

# SISTEMAS MULTIMÍDIA

## TEXTO 1

Prof.: Danilo Coimbra  
([coimbra.danilo@ufba.br](mailto:coimbra.danilo@ufba.br))



# Texto

2

- Texto é o modo mais adequado para transmitir informação de modo preciso
  - ▣ Semântica
  
- Portanto, é o modo dominante de interação/comunicação entre o usuário e dispositivos computacionais

# Texto

3

- É constituído por caracteres abstratos que se agrupam em alfabetos
- Cada idioma(s), possui um alfabeto
  - ▣ Texto digital: caracteres abstratos + ..?
- Texto digital
  - ▣ Define o mapeamento entre os caract. abstratos de um alfabeto e os valores que podem ser armazenados no computador

# Texto

4

- Representação de textos
  - ▣ É uma sequência de caracteres

"O	L	A"
79	76	65
1001111	1001100	1000001

Código ASCII

Hello World!!

Links in context.



# Representação de Texto

5

- Na área de IHM(Interação Homem-Máquina), o conteúdo textual pode assumir três formas:

- **Não Formatado** (*plaintext*)

- Texto puro que contém caracteres de controle, nova linha, espaço, tabulação, .. Dimensão fixa, limitado.

- **Formatado** (*richtext*)

- Permite a criação de páginas de cadeias caracteres com diferentes estilos, tamanhos, formas, cores, incluindo tabela, gráficos, animação,...

- **Hipertexto** (*hypertext*)

- Documentos que são interligados entre eles por meio de hiperlinks

# Representação de Texto

6

## ❑ Não formatado: ASCII (exemplo)

Bit positions	7	0	0	0	0	1	1	1	1
	6	0	0	1	1	0	0	1	1
	5	0	1	0	1	0	1	0	1
4 3 2 1									
0 0 0 0	NUL	DLE	SP	0	@	P	\	p	
0 0 0 1	SOH	DC1	!	1	A	Q	a	q	
0 0 1 0	STX	DC2	"	2	B	R	b	r	
0 0 1 1	ETX	DC3	#	3	C	S	c	s	
0 1 0 0	EOQ	DC4	\$	4	D	T	d	t	
0 1 0 1	ENQ	NAK	%	5	E	U	e	u	
0 1 1 0	ACK	SYN	&	6	F	V	f	v	
0 1 1 1	BEL	ETB	'	7	G	W	g	w	
1 0 0 0	BS	CAN	(	8	H	X	h	x	
1 0 0 1	HT	EM	)	9	I	Y	i	y	
1 0 1 0	LF	SUB	*	:	J	Z	j	z	
1 0 1 1	VT	ESC	+	;	K	[	k	{	
1 1 0 0	FF	FS	,	<	L	\	l		
1 1 0 1	CR	GS	-	=	M	]	m	}	
1 1 1 0	SO	RS	.	>	N	^	n	~	
1 1 1 1	SI	US	/	?	O	_	o	DEL	

(**American Standard Code for Information Interchange**)

Caractere representado por 7 bits (128 combinações)

Ex. Caractere "M": 1001101

Caracteres "imprimíveis" e "de controle"

Ex. BS (backspace), SP (space), DEL, ESC, FF (form feed), FS (file separator), ACK, etc.

# Representação de Texto

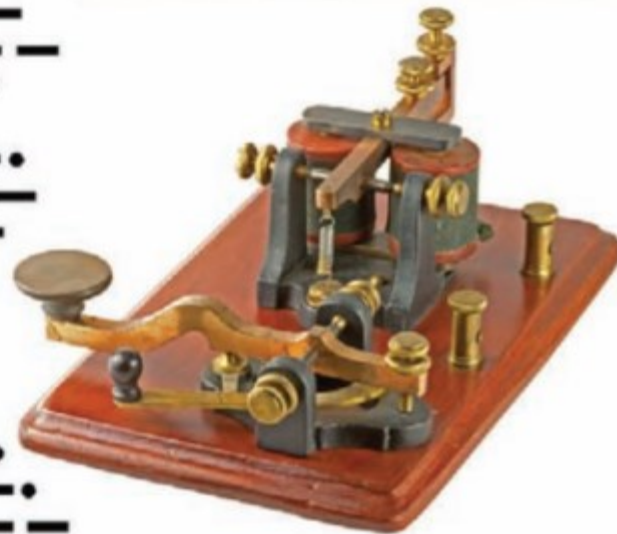
7

- ❑ Não formatado: Telégrafo com código Morse (exemplo)

