

## **SISTEMAS DE NUMERAÇÃO E REPRESENTAÇÃO DE DADOS**

Acredita-se que a criação de números veio com a necessidade de contar, seja o número de animais, alimentos, ou coisas do gênero. Como a evolução nos legou algumas características, como os cinco dedos em cada mão e cinco dedos em cada pé, seria muito natural que os primeiros sistemas de numeração fizessem uso das bases 10 (decimal) e 20 (vigesimal). O número 80 em francês, por exemplo, escrito como *quatre-vingt* (ou, quatro vezes o vinte), é remanescente de um sistema vigesimal. Computadores modernos, por outro lado, usam chaves elétricas para representar números e caracteres. Cada chave pode estar ligada ou desligada e a combinação dos estados de um conjunto destas chaves representa algo (número ou caracter). Visto que o cérebro de um computador é simplesmente um conjunto de chaves elétricas, onde cada chave possui apenas dois estados possíveis (ligada/desligada), computadores pensam usando apenas 2 dígitos: 0 e 1 (0 para desligada e 1 para ligada). Portanto, computadores se utilizam de uma forma de representação de dados para descrever números e caracteres na forma de um conjunto de 0s e 1s. Linguagens humanas usam palavras que contêm um número variável de caracteres. Computadores não possuem a capacidade de trabalhar com palavras de tamanho variável. Por isso, suas palavras (representação de caracteres e números) têm um número predeterminado de caracteres, que, na linguagem binária, são chamados de bits (binary digits). Para o melhor entendimento do funcionamento do computador é importante entender como ele representa os dados. Os caracteres sejam estes: letras, símbolos e números dependem diretamente de uma representação numérica para que estes possam ser acessados e armazenados. Não existiria a possibilidade de uma padronização e compatibilização entre os computadores se não existissem tabelas que representassem esses dados e tornassem possível sua conversão e utilização pelo sistema. A seguir veremos alguns conceitos importantes para o entendimento da importância e da utilização dos sistemas de numeração e representação de dados.

### **Sistemas de Numeração**

Todo o sistema de numeração tem uma base ou raiz. Quando falamos em sistema decimal, estamos estabelecendo que a nossa base de contagem é o número 10, pois o sistema decimal possui um alfabeto de 10 símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Este conjunto de símbolos do alfabeto define o que é chamado de base do sistema de numeração. Assim, se temos 10 símbolos, estamos trabalhando sobre a base 10. Um sistema de numeração é determinado fundamentalmente pela sua base. O sistema decimal possui a base 10, o binário base 2 e o hexadecimal base 16. A base é importante porque permite o entendimento do sistema de numeração utilizado e permite que sejam feitas as conversões para outras bases. O estudo dos sistemas de numeração permite um melhor entendimento sobre o funcionamento dos recursos de armazenamento, recuperação e representação de dados. No atual capítulo será feito um breve descritivo do funcionamento dos sistemas de numeração mais utilizados em informática: o decimal, o hexadecimal e o binário. O uso do sistema decimal será a base para o entendimento dos demais sistemas, por este ser o sistema de numeração mais utilizado em nosso dia-a-dia e ser usado na

apresentação de dados aos usuários Sistema Decimal O sistema decimal é o sistema de numeração normalmente utilizado na representação de valores em todo o mundo. Utiliza uma base de 10 dígitos para representar os valores quando necessário. Mas será que já paramos para pensar em como ele funciona? Devido a própria formação básica, o que fazemos na verdade é decorar a seqüência de valores. Naturalmente não nos damos conta que existe um certo sentido na colocação destes números, por exemplo: de 0 até o valor 9 (0, 1, 2, 3, 4, 5, 6, 8, 9) existem dígitos suficientes para a representação sem a repetição de símbolos, mas a partir do valor 10 percebemos a combinação do valor 1 com 0, no valor 11 a combinação do valor 1 com o 1 e assim sucessivamente. Quanto maior for o número, mais combinações irão ocorrer. Se usar a combinação sugerida anteriormente teríamos por exemplo os valores combinados entre 2 e 0 formando 20 até o valor 29, o valor 3 e 0 formando o 30 até o valor 39, o valor 4 e 0 formando o valor até chegarmos ao valor 9 combinado com o 9 formando o 99. A partir desta posição o que ocorre é que não mais combinaremos o valor 1 com o 0, pois isso já fizemos para representar o 10. Temos de combinar então o 10 com 0 para formar uma centena 100, o 20 com o 0 para formar o duzentos, o 20 com o 1 para duzentos e um e assim por diante. Conclui-se então que conforme maior for a necessidade de valores, maiores combinações irão existir e como consequência mais dígitos teremos de utilizar. Mas, existe uma outra forma de explicar isto, vejamos então: Imaginemos então que temos um valor hipotético, que poderia ser 513. Teríamos: 5 centenas, 1 dezena e 03 unidades. Apenas para relembrar: valor da unidade = 1 valor da dezena = 10 valor da centena = 100 valor da unidade de milhar = valor da dezena de milhar = valor da centena de milhar = Se utilizarmos esta regra pode-se então justificar o valor como 513 da seguinte forma: -  $5 \times 100$  (valor centena) =  $5 \times 10$  (valor dezena) =  $5 \times 1$  (valor unidade) = 5 Total 513 Ou utilizarmos a base 10 para efetivar o mesmo cálculo: -  $5 \times 10^2$  (100) =  $5 \times 10^1$  (10) =  $5 \times 10^0$  (1) = 5 Total 513 É desta forma portanto que são feitas as conversões de valores que utilizam a base de numeração 10.

