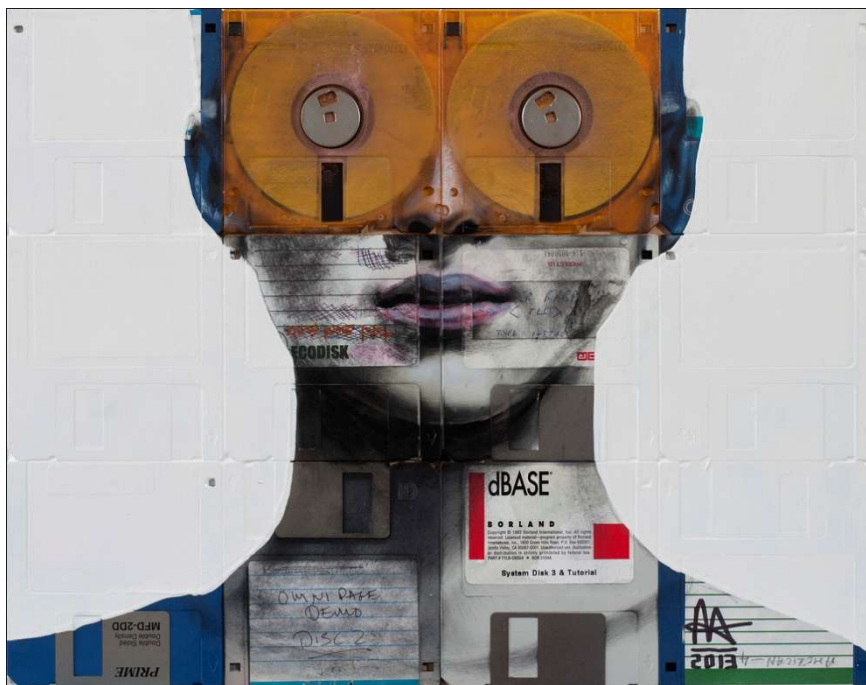


Algoritmos e Programação de Computadores

Disciplina 113476



<http://www.nickgentry.com/>

Prof. Alexandre Zaghetto
<http://alexandre.zaghetto.com>
zaghetto@unb.br

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

O presente conjunto de *slides* não pode ser reutilizado ou republicado sem a permissão do instrutor.

Módulo 02

Organização Básica de um Computador

1. Bases Numéricas

- **Sistemas Numéricos Posicionais**

$$1734 = 1 \times 1000 + 7 \times 100 + 3 \times 10 + 4 \times 1 = \\ 1 \times 10^3 + 7 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

$$5185.68 = 5 \times 10^3 + 1 \times 10^2 + 8 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$$

➤ Aqui, 10 é chamado de base do sistema de numeração.

1. Bases Numéricas

- **Sistemas Numéricos Posicionais**

- Em sistemas de numeração posicionais, a base pode ser qualquer número inteiro

$$r \geq 2.$$

- E o dígito na posição i (da direita para a esquerda) tem peso

$$r^i.$$

- A forma geral de um número em tal sistema de numeração é dada por

$$d_{p-1}d_{p-2}\dots d_1d_0, d_{-1}d_{-2}\dots d_{-n},$$

onde há p dígitos a esquerda da vírgula e n dígitos a direita.

1. Bases Numéricas

- **Sistemas Numéricos Posicionais**

➤ O valor do número é dado pela soma de cada dígito multiplicado pelo peso correspondente:

$$D = \sum_{i=-n}^{p-1} d_i \cdot r^i$$

➤ O dígito mais à esquerda é chamado de dígito **mais significativo** e o dígito mais à direita é chamado de dígito **menos significativo**.

1. Bases Numéricas

- **Números Binários**

➤ Como sistemas de computação manipulam sinais que podem se encontram em apenas uma entre duas possíveis condições:

- ✓ alto ou baixo;
- ✓ carregado ou descarregado;
- ✓ ligado ou desligado;
- ✓ aberto ou fechado;

esses sinais são interpretados como se fossem dígitos binários (*binary digits* ou *bits*), que podem apenas assumir um de dois possíveis valores:

- ✓ 0 ou 1.