A	• —	V	• • • —
B	— • • •	W	— — —
C	— • — •	X	— • • —
D	— • •	Y	— • — —
E	•	Z	— — • •
F	• • — •	.	• — — — • —
G	— — — •	,	— — — • — —
H	• • • •	?	• • — — • •
I	• •	/	— • • — •
J	• — — — —	@	• — — • • •
K	— • — —	1	• — — — —
L	• — • •	2	• • — — —
M	— — —	3	• • • — —
N	— •	4	• • • •
O	— — — —	5	• • • • •
P	• — — — •	6	— • • • •
Q	— — — • —	7	— — • • •
R	• — — •	8	— — — • •
S	• • •	9	— — — — •
T	— —	0	— — — — —
U	• • —		

The Message  
When does the stagecoach depart?

The Transmission in Morse Code  
..... / ..... / ..... / ..... /  
... .. / ..... / ..... / ..... /

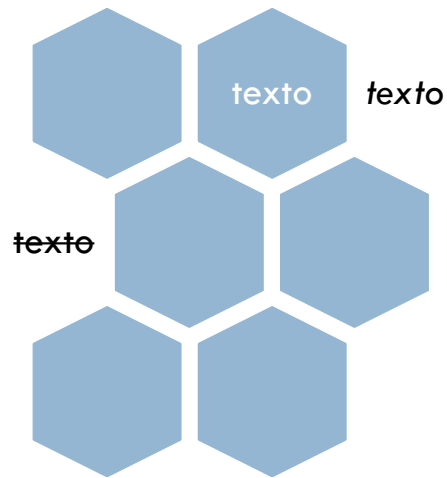


# Representação de Texto

8

## □ Formatado

**Hello** World!!





# Representação de Texto

9

## □ Hipertexto

### Senado rejeita decisão do Supremo e devolve mandato a Aécio

- Placar da sessão foi de 44 a 26; saiba como votou cada senador
- Sadi: ação para salvar Aécio foi coordenada pessoalmente por Temer
- Oposição vê 'corporativismo', e aliados citam respeito à Constituição



City mantém 100%;  
Gabriel Jesus marca e  
Ederson pega pênalti

- Real empata com Tottenham
- Liverpool massacra Maribor

<	ENCERRADO	L. dos Campeões	L. dos Campeões	L. dos Campeões	18/10	L. dos Campeões	L. dos Campeões	L. dos Campeões	L. dos Campeões	>	MAIS JOGOS
	MAR LIV	0 7	MAC NAP	2 1	MAD TOT	1 1	AND PSG	BAR OLY	CHÉ ROM	BEN MAN	
LIGA DOS CAMPEÕES FASE DE GRUPOS - RODADA 3											
MANCHESTER CITY			2 x 1		NAPOLI						
Sterling, Gabriel Jesus					Diawara						

### Jesus deixa o seu, Ederson pega pênalti, e City bate o Napoli para seguir 100% na Champions

Com participação decisiva de brasileiros, time de Guardiola passa por teste de fogo para seguir isolado na liderança do Grupo F. Italianos mostram poder de reação, mas ficam pressionados com triunfo do Shakhtar

Links para outros documentos  
hipertexto

# Representação de Texto

10

## □ Conteúdo Léxico

- ▣ Caracteres que constituem as palavras e outras unidades de pontuação
- ▣ Transmite o seu significado: semântica

## □ Aparência

- ▣ Conjunto de atributos visuais
  - Forma dos caracteres e as suas dimensões
  - Forma como o conteúdo se dispõe no tela
- ▣ Pode afetar a facilidade com que o texto é lido

# Representação de Texto

11

- Processamento de Texto
  - ▣ Operações em caracteres
  - ▣ Operações em strings
  - ▣ Edição de texto
  - ▣ Formatação de texto
  - ▣ Compressão de texto
  - ▣ Cifragem de texto
  - ▣ Verificação ortográfica e gramatical de texto

# Processamento de Texto

12

- Operações em caracteres
  - ▣ Operações mais simples, em caracteres individuais
  - ▣ Comparações na ordem dos caracteres
    - Ex.: ordenação alfabética
  
- Operações em strings
  - ▣ Operações em sequência de caracteres
  - ▣ Ex.: comparação, adição e extração de substrings
    - Prefixos, sufixos, radical, palavras compostas,..

