

m3_SI2025.txt

este arquivo: m3_SI2025.txt

=====
TEMAS e MATERIAIS PARA ESTUDO - SI2025

O 'mesmo' não exige qualquer referência, 'já está aí'. E se não estiver, estabelecer um "mesmo" significa, a rigor, "representar a negação de uma diferença", como quando dizemos "A igual a B".

Mas qualquer - a partir da primeira - alteração observada no 'mesmo' instala um regime de diferenças, um 'antes' e um 'depois', um 'outro' além do 'mesmo', algo que 'não é' diante do que 'é'.

Observados ou considerados ao mesmo tempo, o sim-não, o antes-depois e quaisquer outras oposições "binárias" podem representar ambiguidade, a dualidade, a dúvida, o dúbio, a divisão, a distinção e a divergência.

E em outros idiomas também, "doubt", "doute", "zweifel" (zwei = 2).

Mas os "pares contraditórios" podem permitir a síntese de outra coisa, positivar uma "classe": "quente-frio" representando "temperatura", "guerra-paz" significando "política", "zero-um", ou "dois", a "diferença".

Desde tal ponto de vista, "analítico-sintético", por assim dizer, nossa vida depende muito mais do "dois" do que do "dez", e isso bem antes da existência de qualquer sistema de numeração.

Podemos dizer que o binário aponta para a "diferença essencial".

O BI(nary digi)T significa tanto a classe ("posição" binária que pode assumir dois valores ou conteúdos "opostos"), como o valor "um", "marca" considerada como "positividade" na hierarquia dos conteúdos possíveis.

Assim, se diz que o BIT foi "marcado" (ou setado, set), querendo dizer que a posição binária passou a conter "um". Ou "desmarcado" (unset), passando a valer "zero".

"Conter" e "valer" são aqui intercambiáveis e têm geralmente o sentido de "existir".

Marcamos (um) "o que interessa" na caixa de seleção, no botão de opção ou na lista de verificação (checklist); e desmarcamos (zero), ou deixamos "em branco", "vazio", o que "não interessa".

Como vimos na aula anterior (m2_SI2025), as correspondências entre binários, decimais e hexadecimais codificando símbolos ascii indicam que qualquer sistema de numeração pode servir para representar diferenças e, portanto, sequências arbitrárias de símbolos.

Sequências ordenadas de símbolos podem ser agrupadas e rotuladas para serem transferidas, copiadas ou alteradas como uma unidade de informação. Arquivos, por exemplo, são unidades de informação desse tipo.

Nas transferências de informação deve ser preservada a ordem dos símbolos e sequências.

Como devem ser mantidos inalterados os sucessores e os antecessores de cada símbolo nas transferências e nas cópias, considerar a direção convencional da ordenação é importante.

Podemos "escrever" da direita para a esquerda, ou vice-versa. Podemos representar grupamentos de dígitos apresentando primeiro os mais significativos e depois os menos significativos; ou vice-versa.

Um 'byte' é uma grandeza que representa (hoje) um grupamento ordenado de 8 (oito) bits, ou dois grupos ordenados de 4 (quatro) bits (Chamamos um grupo de 4 bits de "nibble", 1 byte tem 2 nibbles).

Um byte corresponde portanto a dois algarismos hexadecimais, já que, como vimos na aula anterior (m2_SI2025), um hexadecimal corresponde a 4 bits (a 1 nibble).

Podemos escrever bytes da direita para a esquerda, ou da esquerda para a direita, fazendo o byte mais significativo anteceder o menos significativo (big-endian format), ou o byte menos significativo preceder o mais significativo (little-endian format).

Por exemplo, o valor 1024, que pode estar representado em binário na máquina como a sequência de dois bytes 0000 0100 0000 0000b (representação equivalente hexadecimal 04 00h), também poderia aparecer como 0000 0000 0000 0100b, ou 0004h, isto é, com byte menos significativo (0000 0000b ou 00h) antecedendo o mais significativo (0000 0100b ou 04h).

Em outras palavras, podemos escrever (mil e vinte e quatro) assim: 0400h no formato big-endian; ou 0004h no formato little-endian, byte a byte.

Outras sequências que poderíamos imaginar estarem sempre em certa ordem apresentam variações, por assim dizer, inesperadas.

RGB - Red Green Blue são uma tríade de valores que podem variar entre zero e 255 cada, ou, em hexadecimal, entre 00h e FFh.

A tríade é usada para representar intensidades de componentes de cores, por exemplo, nos arquivos com extensão .bmp, ou .dib (device independent bitmap).

Um arquivo bitmap com cores de 24 bits (sem compressão) usa $24 : 3 = 8$ bits (1 byte) para cada componente, 24 bits para a tríade RGB.

Podem ser representadas variações (diferenças) de intensidade de cores de 000000h a FFFFFFFh (aqui, na ordem RGB, dois hexadecimais para cada cor). Vai-se do mais escuro 000000h para o mais claro FFFFFFF, da ausência para a maior intensidade de cores.

