

Práticas de Desenvolvimento de Software

#

Aula 06

Codificação de caracteres, expressões regulares e threads

Preenchendo sua caixa de ferramentas

Sistemas de codificação de
caracteres

Expressões regulares

Threads

Preenchendo sua caixa de ferramentas

Sistemas de codificação de
caracteres

Expressões regulares

Threads

Codificação de caracteres

“Um sistema de codificação de caracteres consiste em um código que mapeia cada caractere de um repertório em outros elementos, tais como padrões de bits, sequências de números, pulsos elétricos, etc.”

Codificação de caracteres

“Hello, world!”



?

Codificação de caracteres

“Hello, world!”



...11001010101011001101111000101011...

Codificação de caracteres

Caracter		Código Morse	ASCII Bin (Dec)	Unicode Hex (Dec)
A	→	● ■■	1000001 (65)	U+0041 (65)
F	→	● ● ■■ ●	1000110 (70)	U+0046 (70)
5	→	● ● ● ● ●	0110101 (53)	U+0035 (53)
6	→	■■ ● ● ● ● ●	0110110 (54)	U+0036 (54)

ASCII

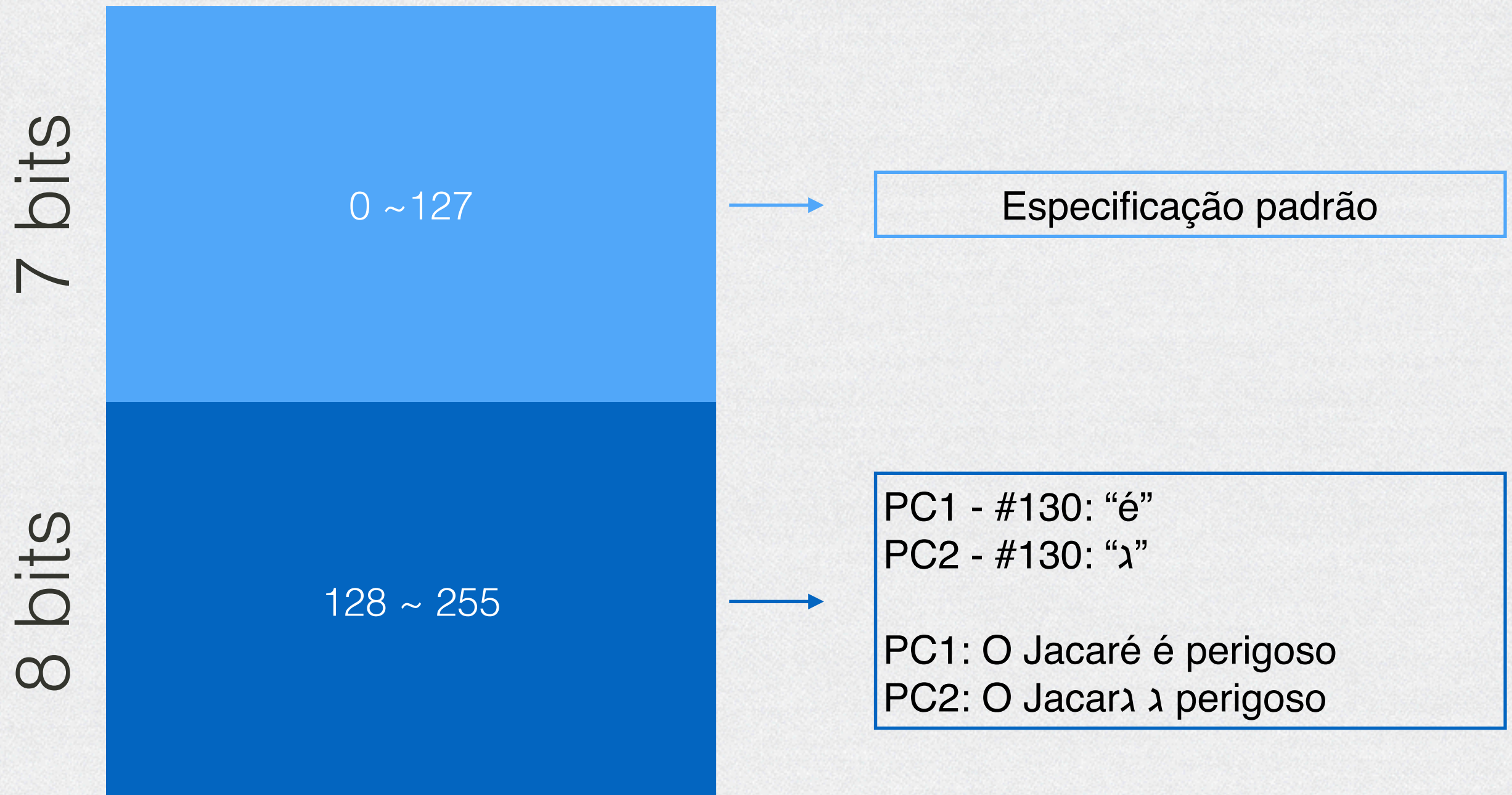
American Standard Code for Information Interchange