1. Bases Numéricas

- **Números Binários**

➤ A forma geral de um número binário é:

$$b_{p-1}b_{p-2}\dots b_1b_0, b_{-1}b_{-2}\dots b_{-n},$$

e seu valor é dado por:

$$B = \sum_{i=-n}^{p-1} b_i \cdot 2^i.$$

➤ Quando se lida com números em diversas bases, utiliza-se um subscrito para indicar com que base se está trabalhando. Exemplos:

$$\begin{aligned} 100010_2 &= 34_{10} \\ 101,001_2 &= 5,125_{10} \end{aligned}$$

1. Bases Numéricas

• Números Octais

- Não são processados diretamente, mas podem ser úteis para documentação. O sistema de numeração octal tem base 8 (2^3) e pode ser útil na representação de números binários, pois sua base é uma potência de 2.
- Esse sistema de numeração tem 8 símbolos: 0, 1, 2, 3, 4, 5, 6, 7.
- Uma sequência de 3 bits pode assumir 8 possíveis valores, assim, dígitos octais podem ser utilizados para representar sequências de 3 bits.
- Exemplos:

$$\begin{aligned} 174_8 &= 124_{10} \\ 4450_8 &= 2344_{10} \end{aligned}$$



1. Bases Numéricas

- **Números Hexadecimais**

- O sistema de numeração hexadecimal tem base 16 (2^4) e pode ser útil na representação de números binários, pois sua base também é um potência de 2.
- Esse sistema de numeração tem 16 símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
- Uma seqüência de 4 bits pode assumir 16 possíveis valores, assim, dígitos hexadecimais podem ser utilizados para representar seqüências de 4 bits.
- Exemplos:

$$\begin{aligned} EB_{16} &= 235_{10} \\ ABCD_{16} &= 43981_{10} \end{aligned}$$



1. Bases Numéricas

- **Números Hexadecimais**

- O sistema de numeração hexadecimal é freqüentemente utilizado para representar endereços de memória.
- Muitas linguagens de programação usam o prefixo "0x" para indicar que o número está escrito em hexadecimal como, por exemplo, 0xABCD.

1. Bases Numéricas

- Decimal x Binária x Octal x Hexadecimal

Base-10	Base-2	Base-8	Base-16
Decimal	Binário	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2. Conversão entre Bases Numéricas

- Conversão de base qualquer para decimal
 - Para se converter um número de uma base qualquer para a base decimal, utiliza-se a seguinte expressão:

$$D = \sum_{i=-n}^{p-1} d_i \cdot r^i$$

- Binário para decimal:

$$10011_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19_{10}$$

$$100010_2 = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 34_{10}$$

2. Conversão entre Bases Numéricas

- Conversão de base qualquer para decimal

➤ Octal para decimal:

$$436_8 = 4 \times 8^2 + 3 \times 8^1 + 6 \times 8^0 = 286_{10}$$

$$1357_8 = 1 \times 8^3 + 3 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 = 751_{10}$$

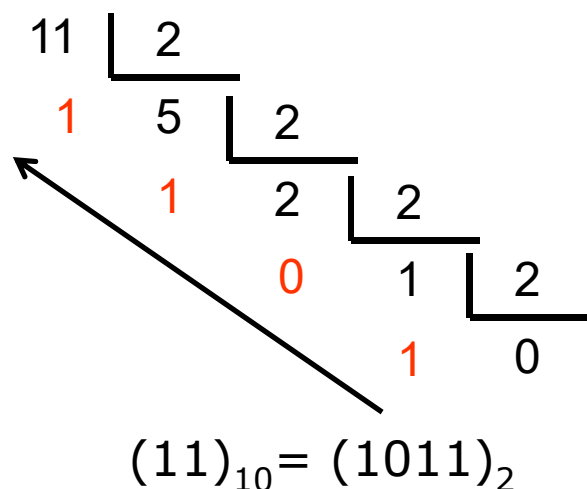
➤ Hexadecimal para decimal

$$A3_{16} = 10 \times 16^1 + 3 \times 16^0 = 163_{10}$$

$$ABCD_{16} = 10 \times 16^3 + 11 \times 16^2 + 12 \times 16^1 + 13 \times 16^0 = 43981_{10}$$

2. Conversão entre Bases Numéricas

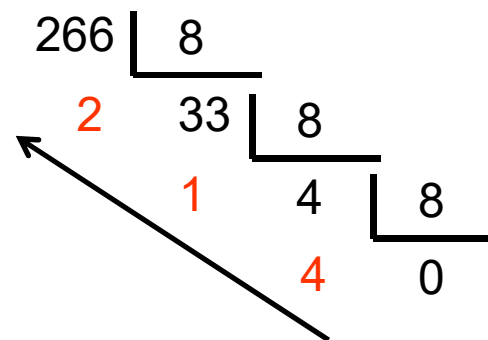
- Conversão de decimal para base qualquer
 - Dividir sucessivamente pela base o número decimal e os quocientes que vão sendo obtidos, até que o quociente de uma das divisões seja 0.
 - Decimal para binário