# Processamento de Texto

13

## □ Formatação de Texto

- ▣ Permite alterar o aspecto visual dos caracteres
- ▣ Permite alterar a disposição espacial do texto (*layout*)
- ▣ Formata a apresentação do texto (renderização)

## □ Cifragem de texto

- ▣ Aplica métodos de criptografia para codificar o texto
  - Somente pode ler quem tem autorização/chave
- ▣ Privacidade dos dados

# Processamento de Texto

14

## □ Compressão de Texto

- ▣ Remove a redundância existente nas representações mais básicas
- ▣ Reduz o espaço de armazenamento de um dado documento de texto
  - Ex.: Huffman, LZW

## □ Verificação Ortográfica e Gramatical de Texto

- ▣ Verifica a ortografia, analisando gramaticalmente e/ou estatisticamente
- ▣ Pode fornecer/sugerir correções ou sinônimos
- ▣ Depende do idioma
- ▣ Pode ser automática ou não

# Codificação de Texto

15

- Usado para representar computacionalmente um conjunto de caracteres
- Necessário uma normalização/padronização para representar texto em dispositivos computacionais distintos
  - ▣ Independente de tecnologia e/ou fabricante
- Exemplos de padrões:
  - ▣ ASCII (1963), Unicode (1990s), Morse (1840s), Braille (1830s), Baudot (1870s), ISO-8859-1 (1987), etc.

# Compressão de Texto

16

- De modo geral, há dois métodos de compressão estatística para texto
  - ▣ 1. Utiliza caracteres únicos como base para derivar um conjunto de palavras-código
  - ▣ Quais caracteres e suas respectivas frequências de ocorrência
  - ▣ Codificação estática
    - Ex.: Huffman
  - ▣ 2. Baseia-se no tamanho variável de strings
  - ▣ Tipo de texto pode variar
  - ▣ Aplicações mais gerais
  - ▣ Codificação dinâmica
    - Ex.: LempelZiv (LZ)



# Codificação por Entropia

17

- Em 1948 o pesquisador Claude E. Shannon publicou um artigo intitulado : *A Mathematical Theory of Communication*
  - ▣ Formulava a teoria da compressão de dados
  - ▣ Indicava que existe um limite fundamental para as compressões **sem perdas**
    - denotada por  $H$  e conhecida como **taxa de entropia**

# Codificação por Entropia

18

- ❑ O valor exato de  $H$  depende da informação – mais especificamente, da natureza estatística da mesma
- ❑ É possível comprimir uma informação (sem perdas) com taxa de compressão próxima a  $H$
- ❑ Entretanto, é impossível comprimir com taxas melhores (menores) que  $H$

# Codificação por Entropia

19

- É independente da informação sendo comprimida
  - ▣ Como consequência..
- Utilizada na codificação de várias mídias
  - ▣ Baseia-se apenas na representação da informação

# Codificação por Entropia

20

## □ Entropia

- ▣ Quantidade de **informação** que um **dado** carrega
- ▣ Quantidade de informação:
  - Menor número de bits (unidade de informação) necessários para conter todos os valores ou significados de uma mensagem
  - Exemplo: quantos bits são necessários para transmitir ou armazenar os números dos meses do ano?

# Codificação por Entropia

21

## □ Entropia da fonte:

- ▣ número de bits mínimo que são **teoricamente** necessários para “transmitir” um código da fonte de informações (entropia da fonte)

## ▣ Taxa de Entropia:

- Fórmula de Shannon

$$H = -\sum_{i=1}^n P_i \log_2 P_i$$

$n$  = número de diferentes símbolos

$P_i$  = probabilidade de ocorrência do símbolo  $i$

# Codificação por Entropia

22

- Eficiência de um **esquema** de compressão:
  - Taxa da entropia da fonte comparado ao número médio de bits por código do esquema
    - Quanto mais próximo o segundo estiver do primeiro, melhor o esquema
  - Número médio de bits por código:

$$\sum_{i=1}^n N_i P_i$$

$n$  = no. símbolos

$N_i$  = no. caracteres/código

$P_i$  = probabilidade

# Codificação por Entropia

23

## Exemplo:

Suponha um novo método de compressão

Alfabeto	M	F	Y	N	O	1
Frequência	0.25	0.25	0.125	0.125	0.125	0.125
Códigos	10	11	010	011	000	001

Qual a entropia da fonte?

