Instituto federal do Piauí
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas
Disciplina: Introdução à Computação
Professor: Ricardo Martins Ramos

Vinícius Gomes Araújo Costa
Matheus Levi da Silva Barbosa
Weslley David Pereira Marques

TIPOS DE DADOS

Teresina-PI
2021

Mindset do código binário

Nesta disciplina, o **armazenamento de dados** é feito, inicialmente, por **Sistemas de Numeração** que envolvem a comunicação do **homem** com a **máquina**. A linguagem do mundo digital/computacional é frequente para recorrer-se a diferentes **Sistemas de Numeração** como representação da **informação digital**.

Eles são importantes pois **o computador** se comunica de maneira diferente do uso convencional, que nós usamos no dia a dia. Isto é: **os dígitos binários**, popularmente chamados de **0 e 1**. São eles que irão caracterizar **quatro** (4) diferentes tipos de sistemasem **Linguagens de Máquina**: **Binário** (base 2), **Decimal** (base 10), **Octal** (base 08) e **Hexadecimal** (base 16).

Deste modo, **o computador** utiliza, por **convenção**, **valores codificados** em níveis de voltagem ou pulsos elétricos. O famoso **"bit de sinal**". O zero significaria desligado e o um ligado. Quando os primeiros computadores foram criados, enormes válvulas de circulação de ar interligavam**os comandos principais**, **registradores** e **os sistemas de memória**. Eram denominados"**mainframs**", usados para pesquisa militar e acadêmica das universidades no mundo todo. Porém, ainda eram lentos em velocidade de processamento, visualizador de imagense disco rígido (com pouca capacidade de espaço).

Uma observação e curiosidade importante a respeito da **Linguagem de Máquina** ou é que, atualmente**, os microcomputadores e eletrônicos** atuais trabalham com tensão elétrica, ou seja, identificam se está ligada(1) ou desligada(0). O **HD**, que armazena a **memória**, é responsável por enviar os códigos para **a placa mãe**, que por sua vez, aciona **o processador** do computador. O processador devolve a execução traduzida através das diversas saídas disponíveis que estão ligadas a placa mãe. O **Monitor**, por exemplo, é um tipo de saída. É por isso que conseguimos entender o código que está no monitor do computador.

Com isso, para aprimorar essa interação homem máquina, foram criados mais de **dez** (10)**tipos de representação** para armazenamento de dados. Será exposto mais adiante a pesquisa sobre esse assunto. O método é simples: o **programador** digita um algarismo (bit) no periférico de entrada, manda a resposta para a ALU (central), o computador calcula automaticamente os dados pela Tabela ASCII e decodifica a mensagem; exibindo-a na tela.

Binários e Encode

Apesar de proporcionarem grandes revoluções, a comunicação entre homem e máquina segue conceitos bastante primitivos. O **código binário**, também conhecido como linguagem de máquina, trata-se de um sistema de numeração composto por dois algarismos, 0 e 1. Claro que 0 e 1 ainda são conceitos sem significado para a máquina, isto é, para que realmente possa haver uma comunicação, é necessário **atribuir valores de presença ou ausência no circuito desses aparelhos**. Sendo assim, convencionou-se a atribuição do estado 0 para ausência de energia e 1 para a presença da mesma.

Tendo-se esclarecido os moldes da comunicação que rege a interação humano-computador, é de grande valia a compreensão do termo ‘encoding’ para a área da computação. Descreve-se encoding como o mecanismo que define como representamos diversos símbolos e letras de diferentes alfabetos de maneira binária. Indo além da mera representação de texto, também é possível a partir de tais princípios o armazenamento de imagens, vídeos e áudios.

Textos

Para **o computador**, um **texto**, seja ele uma **palavra** ou um **capítulo inteiro**, é denominado como string. O tratamento de strings se dá pela subdivisão de suas partes, sendo a menor parte de uma string os seus caracteres, estes por sua vez são convertidos em Bytes para que possam ser armazenados na memória.