2. Conversão entre Bases Numéricas

- Conversão de decimal para base qualquer

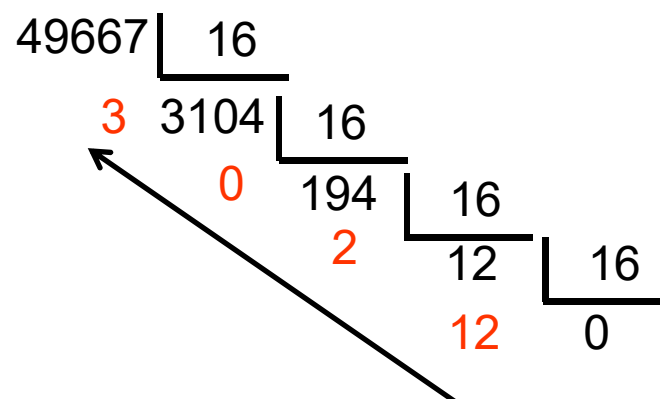
➤ Decimal para octal



$$(266)_{10} = (412)_8$$

2. Conversão entre Bases Numéricas

- Conversão de decimal para base qualquer
 - Decimal para hexadecimal



$$(49667)_{10} = (C203)_{16}$$

2. Conversão entre Bases Numéricas

- Conversão octal \leftrightarrow binário

(100	011	010	001) ₂
(4	3	2	1) ₈

(7	2	6	1) ₈
(111	010	110	001) ₂

2. Conversão entre Bases Numéricas

- Conversão hexadecimal \leftrightarrow binário

(1000	0111	0100	0010) ₂
(8	7	4	2) ₁₆

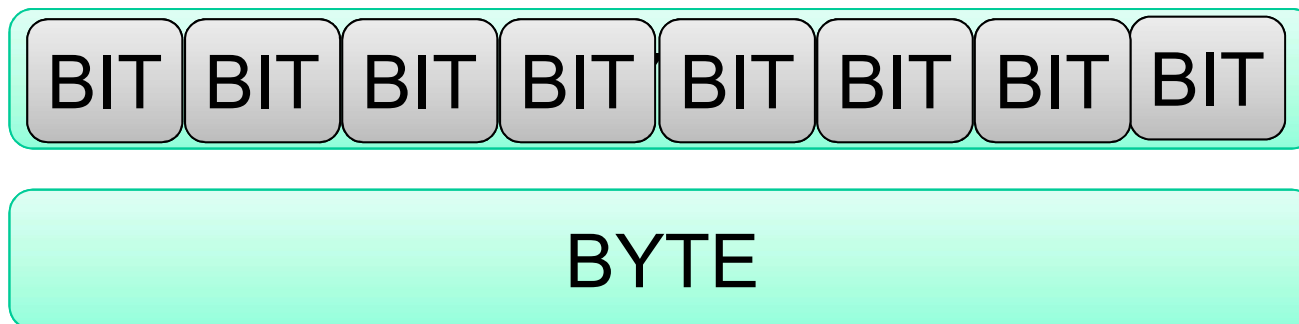
(9	D	8	F) ₁₆
(1001	1101	1000	1111) ₂

3. Unidades de Medida

- Em seu nível mais baixo, tudo (letras, algarismos, sinais de pontuação, símbolos, comandos) no computador é representado por dígitos binários.

BIT

- Embora a unidade fundamental de informação do computador seja o *bit*, na prática também utilizamos o bytes e seus múltiplos.



3. Unidades de Medida

- Bits, bytes e seus múltiplos são utilizados para quantificar capacidade de armazenamento.

bit	(b)		
Quilobit	(kb)	= 1024 bits	= 2^{10} bits
Megabit	(Mb)	= 1024 kb	= 2^{20} bits
Gigabit	(Gb)	= 1024 Mb	= 2^{30} bits
Terabit	(Tb)	= 1024 Gb	= 2^{40} bits
Petabit	(Pb)	= 1024 Tb	= 2^{50} bits
Exabit	(Eb)	= 1024 Pb	= 2^{60} bits
Zettabit	(Zb)	= 1024 Eb	= 2^{70} bits
Yottabit	(Yb)	= 1024 Zb	= 2^{80} bits

3. Unidades de Medida

- Bits, bytes e seus múltiplos são utilizados para quantificar capacidade de armazenamento.

Byte	(B)	= 8 bits	
Quilobyte	(kB)	= 1024 Bytes	= 2^{10} Bytes
Megabyte	(MB)	= 1024 kB	= 2^{20} Bytes
Gigabyte	(GB)	= 1024 MB	= 2^{30} Bytes
Terabyte	(TB)	= 1024 GB	= 2^{40} Bytes
Petabyte	(PB)	= 1024 TB	= 2^{50} Bytes
Exabyte	(EB)	= 1024 PB	= 2^{60} Bytes
Zettabyte	(ZB)	= 1024 EB	= 2^{70} Bytes
Yottabyte	(YB)	= 1024 ZB	= 2^{80} Bytes

- O que vocês acham desses prefixos?

