

### 3.3 – Representação de Informações

Nossa visão em relação aos sistemas computacionais deve ser focada em duas grandes frentes:

- **Hardware** dos sistemas computacionais e
- **Software** dos sistemas computacionais

Nos itens 3.1 e 3.2 de nosso roteiro, descrevemos aspectos do hardware dos sistemas computacionais, mas em nossas aulas anteriores já havíamos falado sobre a relação entre o software (os programas, os comandos e demais controles dados ao hardware) e o hardware. Em relação ao software, vale à pena citarmos três ocorrências de software dentro dos sistemas computacionais que se distinguem entre si:

- **Sistema Operacional** - Utilizado para gerenciar o hardware do sistema computacional;
- **Firmware** - Programas utilizados para auxiliar o sistema operacional no acionamento de dispositivos de hardware e que se encontram nesses dispositivos ou em outros locais no hardware do sistema computacional;
- **Programas de uso geral** – Software criado para desempenhar alguma função no sistema computacional diferente das citadas anteriormente.

Essas três modalidades de códigos (ou programas) foram projetadas pelos seres humanos (originalmente), dessa forma, necessitam ser traduzidas da **linguagem natural humana** para algum tipo de “**dialeto**” (ou linguagem de programação). Então esse **dialeto** deverá ser interpretado ou traduzido pelos compiladores (ou interpretadores) para a “**linguagem de máquina**” (código binário) ou algum código reconhecido pelo sistema computacional, como citamos em nosso primeiro roteiro. Uma vez entendido que o pensamento humano deverá ser traduzido/interpretado algumas vezes até chegar ao código de máquina, agora cabe definirmos como será a representação das informações, tanto dentro quanto fora dos sistemas computacionais. Esse será o objeto do assunto a seguir.

#### O bit, o caractere, o byte e a palavra

Vimos anteriormente que os sistemas computacionais tiveram início através de soluções manuais, mecânicas, eletromecânicas, eletrônicas discretas (transistorizadas) e finalmente como os circuitos integrados (chips). Em nosso estudo sobre a evolução histórica dos sistemas computacionais pudemos ver que a tecnologia avançou em direção aos sistemas binários (Zeros e Uns) devido a sua facilidade de representação e determinação em termos eletrônicos/matemáticos/lógicos.

Dessa forma, o **bit** é a menor porção de informação/dado a ser representada dentro de um sistema computacional.

O bit pode representar estados:

- ligado/desligado
- sim/não
- verdadeiro/falso

O bit pode ser utilizado como dígito para o sistema numérico binário:

- $00_{\text{binário}} = 0_{\text{decimal}}$  ou  $00_2 = 0_{10}$
- $01_{\text{binário}} = 1_{\text{decimal}}$  ou  $01_2 = 1_{10}$
- $10_{\text{binário}} = 2_{\text{decimal}}$  ou  $10_2 = 2_{10}$
- $11_{\text{binário}} = 3_{\text{decimal}}$  ou  $11_2 = 3_{10}$
- :

No exemplo anterior, representamos os números decimais (base dez) através de dois dígitos binários (dois bits). Dessa forma fica fácil entender que será necessário mais de um bit para representar os números decimais. Assim, **caracteres** da linguagem humana e quaisquer outros símbolos necessitarão de conjuntos de bits para serem representados.

Exemplo 01: O código BCD (*Binary Coded Decimal*) utiliza quatro dígitos binários para codificar um dígito decimal, como segue:

- $0_{\text{decimal}} = 0000_{\text{BCD}}$
- $1_{\text{decimal}} = 0001_{\text{BCD}}$
- $2_{\text{decimal}} = 0010_{\text{BCD}}$
- $\vdots$
- $9_{\text{decimal}} = 1001_{\text{BCD}}$
- $10_{\text{decimal}} = 0001\ 0000_{\text{BCD}}$
- $11_{\text{decimal}} = 0001\ 0001_{\text{BCD}}$
- $\vdots$

Até o número 9 decimal a codificação BCD é igual à codificação binária natural. A partir do número 10 a codificação BCD irá necessitar de mais quatro dígitos binários. Veja como fica a codificação binária natural:

- $0_{\text{decimal}} = 0000_2$
- $1_{\text{decimal}} = 0001_2$
- $2_{\text{decimal}} = 0010_2$
- $\vdots$
- $9_{\text{decimal}} = 1001_2$
- $10_{\text{decimal}} = 1010_2$
- $11_{\text{decimal}} = 1011_2$
- $\vdots$

A codificação BCD é mais fácil de ser codificada e lida pelos humanos, porém é menos econômica em termos de dígitos que a codificação binária natural.