- Representação de texto em computadores
- Baseado no alfabeto americano
- Codifica **128 caracteres** em inteiros de **7 bits**
 - 33 não imprimíveis (caracteres de controle)
 - 95 imprimíveis

Sistemas de codificação de caracteres

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		Del

ASCII



Unicode

- **Conjunto de caracteres** que inclui todos os sistemas de escrita do planeta
- Repertório com mais de 110.000 caracteres
- Caracteres não são mapeados em sequências binárias, mas sim em “**code points**” (conceito teórico)

Caracatere	Code point (hex)
A	U+0041
£	U+0510
Œ	U+063D

Unicode

O sistema que codificar o Unicode idealmente deve resolver alguns problemas:

1. Manter compatibilidade com o ASCII
2. Não explodir o espaço de armazenamento necessário
3. Alguns sistemas interpretam sequências de 8 zeros como final de string

UTF-8

UCS Transformation Format — 8-bit
Universal Character Set

Range de code points		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
U+0000	U+007F	0xxxxxxx	-	-			
U+0080	U+07FF	110xxxxx	10xxxxxx	-			
U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx			
...
U+4000000	U+7FFFFFFF	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

Codificação de caracteres

Não há sentido em ter uma string sem saber em qual codificação ela está representada.

Preenchendo sua caixa de ferramentas

Sistemas de codificação de
caracteres

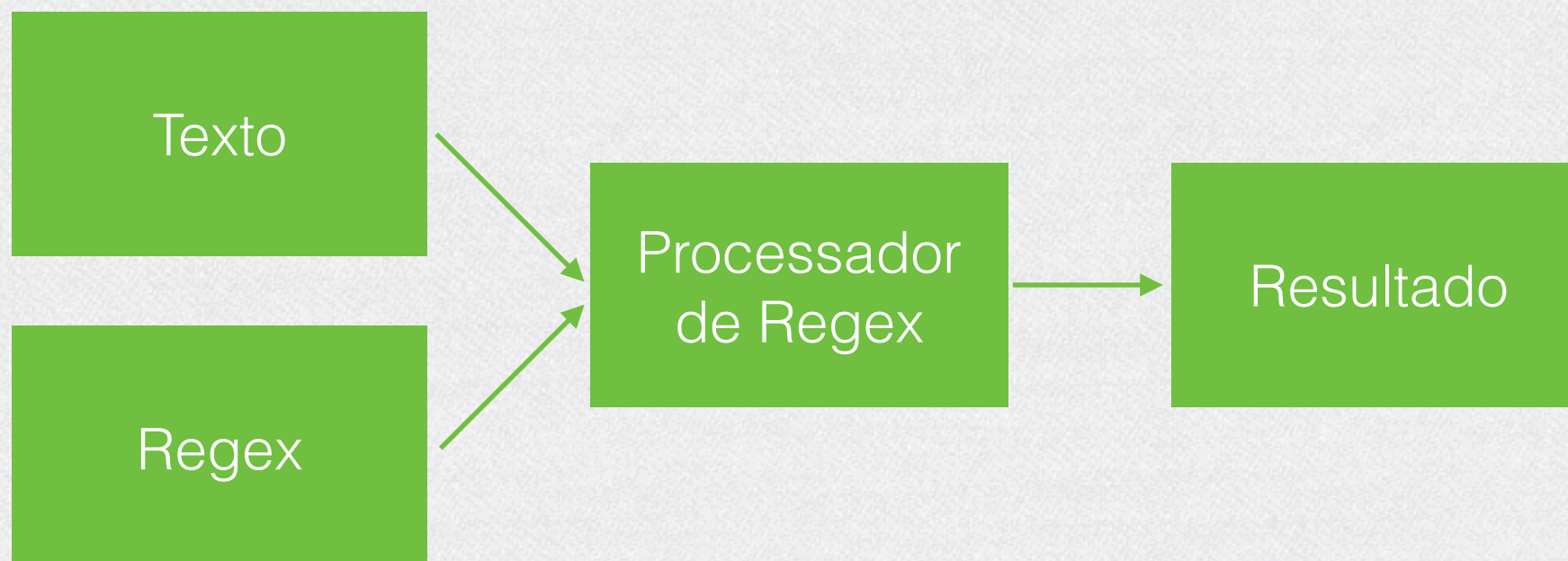
Expressões regulares

Threads

O que são expressões regulares?

A.K.A. Regular Expressions, **Regex**, Regexp

- Símbolos que representam um **padrão de texto**
- Úteis para **procurar, comparar e substituir** partes de um texto
- Uma regex “dá **match**” em um texto se ela o descreve corretamente



OK... E para que servem?

Exemplos de uso

- Procurar a palavra “Linux” em um documento
- Testar se um email está um formato válido
- Testar se um telefone tem o número correto de dígitos
- Substituir todas as ocorrências de “gato” ou “cachorro” por “rato” em um texto
- Contar o número de vezes que “aluno” é sucedido por “inteligente” em um texto

Metacaracteres