3. Unidades de Medida

- Atenção – Na verdade...
 - ✓ Os prefixos no SI referem-se exclusivamente à potências de 10.
 - ✓ Há, inclusive, uma nota na 8ª edição que cita explicitamente o caso dos bits:

"These SI prefixes refer strictly to powers of 10. They should not be used to indicate powers of 2 (for example, one kilobit represents 1000 bits and not 1024 bits) ."

3. Unidades de Medida

- Atenção – Na verdade...

✓ International Electrotechnical Commission (IEC) - IEC 60027-2:

bit	(bit)		
Kibibit	(Kibit)	= 1024 bits	= 2^{10} bits
Mebibit	(Mibit)	= 1024 Kibit	= 2^{20} bits
Gibibit	(Gibit)	= 1024 Mibit	= 2^{30} bits
Tebibit	(Tibit)	= 1024 Gibit	= 2^{40} bits
Pebibit	(Pibit)	= 1024 Tibit	= 2^{50} bits
Exbibit	(Eibit)	= 1024 Pibit	= 2^{60} bits
Zebibit	(Zibit)	= 1024 Eibit	= 2^{70} bits
Yobibit	(Yibit)	= 1024 Zibit	= 2^{80} bits

3. Unidades de Medida

- Atenção – Na verdade...

✓ International Electrotechnical Commission (IEC) - IEC 60027-2:

Byte	(B)	= 8 bits	
Kibibyte	(KiB)	= 1024 Bytes	= 2^{10} Bytes
Mebibyte	(MiB)	= 1024 KiB	= 2^{20} Bytes
Gibibyte	(GiB)	= 1024 MiB	= 2^{30} Bytes
Tebibyte	(TiB)	= 1024 GiB	= 2^{40} Bytes
Pebibyte	(PiB)	= 1024 TiB	= 2^{50} Bytes
Exbibyte	(EiB)	= 1024 PiB	= 2^{60} Bytes
Zebibyte	(ZiB)	= 1024 EiB	= 2^{70} Bytes
Yobibyte	(YiB)	= 1024 ZiB	= 2^{80} Bytes



4. Codificação

- Em seu nível mais baixo, tudo (letras, algarismos, sinais de pontuação, símbolos, comandos) no computador é representado por dígitos binários.
- Exemplo: ASCII (*American Standard Code Information Interchange*).
 - ASCII: utiliza 7 *bits* de um *Byte* para representar caracteres.
 - ASCII estendida: utiliza 8 *bits* de um *Byte* para representar caracteres.

4. Codificação

- ASCII:

Tabela ASCII: Códigos de Controle (0..31):			
Cód	Caractere	Cód	Caractere
0	NULL (nulo)	1	SOH (Start of Heading / Início de cabeçalho)
2	STX (Start of TeXt / Início de Texto)	3	ETX (End of TeXt / fim de texto)
4	EOT (End Of Transmission / fim de transmissão)	5	ENQ (ENQuiry / inquirição, consulta)
6	ACK (ACKnowledge / confirmação, entendido)	7	BEL (BELL, BEEP / Campainha)
8	BS (Backspace / retorno de 1 caractere)	9	HT (Horizontal Tab / Tabulação horizontal)
10	LF (Line Feed / alimentação, mudança de linha)	11	VT (Vertical Tab / Tabulação vertical)
12	FF (Form Feed / Alimentação de formulário)	13	CR (Carriage Return / retorno ao início da linha)
14	SO (Serial Out / Saída Serial) (Shift Out / deslocamento para fora)	15	SI (Serial In / Entrada Serial) (Shift In / deslocamento para dentro)
16	DLE (Data Link Escape / escape de conexão)	17	DC1/XON (Device Control1/control de dispositivo1)
18	DC2 (Device Control 2 / controle de dispositivo2)	19	DC3/XOFF (Device Control3/control de dispositivo3)
20	DC4 (Device Control 4 / controle de dispositivo4)	21	NAK (Negative AcKnowledge / confirmação negativa)
22	SYN (SYNchronous Idle / espera síncrona)	23	ETB (End Transm. Block/bloco de fim de transmissão)
24	CAN (Cancel / cancelamento)	25	EM (End of Media / Fim do meio ou mídia)
26	SUB (SUBstitute, substituir)	27	ESC (ESCape / escape)
28	FS (File Separator / Separador de arquivo)	29	GS (Group Separator / separador de grupo)
30	RS (Request to Send, Record Separator / requisição de envio, separador de registro)	31	US (Unit Separator / separador de unidade)