Qual o número médio de bits por código?

# Codificação por Entropia

24

## Exemplo:

Suponha um novo método de compressão

Alfabeto	M	F	Y	N	O	1
Frequência	0.25	0.25	0.125	0.125	0.125	0.125
Códigos	10	11	010	011	000	001

Qual a entropia da fonte?  $H = -\sum_{i=1}^n P_i \log_2 P_i$

Qual o número médio de bits por código?  $\sum_{i=1}^n N_i P_i$



# Codificação por Entropia

25

## Exemplo:

Suponha um novo método de compressão

Alfabeto	M	F	Y	N	O	1
Frequência	0.25	0.25	0.125	0.125	0.125	0.125
Códigos	10	11	010	011	000	001

Qual a entropia da fonte?  $H = -\sum_{i=1}^n P_i \log_2 P_i$

■  $H = -((0.25 * \log_2 0.25) + (0.25 * \log_2 0.25) + (0.125 * \log_2 0.125) + (0.125 * \log_2 0.125) + (0.125 * \log_2 0.125) + (0.125 * \log_2 0.125)) = -(-1 - 1.5) = 2.5$

Qual o número médio de bits por código?  $\sum_{i=1}^n N_i P_i$

■  $M = ((2 * 0.25) + (2 * 0.25) + (3 * 0.125) + (3 * 0.125) + (3 * 0.125) + (3 * 0.125)) = 1 + 1.5 = 2.5$

# Codificação por Entropia

26

- Se todas as probabilidades forem iguais

$$H = \log_2(n)$$

- Exemplo: dado
- 6 estados (faces)
  - ▣  $H = \log_2(6) = 2,58$  bits
  - ▣ .58 bits?
- 2 bits não são suficientes
- 3 bits: “sobram” 2 estados
  - ▣ Como?



# Codificação por Entropia

27

□  $H = \log_2(6) = \mathbf{2,58 \text{ bits}}$



□ 3 bits: sobram 2 estados

Código	Estado (dado)
000	1
001	2
010	3
011	4
100	5
101	6
110	
111	

# Codificação por Entropia

28

□  $H = \log_2(6) = 2,58 \text{ bits}$



- 3 bits: sobram 2 estados
- Opção: código redundante
- Podemos ser mais eficientes?

Código	Estado (dado)
000	1
001	2
010	3
011	4
100	5
101	5
110	6
111	6

# Codificação por Entropia

29

□  $H = \log_2(6) = 2,58 \text{ bits}$



□ Para estados 5 e 6, não transmitimos o último bit

□ Média de bits transmitidos

□  $(4*3+2*2)/6=16/6=2,67$

Código	Estado (dado)
000	1
001	2
010	3
011	4
10	5
10	5
11	6
11	6

# Codificação por Entropia

30

□  $H = \log_2(6) = \mathbf{2,58 \text{ bits}}$



Próximo da entropia

- Para estados 5 e 6, não transmitimos o último bit
- Média de bits transmitidos
- $(4*3+2*2)/6=16/6=2,67$

Código	Estado (dado)
000	1
001	2
010	3
011	4
10	5
10	5
11	6
11	6

# Codificação por Entropia

31

□  $H = \log_2(6) = \mathbf{2,58 \text{ bits}}$



Próximo da entropia

- Para estados 5 e 6, não transmitimos o último bit
- Média de bits transmitidos
- $(4*3+2*2)/6=16/6=2,67$
- Anterior:  $(6*3)/6=18/6=3$

Código	Estado (dado)
000	1
001	2
010	3
011	4
10	5
10	5
11	6
11	6

# Codificação por Entropia

32

- As técnicas de codificação de entropia podem ainda ser divididas em quatro tipos principais:
  - ▣ Técnicas de supressão de sequências repetitivas
  - ▣ Técnicas de codificação estatística
  - ▣ Técnicas baseadas em dicionários
  - ▣ Codificação por diferença