Met.	Descrição	Regex	Match	Não match
.	Match em qualquer caractere.	/a.c/		
[]	Match em caractere que está indicados dentro dos colchetes.	/[abc]/		
[^]	Match em um caractere que não está indicado dentro dos colchetes.	/[^abc]/		
^	Match no início da string. (Ou de uma linha do texto)	/^abc/		
\$	Match no final da string. (Ou de uma linha do texto)	/abc\$/		

Metacaracteres

Met.	Descrição	Regex	Match	Não match
.	Match em qualquer caractere.	/a.c/	"abc", "a2c"	"ac", "abbc"
[]	Match em caractere que está indicados dentro dos colchetes.	/[abc]/		
[^]	Match em um caractere que não está indicado dentro dos colchetes.	/[^abc]/		
^	Match no início da string. (Ou de uma linha do texto)	/^abc/		
\$	Match no final da string. (Ou de uma linha do texto)	/abc\$/		

Metacaracteres

Met.	Descrição	Regex	Match	Não match
.	Match em qualquer caractere.	/a.c/	"abc", "a2c"	"ac", "abbc"
[]	Match em caractere que está indicados dentro dos colchetes.	/[abc]/	"a", "b", "c"	"x", "2", "u"
[^]	Match em um caractere que não está indicado dentro dos colchetes.	/[^abc]/		
^	Match no início da string. (Ou de uma linha do texto)	/^abc/		
\$	Match no final da string. (Ou de uma linha do texto)	/abc\$/		

Metacaracteres

Met.	Descrição	Regex	Match	Não match
.	Match em qualquer caractere.	/a.c/	"abc", "a2c"	"ac", "abbc"
[]	Match em caractere que está indicados dentro dos colchetes.	/[abc]/	"a", "b", "c"	"x", "2", "u"
[^]	Match em um caractere que não está indicado dentro dos colchetes.	/[^abc]/	"x", "#", "9"	"a", "b", "c"
^	Match no início da string. (Ou de uma linha do texto)	/^abc/		
\$	Match no final da string. (Ou de uma linha do texto)	/abc\$/		

Metacaracteres

Met.	Descrição	Regex	Match	Não match
.	Match em qualquer caractere.	/a.c/	"abc", "a2c"	"ac", "abbc"
[]	Match em caractere que está indicados dentro dos colchetes.	/[abc]/	"a", "b", "c"	"x", "2", "u"
[^]	Match em um caractere que não está indicado dentro dos colchetes.	/[^abc]/	"x", "#", "9"	"a", "b", "c"
^	Match no início da string. (Ou de uma linha do texto)	/^abc/	"abcdef"	"2abcdef"
\$	Match no final da string. (Ou de uma linha do texto)	/abc\$/		