4. Codificação

- ASCII:

Tabela ASCII: Caracteres-padrão (32..127):

Cód	Carac	Cód	Carac	Cód	Carac	Cód	Carac	Cód	Carac	Cód	Carac
32	<espaço>	33	!	34	"	35	#	36	\$	37	%
38	&	39	'	40	(41)	42	*	43	+
44	,	45	-	46	.	47	/	48	0	49	1
50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=
62	>	63	?	64	@	65	A	66	B	67	C
68	D	69	E	70	F	71	G	72	H	73	I
74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U
86	V	87	W	88	X	89	Y	90	Z	91	[
92	\	93]	94	^	95	_	96	`	97	a
98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m
110	n	111	o	112	p	113	q	114	r	115	s
116	t	117	u	118	v	119	w	120	x	121	y
122	z	123	{	124		125	}	126	~	127	<delete>

A → 1000001₂

B → 1000010₂

C → 1000011₂

4. Codificação

- ASCII estendida (pode variar):

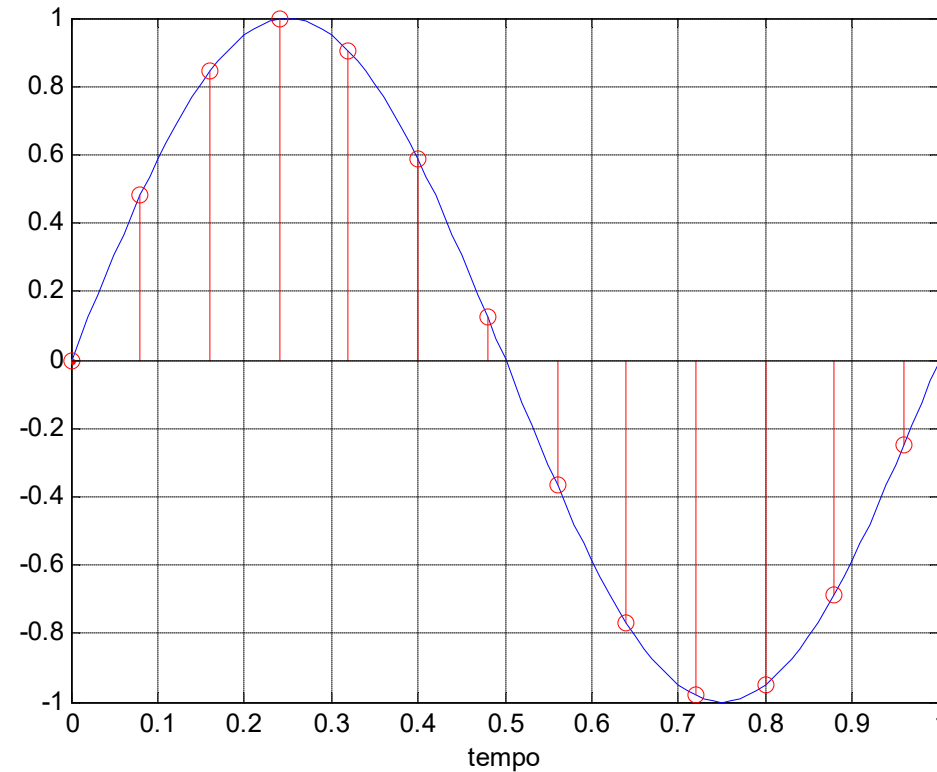
Tabela ASCII: Caracteres Estendidos (128..255):

Cód	Carac	Cód	Carac	Cód	Carac	Cód	Carac	Cód	Carac	Cód	Carac
128	Ç	129	ü	130	é	131	â	132	ä	133	à
134	ã	135	ç	136	ê	137	ë	138	è	139	ï
140	î	141	ì	142	Ä	143	Å	144	É	145	æ
146	Æ	147	ô	148	ö	149	ò	150	û	151	ù
152	ÿ	153	Ö	154	Ü	155	ø	156	£	157	Ø
158	×	159	ƒ	160	á	161	í	162	ó	163	ú
164	ñ	165	Ñ	166	ª	167	º	168	¿	169	®
170	¬	171	½	172	¼	173	¡	174	«	175	»
176	⋈	177	⋉	178	⋊	179		180	¡	181	Á
182	Â	183	Ã	184	©	185	ƒ	186		187	⌋
188	„	189	¢	190	¥	191	¬	192	⌌	193	⌑
194	⌒	195	⌓	196	—	197	+	198	ã	199	Ã
200	⌕	201	⌖	202	⌗	203	⌘	204	⌙	205	=
206	⌚	207	⌛	208	δ	209	Ð	210	Ê	211	Ë
212	Ë	213	¬	214	Í	215	Î	216	Ï	217	⌐
218	⌐	219	■	220	■	221	¡	222	Ì	223	■
224	Ó	225	β	226	Ô	227	Ò	228	õ	229	Õ
230	Μ	231	ρ	232	þ	233	Ú	234	Û	235	Ù
236	ý	237	Ý	238	—	239	’	240	-	241	±
242	≡	243	¾	244	¶	245	§	246	÷	247	,
248	ο	249	“	250	.	251	₁	252	₃	253	₂
254	■	255									