Como dito anteriormente, representar letras (e outros caracteres, como sinais de pontuação, espaços e símbolos) como números é chamado de 'encoding' e requer o uso de algo chamado 'conjunto de caracteres', que na verdade nada mais é do que uma tabela de consulta interna. Um dos conjuntos de caracteres mais antigos é chamado **ASCII (American Standard Code for Information Interchange)**, que fornece uma pesquisa para 128 caracteresdiferentes, incluindo as letras AZ em maiúsculas e minúsculas.

Partindo a um exemplo prático, no armazenamento de uma letra 'H' qualquer, a mesma é codificada por meio do **conjunto de caracteres ASCII** como o inteiro 72, que em binário se torna **01001000**, e que pode ser armazenado como um único byte pelo computador. Quando um usuário deseja recuperar aquele byte da memória, desde que o programa que o recupera saiba usar o conjunto de caracteres ASCII, o byte 01001000 será traduzido para que apareça na tela como 'H'. Sem o conjunto de caracteres, no entanto,o computador apenas exibiria o número 72, que é parte da razão pela qual os conjuntos de caracteres são tão importantes. Lembrando de que uma string também pode incluir caracteres numéricos, levando à noção confusa de que o número 49 é usado em ASCII para representar o caractere numérico '1'.

Imagens

Pixel é a menor unidade de uma imagem digital, a partir da combinação entre pixels um computador é capaz de representar os mais diversos objetos. Em nível de linguagem de máquina, **cada pixel é representado por um bit ou um conjunto de bits**. A lógica que determina a quantidade de bits que cada pixel precisa para ser representado depende do número de cores presente na imagem, por exemplo.

Representação de Dados

bit: Deriva do inglês ***binary digit*** *(dígito binário)*, é **a menor unidade de um dado**. Pode representar dois valores quaisquer e é representado pela letra minúscula b. A sua importância em **Introdução à Computação** e matérias afins seria como o observador geraria valores desconhecidos para uma variável. Valor falso, verdade, desligado e ligado, assim por diante.

Unidade	Símbolo	Valor Equivalente	Múltiplo
Bit	b*		
Byte	B*	8 bits	10^0
Kilobyte	KB	1024 B	10^3
Megabyte	MB	1024 KB	10^6
Gigabyte	GB	1024 MB	10^9
Terabyte	TB	1024 GB	10^{12}
Petabyte	PB	1024 TB	10^{15}
Exabyte	EB	1024 PB	10^{18}
Zettabyte	ZB	1024 EB	10^{21}
Yottabyte	YB	1024 ZB	10^{24}

Na imagem acima, têm-se a classificação de bits na computação e a ideia de **Banco de Dados na computação**.

Nibble:

- São 4 dígitos binário
- Pode representar até 16 valores;
- 1 nibble equivale a 1 dígito hexadecimal.

Byte:

- Representado pela letra B maiúscula;
- O menor item de dados que pode ser acessado;
- São 8 bits, logo permite 256 combinações ($2^8 = 256$)

-1 **Byte** = 1 caractere de 8 bits

-1 **kiloByte** (KB) = 1024 Bytes ou 2^3

-1 **Megabyte** (MB) = 1024 kiloBytes ou 2^6

-1 **GibaByte** (GB) = 1024 MegaBytes ou 2^9

-1 **TeraByte** (TB) = 1024 GigaBytes ou 2^{12}

Tabela ASCII

Assim como **o observador** precisou olhar uma **tabela** para estabelecer parâmetros de medida, esta tabela mostra **o espelho** do nosso **alfabeto** e traduz uma determinada quantidade de bits em **caracteres** do nosso alfabeto.

A **Tabela ASCII** (do inglês "*Standard Code of Informaton Interchange"*, significa **Código Padrão Americano para Intercâmbio de Informação**) é usada pela maioria da indústria de computadores para a troca de informações. Cada caractere é representado por **um código de 8 bits** que inclui os caracteres acentuados.