Metacaracteres

Met.	Descrição	Regex	Match	Não match
.	Match em qualquer caractere.	/a.c/	"abc", "a2c"	"ac", "abbc"
[]	Match em caractere que está indicados dentro dos colchetes.	/[abc]/	"a", "b", "c"	"x", "2", "u"
[^]	Match em um caractere que não está indicado dentro dos colchetes.	/[^abc]/	"x", "#", "9"	"a", "b", "c"
^	Match no início da string. (Ou de uma linha do texto)	/^abc/	"abcdef"	"2abcdef"
\$	Match no final da string. (Ou de uma linha do texto)	/abc\$/	"2abc"	"2abcdef"

Metacaracteres

Met.	Descrição	Match	Match	Não match
*	Match do elemento precedente zero ou mais vezes.	<code>/ab*c/</code>		
?	Match do elemento precedente zero ou uma vez.	<code>/ab?c/</code>		
{m,n}	Match do elemento precedente entre m e n vezes (inclusive).	<code>/a{1,2}/</code>		
\d	Match em caracteres numéricos (0-9).	<code>/\d{3}/</code>		
()	Captura o trecho de texto entre os parênteses.	—	—	—

Metacaracteres

Met.	Descrição	Match	Match	Não match
*	Match do elemento precedente zero ou mais vezes.	<code>/ab*c/</code>	<code>"abc"</code> , <code>"abbbc"</code>	<code>"abbbbdcc"</code>
?	Match do elemento precedente zero ou uma vez.	<code>/ab?c/</code>		
{m,n}	Match do elemento precedente entre m e n vezes (inclusive).	<code>/a{1,2}/</code>		
<code>\d</code>	Match em caracteres numéricos (0-9).	<code>/\d{3}/</code>		
()	Captura o trecho de texto entre os parênteses.	–	–	–

Metacaracteres

Met.	Descrição	Match	Match	Não match
*	Match do elemento precedente zero ou mais vezes.	<code>/ab*c/</code>	<code>"abc", "abbbc"</code>	<code>"abbbbdc"</code>
?	Match do elemento precedente zero ou uma vez.	<code>/ab?c/</code>	<code>"abc", "acd"</code>	<code>"abbc", "bc"</code>
<code>{m,n}</code>	Match do elemento precedente entre m e n vezes (inclusive).	<code>/a{1,2}/</code>		
<code>\d</code>	Match em caracteres numéricos (0-9).	<code>/\d{3}/</code>		
<code>()</code>	Captura o trecho de texto entre os parênteses.	—	—	—

Metacaracteres

Met.	Descrição	Match	Match	Não match
*	Match do elemento precedente zero ou mais vezes.	<code>/ab*c/</code>	<code>"abc", "abbbc"</code>	<code>"abbbbbc"</code>
?	Match do elemento precedente zero ou uma vez.	<code>/ab?c/</code>	<code>"abc", "acd"</code>	<code>"abbc", "bc"</code>
{m,n}	Match do elemento precedente entre m e n vezes (inclusive).	<code>/a{1,2}/</code>	<code>"a", "aa"</code>	<code>"aaa", "b"</code>
<code>\d</code>	Match em caracteres numéricos (0-9).	<code>/\d{3}/</code>		
()	Captura o trecho de texto entre os parênteses.	–	–	–

Metacaracteres

Met.	Descrição	Match	Match	Não match
*	Match do elemento precedente zero ou mais vezes.	/ab*c/	"abc", "abbbc"	"abbbbdc"
?	Match do elemento precedente zero ou uma vez.	/ab?c/	"abc", "acd"	"abbc", "bc"
{m,n}	Match do elemento precedente entre m e n vezes (inclusive).	/a{1,2}/	"a", "aa"	"aaa", "b"
\d	Match em caracteres numéricos (0-9).	/\d{3}/	"123y", "228"	"7b22", "65"
()	Captura o trecho de texto entre os parênteses.	-	-	-

É preciso praticar :)

rubular.com

Exemplos práticos

Notícias

- 09:55 - [política] Prefeito da Krakosia é eleito o melhor para economia
- 10:14 - [cotidiano] Cachorro salva dono de assalto
- 11:01 - [esporte] Seleção local de golfe se classifica para mundial
- 11:36 - [esporte] João da Silva Sauro é cortado da seleção de golfe
- 13:19 - [economia] Desemprego sobe para 12% no último mês
- 14:55 - [política] Lei do esporte é aprovada às 10:30
- 16:13 - [economia] Cotação do dólar sobe 5% em uma semana

Exemplos práticos

Notícias

- 09:55 - [política] Prefeito da Krakosia é eleito o melhor para economia
- 10:14 - [cotidiano] Cachorro salva dono de assalto
- 11:01 - [esporte] Seleção local de golfe se classifica para mundial
- 11:36 - [esporte] João da Silva Sauro é cortado da seleção de golfe
- 13:19 - [economia] Desemprego sobe para 12% no último mês
- 14:55 - [política] Lei do esporte é aprovada às 10:30
- 16:13 - [economia] Cotação do dólar sobe 5% em uma semana

Quantas notícias de **esporte** existem?