Um arquivo .bmp tem dois cabeçalhos simples, um para os meta-dados do arquivo, e o seguinte para os meta-dados da imagem: entre que posições estão os cabeçalhos, a partir de que posição começa a sequência das tríades de cores, qual o tamanho do arquivo, bits por tríade, dimensões horizontal e vertical da imagem, se há ou não compressão, etc.

Após os cabeçalhos, vem a sequência de tríades.

Mas, em um arquivo .bmp, temos triádes BGR, e não RGB; e as últimas linhas da imagem vêm representadas antes das primeiras. (Verificar mais detalhes em <https://pater.web.cip.com.br/MaquInfo/ch10.html#idp1050001>)

Em formatos de arquivo e estruturas de dados, em geral, é sempre prudente ter atenção com a ordem das representações: big/little-endian, ordem direta ou inversa.

Hoje os computadores mais modernos trabalham indiferentemente nos dois formatos (bi-endian format), mas até pouco tempo atrás muitos computadores trabalhavam em apenas um dos dois sistemas.

Sequências binárias são tudo o que um computador digital pode manipular, isto é, alterar (transformar certas entradas em outras saídas), armazenar ou transferir (substituir ou copiar informações).

Para serem acessíveis, essas sequências binárias devem residir em algum meio de armazenamento [memória de acesso aleatório RAM, em geral volátil ou de curta persistência; ou memória permanente, de massa (mass storage)].

No caso dos computadores digitais de programa armazenado em memória, ou computadores de arquitetura Von-Neumann, como são chamados, que são os mais comuns (main-frames, desktops, laptops, smartphones, que conhecemos), os programas e os dados devem estar na memória principal, de acesso aleatório, R(andom)A(ccess)M(emory), para poderem ser tratados pelo processador da máquina. Isto é uma característica da arquitetura Von-Neumann.

As redes de computadores, portanto, devem fazer circular suas mensagens (dados) como sequências binárias, ou sinais codificáveis como sequências binárias, de modo que os computadores digitais binários integrantes daquelas redes possam receber, alterar, armazenar, transferir, enfim, interagir com as mensagens.

Assim também, quaisquer dispositivos controlados por computador, impressoras, monitores de vídeo, reprodutores de áudio, eletrodomésticos, instrumentos, máquinas, etc., devem poder receber e transmitir informações codificáveis como sequências binárias.

Em geral, essas sequências, como dissemos acima, são agrupadas em unidades de armazenamento, ou arquivos, para que possam ser rotulados (nomeados), guardados e processados.

Mesmo os assim chamados fluxos contínuos (streams) de dados que um computador recebe, como numa transmissão de vídeo, ou de mensagens "instantâneas", devem ser guardados como fragmentos identificados de sequências de bits, pacotes ou arquivos.

Portanto, modificar a aparência de uma imagem, significa mudar certos padrões de bits nos arquivos - como os de formato bitmap que vimos - que armazenam as sequências que contêm informações suficientes para a geração da imagem.

Podemos considerar arquivos quaisquer sequências binárias existentes em qualquer intervalo de memória identificado, independentemente do dispositivo de armazenamento (memória principal, memória de massa, dispositivos remotos).

Imagine agora todo processamento, entrada, saída, cômputo e movimentos de dados ocorrendo em um espaço que chamaremos de

'bitsfera'.

Essa "esfera s gnica bin ria", por assim dizer,   um lugar onde se transformam, de onde partem e para onde v o todas as representa es digitalizadas (numerizadas).

Sa da de dados, entrada de dados, acesso, leitura, escrita, calcular, transferir, comunicar, representam e constituem, por um lado, movimentos na bitsfera, mudan as de um lugar para a outro, replica es, altera es de padr es.

Por outro lado, s o termos cujos significados n o se esclarecem fora de suas rela es m tuas, fora do sistema ub quo e universalizante que descrevem.

Os conte dos da bitsfera s o padr es bin rios (bits, bytes), fragmentos limitados que coexistem em um mesmo espa o de interconex es.

Aqueles padr es s o amb guos, passivos e ativos, ao mesmo tempo dados e programas do mesmo regime, potencial ou efetivo de processamento e de transforma es.

Passivos, pois s o dados processados - pela m quina digital universal - sob o controle de programas.

Ativos, uma vez que programas nada mais s o do que dados que controlam o processamento de outros dados.

Nota:

O 'byte' n o faz parte do Sistema Internacional de Medidas. O 'octeto'   a unidade oficial que representa um agrupamento de oito bits. Mas 'byte'   um termo muito mais utilizado do que 'octeto'.

O byte j  representou "o n mero de bits suficiente para representar um caracter de texto em um computador". Bytes de seis, sete, oito e nove bits foram muito usados nos anos 60.

Isso dependia do peda o de mem ria que podia ser acessado como unidade em um computador.

Antes de se usar 'byte', foi comum o termo "s laba" (syllable, ou 'slab') para falar de palavras de 2 a 10 grupos de bytes-de-seis-bits cada.

(Ver <https://en.wikipedia.org/wiki/Byte>)

=====

fim_de_m3_SI2025.txt