Representa **128 sinais**, **95 sinais gráficos** (letras do alfabeto latino, sinais de pontuação e sinais matemáticos) e **33 sinais de controle**, utilizando portanto apenas **7 bits** para representar todos os seus símbolos. A codificação ASCII é usada para representar **textos em computadores**, **equipamentos de comunicação**, entre outros dispositivos que trabalham com **texto**. Desenvolvida a partir de 1960, grande parte das codificações de caracteres modernas a herdaram como base.

Trata-se de uma **tabela-base de conversão** de **números/caracteres binários** em letras do alfabeto maiúsculas e minúsculas, armazenando a quantidade de bits por arquivos de demonstração.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Processamento de imagens

Em **jogos**, **imagens** são muito utilizadas em formato de **sprites**, **imagens de cenários** e na **composição da visualização de objetos**, sendo utilizadas na forma de **texturas** (mapeamento de texturas).

A partir de **dados vetoriais**, **as informações** descrevem **primitivas gráficas** para formar **desenhos**, ou seja, **pontos**, **curvas**, **linhas** ou **formas geométricas** quaisquer. Um programa que manipula este tipo de dado deve interpretar esta informação primitiva e transformá-la numa imagem.

Dados tipo **bitmap**:

- Dado gráfico é descrito como uma ***array de valores***, aonde cada valor representa uma **co**r;
- Chamamos cada elemento da imagem de **pixel**;
- O pixel é uma **estrutura de dados** que contém múltiplos bits para **representação de cores**;
- A quantidade de **bits** determina a quantidade de cores possível de se representar numa imagem.

E mais um conjunto de formação de códigos processuais que envolvem **Geometria Euclidiana**, **Programação Orientada a Objetos** e **Conjuntos Numéricos**.

Exemplos: ***raster scan-lines***, ***pixel-depth***, **gráficos de imagem**, entre outros.

Quando se navega na **Internet** e em **redes sociais**, nos deparamos com imagens de diversos formatos e que se apresentam os arquivos. Salvá-las em pastas fica a gosto dos usuários.

Vemos com facilidade o **JPG**, **GIF**, **PNG** (mais comuns atualmente) e **BMP**. Mas além desses, existe uma grande quantidade de formatos para diversos tipos de uso. Cada um possui uma especificação técnica diferente, pois são **compressão de pixels diferentes**.

- **JPEG:**

Joint Pictures Expert Group é o melhor formato para quem deseja enviar imagens por email. Surgiu em 1983 e acabou virando um dos padrões mais populares da Internet. Conhecido também como **JPG**.

- **GIF:**

Graphics Interchange Format é outro **formato** muito comum na **Internet**. É um arquivo leve e famoso pelas fotografias em movimento, **os gif's animados**. Só trabalha com 256 cores (8 bits), por isso não é muito comum em **fotografias**. Criado em 1987, o GIF foi projetado pela CompuServe nos primeiros dias de **vídeo dos computadores** de 8 bits, antes mesmo JPG, para visualização em velocidade de conexão para modem dial-up (discado).

- **PNG:**

Portable Network Graphics, ao contrário do GIF, o PNG suporta mais cores. É um concorrente do **GIF**. Surgiu em 1996 e possui características que tornaram o GIF tão bem aceito: **animação**, **fundo transparente** e **compressão sem perda de qualidade**, mesmo com salvamentos constantes do arquivo. Suporta milhões de cores, uma ótima opção para fotos. E **transparência por 24 imagens de bit RGB**.

- **Outros formatos:**

-TIFF, TIFF LZW, RAW, EXIF, PPM, PGM, PNM, SVG, CDR, WebP, Ilustrator,Photoshop entre outros recursos...