Exemplos práticos

Notícias

- 09:55 - [política] Prefeito da Krakosia é eleito o melhor para economia
- 10:14 - [cotidiano] Cachorro salva dono de assalto
- 11:01 - [esporte] Seleção local de golfe se classifica para mundial
- 11:36 - [esporte] João da Silva Sauro é cortado da seleção de golfe
- 13:19 - [economia] Desemprego sobe para 12% no último mês
- 14:55 - [política] Lei do esporte é aprovada às 10:30
- 16:13 - [economia] Cotação do dólar sobe 5% em uma semana

Quais são as **manchetes** das notícias de **economia**?

Exemplos práticos

Notícias

- 09:55 - [política] Prefeito da Krakosia é eleito o melhor para economia
- 10:14 - [cotidiano] Cachorro salva dono de assalto
- 11:01 - [esporte] Seleção local de golfe se classifica para mundial
- 11:36 - [esporte] João da Silva Sauro é cortado da seleção de golfe
- 13:19 - [economia] Desemprego sobe para 12% no último mês
- 14:55 - [política] Lei do esporte é aprovada às 10:30
- 16:13 - [economia] Cotação do dólar sobe 5% em uma semana

Quais são os **horários de publicação** das notícias?

Preenchendo sua caixa de ferramentas

Sistemas de codificação de
caracteres

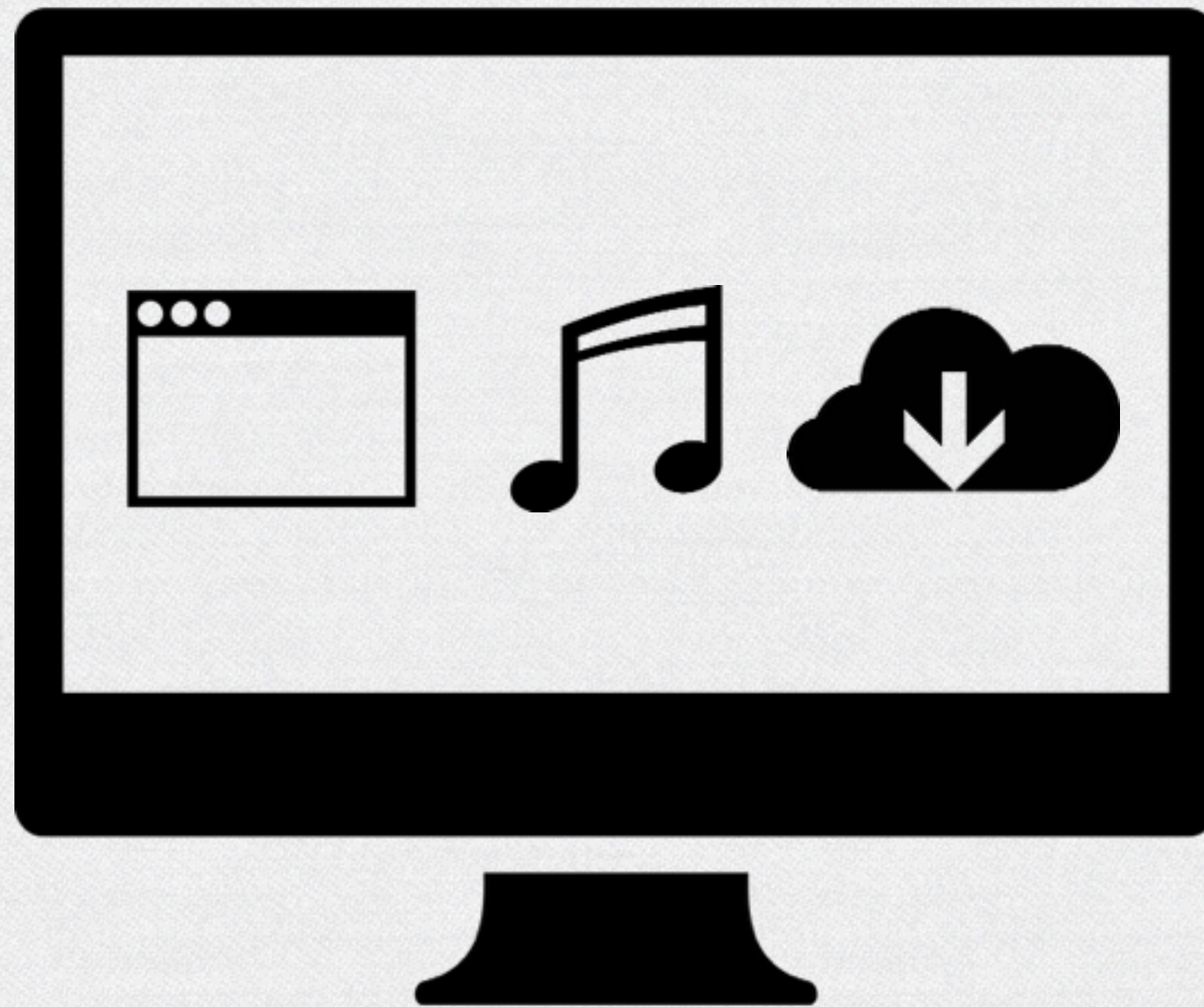
Expressões regulares

Threads

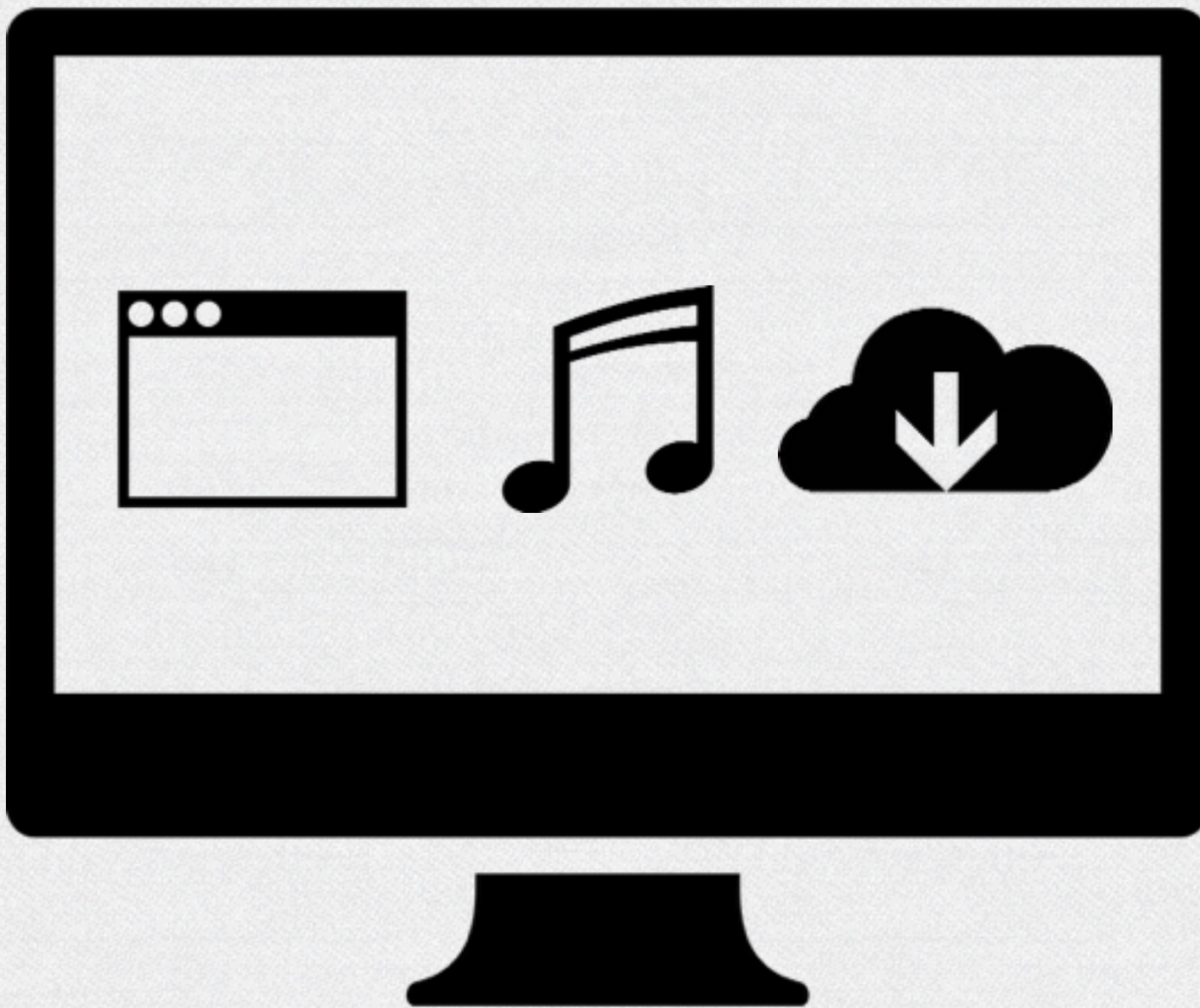
Tarefas concorrentes no computador



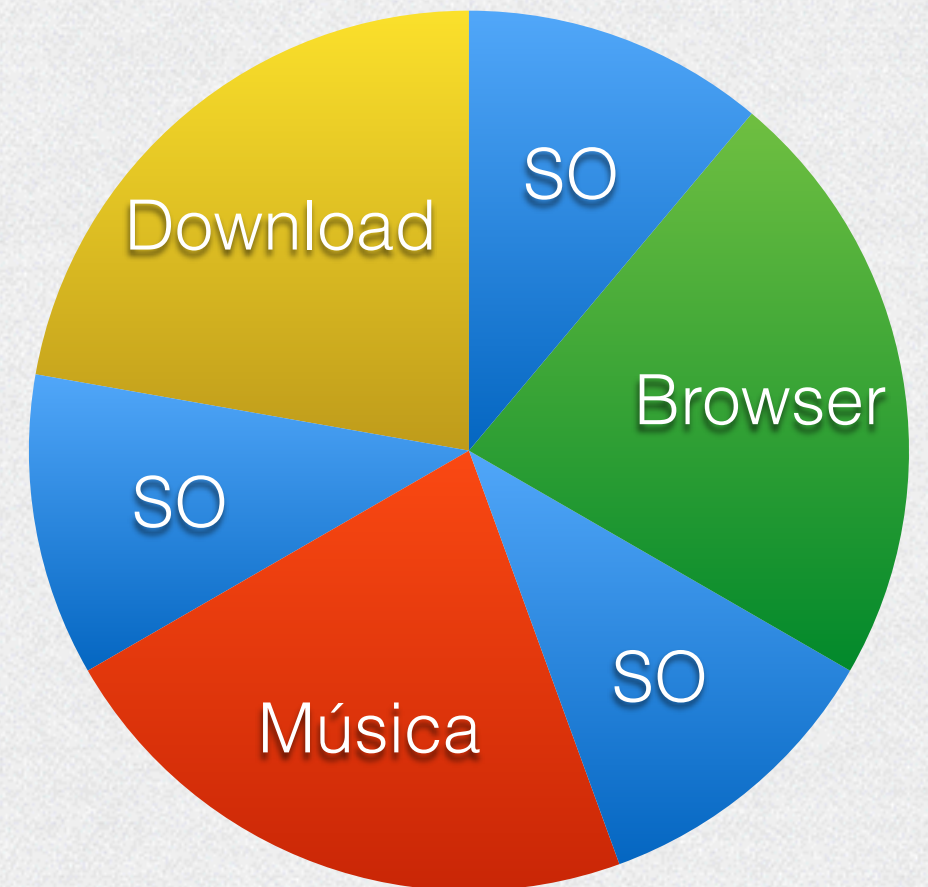
Tarefas concorrentes no computador



Tarefas concorrentes no computador

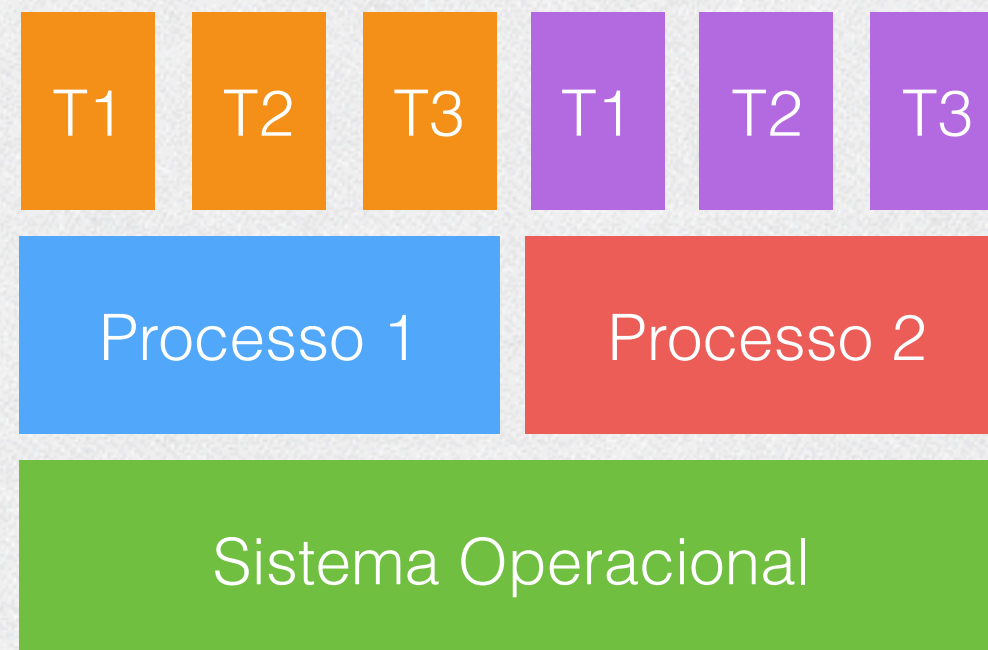


Tarefas concorrentes no computador



Processos e Threads

- O sistema operacional gerencia os processos e gasta tempo para isso
- Os processos aparentam ser executados em paralelo
- As threads compartilham recursos do processo



Threads in Ruby

```
