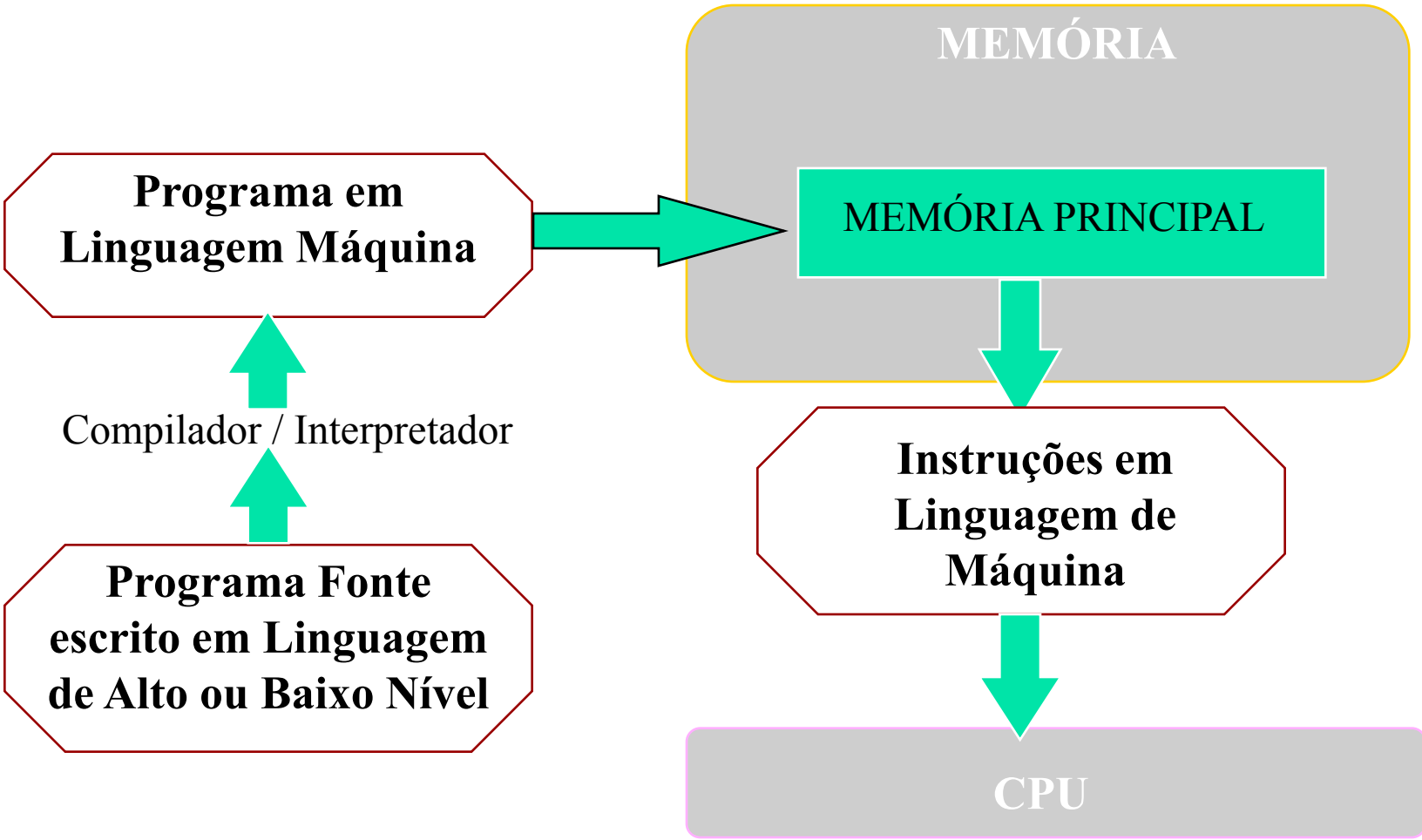
O QUE

COMO

ALTERAÇÕES

Etapas da Construção de Programas

4. Programação e Níveis de Linguagem



4. Programação e Níveis de Linguagem

- **Linguagem de máquina**
  - É o conjunto das instruções primitivas projetadas para um computador. Uma CPU somente pode compreender instruções que sejam expressas em termos de sua LINGUAGEM DE MÁQUINA.
  - Um programa escrito em linguagem de máquina consiste de uma série de números binários e é muito difícil de ser entendido pelas pessoas.
  - Exemplo: Cada instrução é constituída de 2 partes:

o código da operação e o operando

0010 (Load) 011000 (endereço)

0110 (multiplica) 000101 (valor)

...

0010011000

0110000101

...

4. Programação e Níveis de Linguagem

- Linguagem de baixo nível
  - Os programas são escritos em uma notação que está próxima da linguagem de máquina

Exemplo:

código da operação	operando	significado
LD	A	load A
MPI	5	multiplica 5

4. Programação e Níveis de Linguagem

- Linguagem de alto nível
  - Se pode escrever programas em uma notação próxima à maneira natural de expressar o problema que se deseja resolver.

    - Exemplo:
      RESULT = D-((A+B)/C)

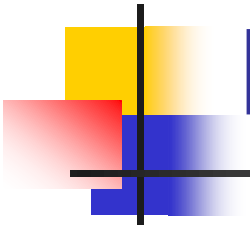
    - Aplicações Científicas : FORTRAN, ALGOL, BASIC, APL, LISP, PASCAL, ADA, C, PROLOG, PLI, Java, ...

    - Aplicações Comerciais: COBOL, RPG, PLI, C, Java, ...

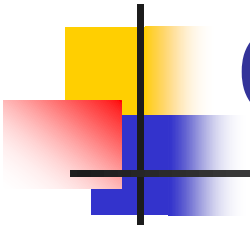
4. Programação e Níveis de Linguagem

- COMPILADOR
  - Traduz os comandos simbólicos de uma linguagem de alto nível, para linguagem de máquina.
- MONTADOR
  - Traduz os comandos simbólicos de uma linguagem de baixo nível , para linguagem de máquina.

4. Programação e Níveis de Linguagem



- INTERPRETADOR
  - Lê e executa uma declaração do programa por vez.
  - Nenhuma fase intermediária de compilação é necessária.
  - A execução do programa interpretado requer que o interpretador da linguagem esteja sendo executado no computador.



Créditos

- Agradecimentos à Profa. Dra. Rosely Sanches e ao Prof. Dr. João Luís G. Rosa.