# Codificação por Entropia

33

- Técnicas de supressão de sequências repetitivas
  - Baseia-se na produção de códigos de comprimento fixo e funciona em dois passos:
    - 1) Detecção de sequência repetitivas de bits ou bytes
    - 2) Consequente substituição destas sequências pelo seu número de ocorrências
- Ex.: Técnicas de supressão de zeros ou espaços e codificação de comprimento de carreira (RLE – *Run Length Encoding*)

# Codificação por Entropia

34

- Técnicas de codificação estatística
  - ▣ Baseada na frequência dos símbolos
  - ▣ Símbolos com maior frequência = menor código
  
- ▣ Propriedade do prefixo
  - Um código não pode ser prefixo de um código mais longo
- ▣ Ex.:  $a=01$ ,  $b=10$ ,  $c=110$ ,  $d=001$
- ▣  $011010110001 \Rightarrow abbc d$
  
- ▣ Ex.: Códigos de Huffman, Shannon-Fano e a codificação aritmética

# Codificação por Entropia

35

## □ Técnicas de codificação estatística

### ▣ Contra-exemplo :

■  $a=01, b=10, c=110, d=101$

■ 01|101|01|10?    adab

ou

■ 01|10|10|110?    abbc

# Codificação por Entropia

36

- Técnicas baseadas em dicionários
  - Ocorre quando, toda vez que uma frase é **repetida**, **ela é substituída por uma referência** à ocorrência original da frase
  - A compactação resultante pode ser significativa dependendo da **redundância** de informações
  - Esse tipo de compressão é feita pelos códigos de Lempel, Ziv e Welch (LZW)

# Codificação por Entropia

37

- Codificação por diferença
  - ▣ Usado quando a amplitude de um sinal pode assumir valores em uma **faixa larga**, mas a diferença entre valores consecutivos é **pequena**
  - ▣ Codifica a diferença
    - Primeiro valor
  - ▣ Pode ocasionar ou não perdas
    - Depende do número de bits usados para armazenar a maior diferença entre amplitudes sucessivas
  - ▣ Exemplo:
    - 33200, 33100, 33050, 33152, ...
    - 33200, -100, -50, 102, ...

# Compressão em Texto

38

Método comuns de compressão de texto

- Keyword Encoding
- Técnicas de supressão de sequência repetitivas ou codificação de comprimento de carreira
  - ▣ Técnicas de supressão de zeros ou espaços
  - ▣ Técnica *Run-Length Encoding* (RLE)
- Técnicas Estatísticas
  - ▣ Codificação de Huffman
  - ▣ Codificação Aritmética
- Técnica Baseada em dicionário
  - ▣ Lempel-Ziv e Lempel-Ziv-Welsh

# Compressão em Texto

39

## Codificação com palavras-chave (*Keyword Encoding*)

- Substituir palavras muito comuns por caracteres especiais ou sequências de caracteres
- As palavras são substituídas de acordo com uma tabela de símbolos

Chave	Significado
%	Carro
\$	Acidente
&	Senhor
#	Do

# Compressão em Texto

40

## **Keyword Encoding:** Exemplo

- “No acidente estiveram envolvidos três carros. O carro do senhor Antonio ficou destruído. O carro do senhor José não sofreu grandes danos no acidente. O carro do senhor Carlos... bom, depois do acidente, nem se pode chamar aquilo um carro...” → 241bytes
- “No \$ estiveram envolvidos três carros. O % # & Antonio ficou destruído. O % # & José não sofreu grandes danos no \$. O % # & Carlos... bom, depois # \$, nem se pode chamar aquilo um %...” → 185bytes
- Taxa de compressão =  $185 / 241 = \sim 0.77$  (1-0.77) = 0.23



# Referências

41

- ❑ Halsall, F. Multimedia Communications: Applications, Networks, Protocols, and Standards, Addison-Wesley Publishing, 2001. ISBN: 0201398184. Capítulos 2 e 4.
- ❑ Mandal, M. K. Multimedia Signals and Systems. Kluwer Academic Publishers, 2002. ISBN: 1402072708. Capítulo 6
- ❑ Ribeiro, Nuno, and José Torres. "Tecnologias de Compressão Multimédia." (2009).
- ❑ Notas de aula da equipe de professores de natureza da informação. Universidade Federal do ABC.