4. Codificação

- Red Book (padrão para CD de áudio)

- 44.1 kHz
- 16 *bits*
- Estéreo



4. Codificação

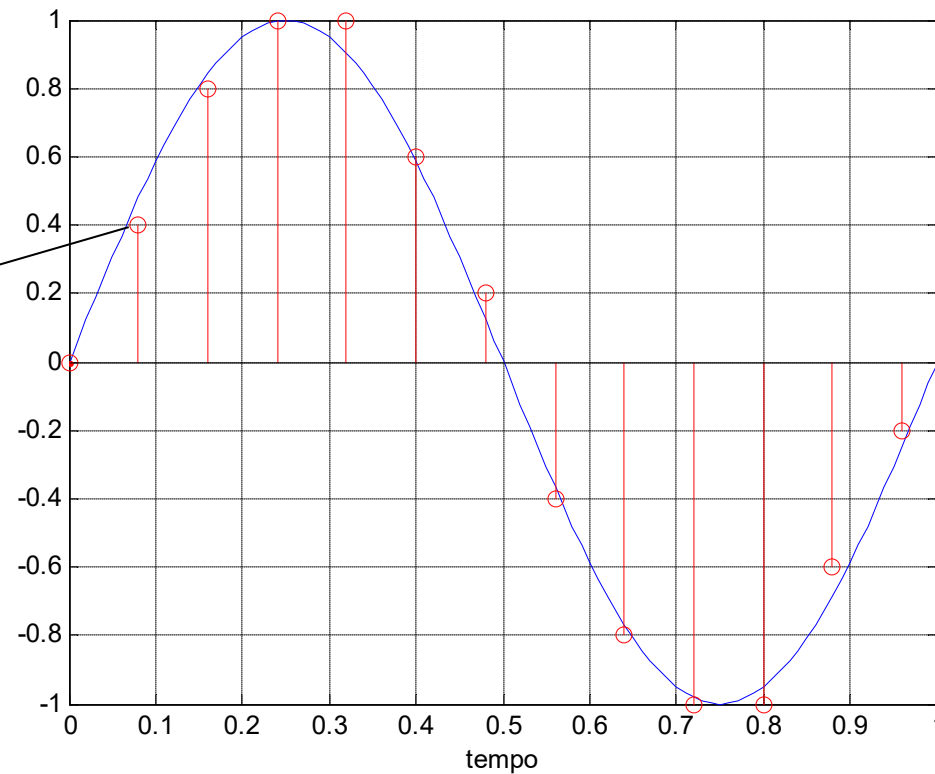
- Red Book (padrão para CD de áudio)

- 44.1 kHz
- 16 *bits*
- Estéreo

✓ Uma amostra a
Cada $1/44100$ s

✓ 16 *bits*/amostra

✓ 2 canais →
1.411.200 bits/ segundo



4. Codificação

- BMP ou *Bitmap* (formato de arquivo de imagem)
 - 8 *bits* (1 *Byte*) /amostra (*pixel*)/plano
 - 3 planos (RGB)