Processamento de vídeos

Criação do vídeo digital

Esses **dados brutos** consistem em vários sinais de bits (liga / desliga) -Para cada um desses pontos. Depende da **eletrônica** quantos e o que significa cada combinação possível, mas o princípio é sempre assim: Algum tipo de valor numérico para definir a cor e / ou quantidade de luz. Muitos usam uma forma do que é chamado de **HLV** (Hue, Luminescence, Value - efetivamente qual banda no espectro de cores, quão brilhante é e quão profunda é a cor), outros vão com a mistura de cores primárias (por exemplo, RGB, quanto vermelho, Verde e Azul estão incluídos), e vários outros tipos semelhantes podem ser usados - basicamente por que todos eles são chamados de "brutos".

Portanto, para armazenar tal valor - leia sobre **a numeração binária**. É assim que você transforma qualquer coisa em um lote de sinais liga / desliga. Digamos que optemos pelo formato HLV. Isso então é "digitalizado" pela divisão de todo o espectro para cada um desses valores em um número discreto, digamos que H = 0 significa vermelho puro, H = 127 significa verde puro, H = 255 significa roxo puro, etc. O mesmo vale para a luminosidade, 0= preto (sem luz), 255 significa branco absoluto. E novamente para o componente Valor / Saturação.Observe os números que usei? Esses são números "redondos" no **sistema de numeração binária** - o mesmo que 100 e 1000 são números redondos no sistema de numeração decimal. Basicamente, eles são potências da base do sistema ($100 = 10^2$, $1000 = 10^3$, etc.

A base do binário é 2 em vez de 10, portanto, $128 = 2^7$, $256 = 2^8$), mas eu estou procurando o mínimo e máximo possível em um conjunto de dígitos. Neste caso, 8 dígitos (bits) cada, como em 8 sinais consecutivos de ligar / desligar, me dá 256 números possíveis que variam de 0 a 255 (inclusive).

Então são 3 bytes (3 grupos de 8 bits em sequência) ou 24 bits para **cada ponto** na imagem (chamado de **pixel**). Dependendo de quantos pixels existem, isso é repetido o mesmo número de vezes. Portanto, uma foto HD 720p tem 1280x720 = 921600 pixels para uma única imagem. Então, tudo isso é repetido para cada quadro - ou seja, no nosso caso, 30 deles para cada segundo de vídeo.

FPS

O FPS (Frames per Second), ou Quadros por Segundo, é o que determina a quantidade de **quadros** que **o vídeo** terá a cada segundo. Por exemplo: se o vídeo tiver 30 FPS, isso significa que a cada segundo de vídeo serão mostradas 30 imagens (quadros) em sequência.

PixelSize × Width × Height × FramesPerSecond × 60SecondsPerMinute × 60MinutesPerHour
=3Bytes×1280×720×30fps×60spm×60mph
=3×921600×30×3600
=2764800×108000=298598400000Bytes
=291600000kB
(1 quilo Bytes = 1.024 Bytes, novamente 1024 = 2^{10})
=284765.625MB=278.091GB

Portanto, você precisa de 280 GB inteiros para armazenar cerca de uma hora de vídeo 720p 30fps.

Exibindo o vídeo

Todo o processo é basicamente revertido. Esses **dados** são enviados para alguns componentes eletrônicos que os convertem em combinações / intensidade elétrica para fazer um único pixel na tela parecer ter a mesma cor / intensidade / saturação de quando foi codificado.

Quanto espaço de armazenamento é isso?

Nenhum dos **AVI / MP4** / etc. Os arquivos que vi são enormes, mesmo aqueles contendo 2 horas de filme têm apenas um a 3 GB de tamanho. Isso porque há muita **repetição nos dados brutos**. Quaisquer dois quadros consecutivos podem ter grandes faixas de seus pixels com exatamente o mesmo valor. Assim, técnicas de compressão são usadas para reduzir o número de dados necessários para exibi-los. Ao compactar o mesmo pixel em vários quadros - é conhecido como compactação temporal (como ao longo do tempo). Outros meios são procurar pixels vizinhos para compactar da mesma maneira que os arquivos JPG / PNG / GIF reduzem a foto original. E outros testam para ver se os pixels (vizinhos e "futuro") estão "quase" o mesmo - quem vai notar a diferença entre um valor 176 e 175 em um único pixel? E então a maioria faz combinações disso.

