

Representação Binária de Dados

O computador utiliza uma representação binária para fazer o armazenamento e manipulação dos dados. Dados neste caso podem ser programas armazenados, imagens, sons, textos, vídeos, dentre outros. Qualquer informação ou dado dentro de um computador é representado em números binários (zeros ou uns). Eles são a menor unidade de informação possível de ser representada digitalmente.

Para facilitar o processamento e gerenciamento dos dados, eles são agrupados em bytes (conjunto de 8 bits). Cada byte pode representar um intervalo de 0 a 255, ou seja, pode representar até 256 combinações diferentes de dados, pois $2^8=256$. Para melhor compreender a aritmética de números binários, basta fazer todas as combinações possíveis destes 8 bits:

00000000 = 0
00000001 = 1
00000010 = 2
00000011 = 3
00000100 = 4
....
11111111 = 255 = 128+64+32+16+8+4+2+1

Para compreender a conversão, veja o seguinte exemplo. O número 4 em binário vale

00000100 pois $4 = 0*2^7 + 0*2^6 + 0*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0$

Ou seja, $0 + 0 + 0 + 0 + 0 + 4 + 0 + 0 = 4$. Para mostrar que isso não tem mistério nenhum, podemos fazer uma analogia ao sistema decimal e mostrar que o número 2501 é formado pela seguinte expressão:

$$2*10^3 + 5*10^2 + 0*10^1 + 1*10^0 = 2000 + 500 + 0 + 1 = 2501.$$

Números **inteiros positivos** menores que 255 podem ser representados com um único byte, como mostrado anteriormente. Caso o número for maior, deve-se utilizar uma quantidade de bytes maior. Com 16 bits, pode-se representar até o número 65.535, com 24 até o número 16.777.216 e com 32 bits (4 bytes) até o número 4.294.967.295.

Por convenção, bytes são representados pela letra B e bits pela letra b. Essa terminologia é muito usada para descrever taxas de transferência de dados, como no caso de redes, modems e dispositivos de armazenamento. Como exemplo, um modem pode trafegar dados a uma taxa de 54kbps, ou seja, 54 kilo **bits** por segundo. Um HD pode ler dados a uma taxa de 10 MB/s, ou seja, 10 Mega **Bytes** por segundo. Muitas fontes de informação de baixa credibilidade (Internet, por exemplo) usam B e b como sinônimos. Deve-se também observar que em informática as convenções de grandezas não são iguais às conhecidas habitualmente, como mostrado na Tabela 1.

Tabela 1: Grandezas utilizadas em Informática

Símbolo	Tamanho	Comentários
Bit (b)	1	$2^0 = 1$. Menor unidade de informação: vale 0 ou 1.
Byte (B)	8 bits	$2^3 = 8$. Por convenção, e por ser potência de 2
Kilo (K)	1024 Bytes	$2^{10} = 1.024$
Mega (M)	1024 Kilo	$2^{20} = 1.048.576$
Giga (G)	1024 Mega	$2^{30} = 1.073.741.824$. Unidade dos HDs atuais
Tera (T)	1024 Giga	$2^{40} = 1.099.511.627.776$
Peta (P)	1024 Tera	$2^{50} = 1.125.899.906.842.624$
Exa (E)	1024 Peta	2^{60} . Talvez seja para os seus netos ☺
Zetta (Z)	1024 Exa	2^{70} . O que? Como ?
Yotta (Y)	1024 Zetta	2^{80} . Tu tá de brincadeira, né?

Representação de Caracteres

Os caracteres (texto) também são internamente representados por números de 8 bits (Tabela ASCII). Atualmente utiliza-se codificação UTF-8 (8-bit Unicode Transformation Format), codificação binária (Unicode) de comprimento variável que pode representar qualquer caractere universal padrão do Unicode, sendo também compatível com o ASCII.

```
<html class="client-nojs" lang="pt" dir="ltr">
<head>
<meta charset="UTF-8"/>
```

Neste caso, existe uma associação entre cada caractere com um número decimal, como mostrado na seguinte tabela. Nesta tabela, existem seqüências que representam os caracteres do alfabeto (verde e azul), e outra os dígitos numéricos (vermelho). Muitos caracteres desta tabela são considerados não imprimíveis. Como exemplo, pode-se citar os dígitos da primeira coluna. Para inserir em um arquivo texto os caracteres que não

estão presentes do teclado, deve-se manter pressionada a tecla <alt> e digitar o código decimal associado ao caractere em questão.