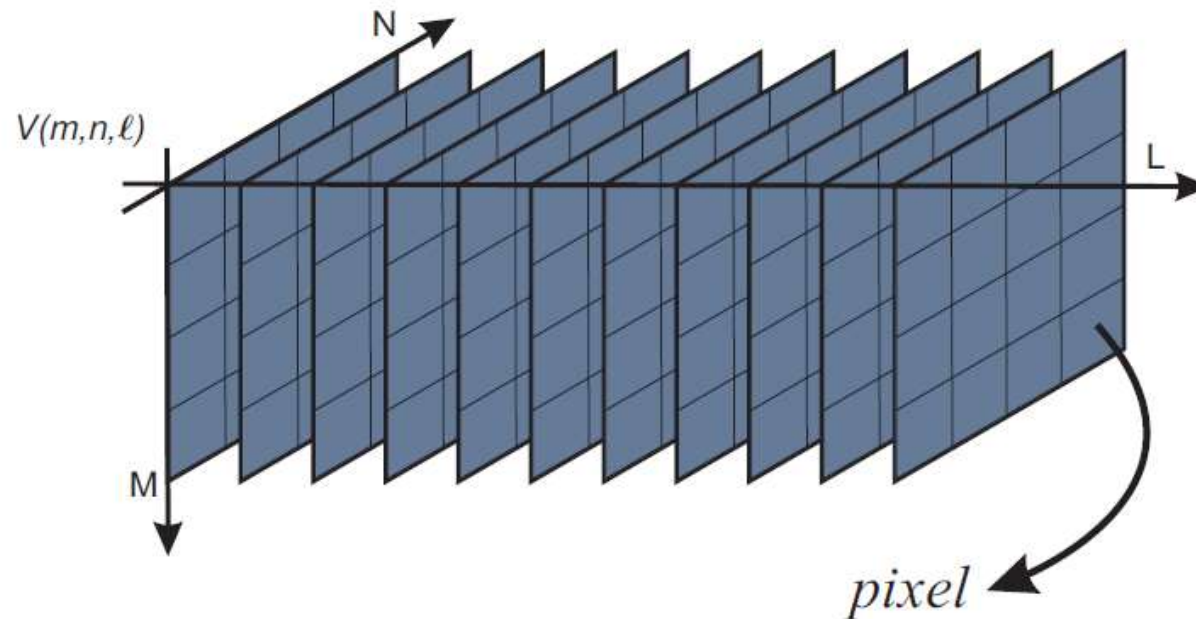


4. Codificação

- BMP ou *Bitmap* (formato de arquivo de imagem)
 - 8 *bits* (1 *Byte*) /amostra (*pixel*)/plano
 - 3 planos (RGB)
 - Imagem de 1920 x 1080:
 - ✓ No. de pixels: $1920 \times 1080 = 2073600$ *pixels*
 - ✓ No. de bytes/*pixel* = 3 Bytes
 - ✓ No. total de Bytes = 6220800 Bytes (aprox. 6 MB)

4. Codificação

- Vídeo sem compressão (uma possível configuração)
 - 8 *bits* (1 *Byte*) /amostra (*pixel*)/plano
 - 3 planos (RGB) → 1 quadro
 - 30 quadros/s