Isso é o que você encontra dentro daqueles **AVI / MP4 / OGM / MKV** / etc. Eles têm alguma compressão que reduz bastante a quantidade de **espaço de armazenamento** necessária. Mesmo compressores muito pobres podem atingir uma redução de dez vezes no tamanho sem perder (muito) a fidelidade. Bons compressores tendem a obter uma redução de cerca de 100 vezes antes de você começar a notar artefatos de compressão, alguns até melhores do que isso.

Um vídeo é uma sequência de quadros. Cada quadro é uma coleção de pixels. Cada pixel é representado por uma sequência de números representando sua **cor**. A maior parte do vídeo é compactada. O arquivo compactado geralmente está em conformidade com uma sintaxe conhecida de um algoritmo de compactação de vídeo. Um reprodutor de vídeo de computador lê o arquivo e o decodifica para gerar quadros de vídeo, que são exibidos na tela.

Processamento de áudio

Em **computação**, **arquivo de som** (ou arquivo sonoro) é **um formato de arquivo** que permite **o armazenamento digital de áudio**. Em geral, esse arquivo armazena **intervalos regulares de amostras de som**, que representam a posição em que a **membrana da caixa de som** deve estar no momento da **gravação**.

Há três propriedades destes arquivos que determinam a qualidade do som armazenado e o seu tamanho. São eles: **a resolução**, ou seja***,* quantos bits** são usados para representar cada amostra, **a taxa de amostragem**, ou seja, **quantas amostras** são tomadas do som por segundo e por último, **o codec** que pode proporcionar formas mais ou menos eficientes para **armazenar estas informações**.

Observações:

- Placas sonoras armazenadas no HD transmitem frequências diferentes;
- Em contrapartida, o formato **MIDI** não segue esses princípios;
- Ele não armazena áudio propriamente dito, mas sim uma **sequência de notas musicais** que podem ser executadas por **sintetizadores**

Não se deve confundir **o codec** com o formato do arquivo;

- O **formato** especifica a disposição dos dados dentro do arquivo e o codec a forma como a informação sobre **o som** é tratada. Há **formatos de arquivo** que proporcionam a possibilidade de usar vários **codec's** para **codificar o som no arquivo**.

Formatos:

WAV (criado pela Microsoft), **AIFF** (criado pela Apple Inc.), **MP3**, **MP4**, **compressão Ogg-Vorbis**, **RA** (Real Player), **M4P** ...

- A **possibilidade de compressão** do formato **MP3** foi responsável por parte da popularização do mesmo, pois possibilitou o armazenamento de uma quantidade muito superior de **músicas** em um mesmo espaço de armazenamento.

Referências bibliográficas:

Acesso pelo link:https://www.portaleducacao.com.br/conteudo/artigos/educacao/compreendendo-o- codigo-binario/48352#
Acesso pelo link: https://www.techtudo.com.br/artigos/noticia/2012/07/entenda-os-formatos-dos-arquivos-de-imagem.html
Acesso pelo link: https://www.embarcados.com.br/tabela-ascii/
Acesso pelo link: https://pt.wikipedia.org/wiki/Arquivo_sonoro
Vídeo sobre O que é Frame, FPS, Fields e Distorção de tempo – Fundamentos da Edição EP05: https://www.youtube.com/watch?v=G0GnVxaXANM
Vídeo sobre Spriter Pro Animador 2D para Sprites de jogos: https://www.youtube.com/watch?v=WO9vUvuSXLs