0=	25=↓	48=0	71=G	94=^	117=u	140=î	163=ú	186=
1=☺	26==→	49=1	72=H	95=_	118=v	141=ì	164=ñ	187=
2=☉	27=←	50=2	73=I	96=`	119=w	142=Ä	165=Ñ	188=
3=♥	28=└	51=3	74=J	97=a	120=x	143=Å	166=ª	189=
4=♦	29=↔	52=4	75=K	98=b	121=y	144=É	167=º	190=
5=♣	30=▲	53=5	76=L	99=c	122=z	145=æ	168=¿	191=
6=♠	31=▼	54=6	77=M	100=d	123={	146=Æ	169=	192=
7=	32=	55=7	78=N	101=e	124=	147=ô	170=¬	193=└
8	33=!	56=8	79=O	102=f	125=}	148=ö	171=½	194=└
9=	34="	57=9	80=P	103=g	126=~	149=ò	172=¼	195=└
10=	35=#	58=:	81=Q	104=h	127=△	150=û	173=;	196=
14=♪	36=\$	59=;	82=R	105=i	128=Ç	151=ù	174=«	197=└
15=☼	37=%	60=<	83=S	106=j	129=ü	152=ÿ	175=»	198=
15=☼	38=&	61==	84=T	107=k	130=é	153=Ö	176=	199=
16=▶	39='	62=>	85=U	108=l	131=â	154=Ü	177=	200=
17=◀	40=(63=?	86=V	109=m	132=ä	155=ç	178=	201=
18=↑	41=)	64=@	87=W	110=n	133=à	156=£	179=	202=
19=!!	42=*	65=A	88=X	111=o	134=å	157=¥	180=	203=
20=¶	43=+	66=B	89=Y	112=p	135=ç	158=Pts	181=	204=
21=§	44=,	67=C	90=Z	113=q	136=ê	159=f	182=	205=
22=—	45=-	68=D	91=[114=r	137=ë	160=á	183=	206=
23=↑	46=.	69=E	92=\	115=s	138=è	161=í	184=	207=
24=↑	47=/	70=F	93=]	116=t	139=ï	162=ó	185=	208=

Arquivos

Arquivos são conjuntos de bytes armazenados em alguma mídia (dispositivo de armazenamento permanente) que representam uma informação, como uma imagem, um texto ou uma música. A extensão do arquivo é dada em função do seu conteúdo. Por exemplo:

TXT, HTM, BAT – representa um arquivo texto
DOC – representa um documento do MS Word
PCX, BMP, JPG, GIF, TIF – formatos de imagem
MPG, AVI – formatos de vídeo
XLS – planilha do MS Excel
PPT – apresentação do MS PowerPoint
EXE, DLL – programa executável

Arquivos no formato Texto

Em um arquivo texto, toda informação é tratada como sendo um conjunto de caracteres, ou seja, um conjunto de bytes **individuais**. Os arquivos texto não possuem uma estrutura como ocorre com arquivos do tipo binário. São exemplos de arquivos texto arquivos que possuem as seguintes extensões: txt, C, pas, html, htm, tex, cpp, ini, dentre outras. Qualquer editor de texto pode manipular qualquer arquivo texto, visto que o arquivo não possui uma estrutura que deva ser compreendida para que a informação seja extraída. O formato de arquivo do Microsoft Word (extensão doc) não é um formato texto, pois o arquivo permite que sejam incorporados gráficos, imagens, sons, formatação ao texto, dentre outros. Qualquer editor de texto (como exemplo o Notepad), pode abrir arquivos binários, porém como a informação não está estruturada em bytes individuais, representando caracteres, o conteúdo apresentado não será legível.