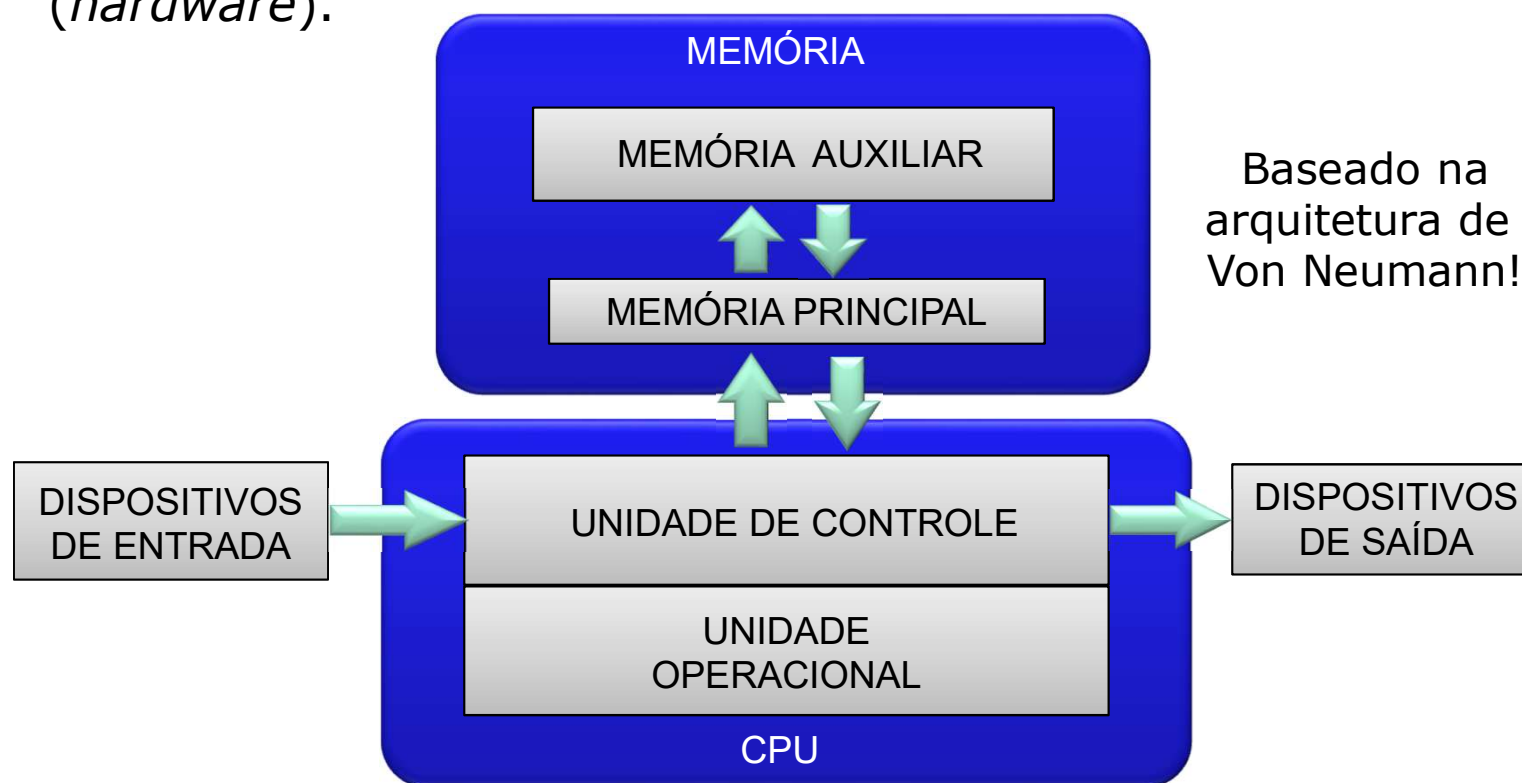


4. Codificação

- Vídeo sem compressão (uma possível configuração)
 - 8 *bits* (1 *Byte*) /amostra (*pixel*)/plano
 - 3 planos (RGB) → 1 quadro
 - 30 quadros/s
 - Duas horas de vídeo:
 - ✓ Resolução de um quadro:
288x352 *pixels*
 - ✓ No. de *pixels*/quadro:
101376 *pixels*
 - ✓ No. de Bytes/quadro (3 planos - RGB):
 $3 \times 101376 = 304128$ Bytes
 - ✓ No. de quadros em 2 h:
 $2 \times 3600 \times 30 = 216000$ quadros
 - ✓ No. total Bytes:
 216000×304128 Bytes (aprox. 62 GB).

5. Componentes Básicos de um Computador

- Um computador é composto por blocos convencionalmente chamados de: **Memória**, **Unidade Operacional**, **Unidade de controle** e **Dispositivos de entrada e saída** (*hardware*).



6. Sistema Operacional

- Cria um ambiente onde os usuários podem desenvolver seus programas (*softwares*) e executá-los sem se preocupar com detalhes de *hardware*.
- É um conjunto de programas que desempenham rotinas necessárias ao funcionamento do computador, tais como:
 - a. gerenciamento da memória;
 - b. administração dos dados;
 - c. acionamento dos dispositivos;
 - d. execução de programas utilitários.

6. Sistema Operacional

- <http://www.w3counter.com/globalstats.php> (Março 2018)

Top 10 Platforms		
1	Windows 10	17.00%
2	Windows 7	16.77%
3	Android 7	11.51%
4	iOS 11	10.88%
5	Android 6	9.69%
6	Android 5	8.01%
7	Android 4	4.84%
8	Mac OS X	4.36%
9	Windows 8.1	3.17%
10	iOS 10	2.95%



7. Linguagem de Programação

- É um conjunto de termos (vocabulário) e regras (sintaxe) que permitem a formulação de instruções a um computador.
- Permite construir programas.
- Existem várias linguagens diferentes, cada uma com recursos que facilitam aplicações específicas.

Para um programador é mais importante compreender os fundamentos e técnicas da programação do que dominar esta ou aquela linguagem.

7. Pseudo-linguagem de Programação

- É uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples sem a necessidade de se conhecer alguma linguagem de programação.

9. Compilação

- É o processo de tradução de um programa escrito em linguagem de alto nível para código em linguagem de máquina.

10. Interpretação

- Cada comando é lido, verificado, convertido em código executável e imediatamente executado, antes que o comando seguinte seja sequer lido.

“A ciência normal, atividade que consiste em solucionar quebra-cabeças, é um empreendimento altamente cumulativo, extremamente bem sucedido no que toca ao seu objetivo, a ampliação contínua do alcance e da precisão do conhecimento científico. Contudo, falta aqui um produto comum do empreendimento científico. A ciência normal não se propõe descobrir novidades no terreno dos fatos ou da teoria; quando é bem sucedida, não as encontra.”

*Thomas S. Kuhn, Físico Teórico,
em seu livro A Estrutura das Revoluções Científicas.*