Exemplo 02: Outro exemplo é o código ASCII (*American Standard Code for Information Interchange*) ou "Código Padrão Americano para o Intercâmbio de Informação". Este código foi projetado originalmente para trabalhar com sete dígitos binários e contém caracteres "visíveis" (imprimíveis) e caracteres "invisíveis" (caracteres de controle). Os caracteres de controle eram utilizados para enviar comandos a equipamentos de transmissão das informações como "teletipos", telex ou para impressoras, etc. Veja alguns caracteres:

- $10_{\text{decimal}} = 000\ 1010_{\text{ASCII}} \rightarrow \text{"line feed"} \rightarrow \text{LF} \rightarrow \text{Alimenta linha}$
- $\vdots$
- $13_{\text{decimal}} = 000\ 1101_{\text{ASCII}} \rightarrow \text{"carriage return"} \rightarrow \text{CR} \rightarrow \text{retorno de carro (ENTER)}$
- $\vdots$
- $48_{\text{decimal}} = 011\ 0000_{\text{ASCII}} \rightarrow \text{"0"} \rightarrow \text{caractere zero}$
- $49_{\text{decimal}} = 011\ 0001_{\text{ASCII}} \rightarrow \text{"1"} \rightarrow \text{caractere um}$
- $\vdots$
- $57_{\text{decimal}} = 011\ 1001_{\text{ASCII}} \rightarrow \text{"9"} \rightarrow \text{caractere nove}$
- $\vdots$
- $65_{\text{decimal}} = 100\ 0001_{\text{ASCII}} \rightarrow \text{"A"} \rightarrow \text{caractere "A" maiúsculo}$
- $\vdots$
- $97_{\text{decimal}} = 110\ 0001_{\text{ASCII}} \rightarrow \text{"a"} \rightarrow \text{caractere "a" minúsculo}$

Este tipo de codificação se acomoda facilmente em um conjunto de oito bits, chamado de **byte**. Originalmente o byte foi estabelecido pela empresa IBM e se tornou padrão para a manipulação eficiente de conjuntos de bits dentro dos sistemas computacionais. Atualmente, todos os sistemas computacionais utilizam o byte e seus múltiplos como unidade básica tanto para codificação quanto para armazenamento de dados.

Após estabelecer o **byte** como unidade, este se tornou sinônimo de **caractere**, pois qualquer tipo de caractere é representado em um sistema computacional por pelo menos um byte (existem métodos de codificação onde o caractere pode ser representado por mais de um byte).

### **Kilobytes, Megabytes, Gigabytes, Terabytes...**

Uma vez estabelecido o byte como padrão para a manipulação de conjuntos de bits para entradas e saídas dos sistemas computacionais (I/O ou "*In/Out*"), devemos descrever os múltiplos do byte a fim de possibilitar sistemas que acompanhem a evolução da velocidade de transferência, capacidade de

armazenamento e mesmo o poder de processamento dos sistemas computacionais atuais.

- **Kilobyte (K)** → 1024 bytes =  $2^{10}$  (décima potencia de 2)
- **Megabyte (M)** → 1024 x K bytes =  $2^{20}$  (vigésima potencia de 2)
- **Gigabyte (G)** → 1024 x M bytes =  $2^{30}$  (trigésima potencia de 2)
- **Terabyte (T)** → 1024 x G bytes =  $2^{40}$  (quadragésima potencia de 2)
- **Petabyte (P)** → 1024 x T bytes =  $2^{50}$  (qüinqüagésima potencia de 2)

Atualmente é comum ouvir sobre sistemas de armazenamento com centenas de Gigabytes para computadores pessoais, mas num futuro próximo com certeza será possível utilizarmos sistemas com capacidade maior (com centenas de Terabytes ou até mesmo Petabytes).

#### **Finalmente: A palavra**

Após entendermos a natureza dos bits, bytes e caracteres, falta descrevermos como foi definida a **palavra** para um sistema computacional.

Em nossos sistemas de representação de linguagens naturais (humanas), a palavra é um conjunto de caracteres que expressa uma idéia, ação, etc. Para um sistema computacional, a palavra está intrinsecamente relacionada com o processamento de informações dentro do microprocessador e da CPU. Dessa forma, podemos exemplificar que sistemas baseados em microprocessadores Intel podem ter palavras de 32 bits (4bytes). Também existem novos microprocessadores com palavras de 64bits(8bytes).

Podemos entender a **palavra** como sendo uma **instrução** que pode ser recebida pelo microprocessador. Dessa forma, um programa seria composto por conjuntos de palavras e também conjuntos de dados (em forma de caracteres, bytes ou mesmo palavras).

Certamente a definição de palavra não é única, existem empresas ligadas a setores específicos dos sistemas computacionais que definem a palavra de forma diferente.