Thread.new {  
  #  
  # What should this  
  # thread do?  
  #  
}
```


Threads em Ruby

```
threads = []

4.times do |i|

  t = Thread.new {
    puts "#{i}\n"
    puts "#{i}.#{i}\n"
    puts "#{i}.#{i}.#{i}\n"
  }

  threads << t

end

threads.each { |t| t.join }
```

Execução 1

```
0
0.0
0.0.0
1
1.1
1.1.1
2
2.2
2.2.2
3
3.3
3.3.3
4
4.4
4.4.4
```

Execução 2

```
2
1
0
0.0
0.0.0
4
1.1
3
1.1.1
2.2
4.4
3.3
3.3.3
4.4.4
2.2.2
```

Execução 3

```
0
1
2
0.0
1.1
3
0.0.0
4
1.1.1
4.4
2.2
3.3
3.3.3
2.2.2
4.4.4
```


Threads em Ruby

Execução 1

```
0
0.0
0.0.0
1
1.1
1.1.1
2
2.2
2.2.2
3
3.3
3.3.3
4
4.4
4.4.4
```

Execução 2

```
2
1
0
0.0
0.0.0
4
1.1
3
1.1.1
2.2
4.4
3.3
3.3.3
4.4.4
2.2.2
```

Execução 3

```
0
1
2
0.0
1.1
3
0.0.0
4
1.1.1
4.4
2.2
3.3
3.3.3
2.2.2
4.4.4
```

- Não é possível prever a ordem de execução das threads
- O ambiente e o SO escalonam a execução
- Resultados diferentes entre execuções

Threads

Exemplo: downloads de múltiplos arquivos

Threads

Cuidado: aumenta a complexidade do programa, dificuldade depuração de erros e manutenção do código.

Exercícios

Os exercícios devem ser entregues pelo GitHub seguindo o tutorial da aula 1.

Prazo: 22/04/2014 até às 18:30.

Exercícios

Sistemas de codificação de caracteres

6.1. O arquivo `ex6.1.html` do repositório não está sendo exibido corretamente quando aberto no Google Chrome. Sabe-se que o arquivo está codificado em UTF-8. Altere o arquivo HTML de modo que o browser consiga apresentá-lo corretamente.

6.2. Abra o arquivo `ex6.2.rb` e complete o exercício.

Exercícios

Threads

6.3. Abra o arquivo ex6.3.rb e complete o exercício.

Exercícios

Expressões regulares

6.4. Abra o arquivo ex6.4.rb e complete o exercício.