Arquivos no formato Binário

São arquivos cuja principal característica é a existência de uma estrutura rígida que define como os dados estão armazenados no arquivo (tanto no cabeçalho como nos dados). Cada formato de arquivo possui uma estrutura diferente, o que exige programas específicos a cada formato de arquivo. São exemplos de arquivos binários: bmp, jpg, gif, tif, tga, png, avi, mpg, doc, xls, exe, bin, dll, ico, wmf, dentre outros. Geralmente programas que trabalham com imagens tem suporte a manipulação de vários formatos de arquivos de imagens, como no caso das extensões bmp, jpg, gif, tif, tga, png. Programas que manipulam vídeo, também podem ter suporte a mais de um formato. Também existem formatos proprietários, como xls para o Microsoft Excel, doc para o Microsoft Word, max para arquivos do 3D Studio Max, dentre outros. Programas que manipulam arquivos binários não conseguem (e não sabem) abrir arquivos de extensões que não foram incorporadas. Por exemplo, o Paint do Windows não sabe abrir um arquivo .txt ou um arquivo .mpg. Uma mensagem de erro será dada.

Exemplo de um arquivo binário: Para melhor compreender a estrutura de um arquivo binário, vamos criar um formato de arquivo para representar imagens, neste caso batizado com a extensão XYZ. Toda imagem é composta por uma matriz de pixels. Neste exemplo, cada pixel pode assumir 16777215 cores possíveis, ou seja, 16 milhões de cores. Como cada pixel é representado por três componentes (RGB – Vermelho, Verde e Azul), cada uma destas 3 componentes vai ser armazenada em 1 byte, totalizando assim 24 bits/pixel, ou seja, $2^{24} = 16777215$ combinações de cores por pixel. O cabeçalho do arquivo têm a seguinte estrutura:

- Os primeiros 2 bytes do arquivo devem conter o número 7101, para indicar que o arquivo é do tipo xyz. Este valor foi definido de modo aleatório.
- O terceiro byte do arquivo vai indicar a largura da imagem, que pode ser no máximo 255;
- O quarto byte do arquivo vai indicar a altura da imagem, que pode ser no máximo 255;
- Os bytes 5 e 6 armazenam o número que indica o tamanho em bytes do arquivo;

- Os bytes subsequentes, em grupos de 3, representam cada pixel da imagem.

Na seguinte figura, ilustra-se um exemplo de arquivo xyz que contém uma imagem com 2 x 3 pixels. Os primeiros 6 bytes representam o cabeçalho do arquivo e os demais 18 a informação dos pixels.

7101	2	3	24	200	1
100	200	40	10	44	255
40	90	10	0	222	20

Neste exemplo utilizaram-se grupos de bytes inteiros para representar a informação. Porém é possível e muito comum em diversos formatos de arquivos encontrar bits individuais ou em grupos para representar campos do arquivo (o mesmo vale para programas armazenados na memória RAM). Por exemplo, um byte qualquer de um arquivo pode estar organizado da seguinte forma: os primeiros 4 bits podem representar a versão do arquivo, os próximos 3 a codificação de cores e o último se o arquivo é monocromático ou colorido.

Material Adicional

1 – Código Binário

- <http://www.digitaldata.com.br/machines/codigos.htm>
- <http://www.widesoft.com.br/users/virtual/parte6.htm>

2 – Conceitos

- <http://www.infowester.com/bit.php>
- <http://www.cultura.ufpa.br/dicas/progra/arq-uni.htm>
- <http://www.jornadapc.hpg.ig.com.br/Byte%20e%20Bit.htm>

3 – Exemplos de Cabeçalhos

- <http://astronomy.swin.edu.au/~pbourke/dataformats/bmp/> (BMP)
- <http://www.freesoft.org/CIE/Course/Section4/8.htm> (TCP)
- <http://www.networksorcery.com/enp/protocol/tcp.htm> (TCP)
- <http://www.protocols.com/pbook/tcpip2.htm> (IP)