Sistema Binário Como já dito anteriormente o valor binário utiliza como base ou raiz dois dígitos os valores 0 e 1. Isto significa dizer que se necessitarmos representar dois valores teremos símbolos suficientes. Mas para representar mais de dois valores? A resposta é: fazendo combinações e assim por diante... Se precisarmos converter os valores binários em decimais, usamos a mesma regra citada para valores decimais, ou seja, usamos a base 2 dos binários e multiplicamos pelos seus dígitos. Vejamos: Valor em Binário:  $1011_2$  (8) =  $8 \times 10^3$  (4) =  $0 \times 10^2$  (2) =  $1 \times 10^1$  (2) =  $1 \times 10^0$  (1) = 11 Valor em Decimal = 11 Sempre partindo do dígito menos significativo (esquerda) com o valor da base 2 0 para o mais significativo neste caso 2 3. Para ficar mais simples pode-se usar uma tabela: Decimal Base Binário = 11

Para efetivar a conversão de decimal para binário deve decompor o valor decimal e utilizar os valores restantes da divisão: Da esquerda para direita teremos então o nosso valor em binário Sistema Hexadecimal Como já visto, computadores usam o sistema binário para representar seus dados. Mas é difícil e trabalhoso para um programador descrever com conjuntos de 0 e 1 os números a serem postos na memória do computador por um dado programa. Por isso, usa-se uma forma mais compacta de representação em que os bits são agrupados de 4 em

4. Assim, cada grupo de 4 bits é transformado em um único símbolo. O sistema hexadecimal utiliza uma base com 16 dígitos para a representação de valores (0 à 15). Da mesma forma que o binário e o decimal, após atingir o número de símbolos que ele pode representar sem repetição ele então passa a fazer combinações. Uma particularidade do sistema hexadecimal é que ele utiliza letras a partir do 11º valor: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Valor 10 = A Valor 11 = B Valor 12 = C Valor 13 = D Valor 14 = E Valor 15 = F Valor 16 = 10 Valor 17 = 11 Valor 18 = 12 Valor 19 = 13 Valor 20 = 14 Valor 21 = 15 Valor 22 = 16 Valor 23 = 17 Valor 24 = 18 Valor 25 = 19 Valor 26 = 1A Valor 27 = 1B Valor 28 = 1C Valor 29 = 1D Valor 30 = 1E Valor 31 = 1F Valor 32 = 20 Para se converter um valor hexadecimal em decimal usando nosso valor 513, deve proceder da mesma forma que o binário, só que efetuando a divisão pela base do sistema hexadecimal que é 16.

Usando o valor resultante poderemos novamente converter o valor hexadecimal para decimal usando a tabela sugerida no modelo binário, apenas usando a base do hexadecimal (16)  $513_{10} = 1A1_{16}$   $1A1_{16} = 1 \times 16^2 + 10 \times 16 + 1 = 256 + 160 + 1 = 417$  Para proceder a conversão do valor 201 hexadecimal = 513 em decimal para binário deve-se converter todos os valores em binário: deve-se fazer com que todos os dígitos sejam convertidos em uma cadeia binária de 04 dígitos. Isto justifica-se pelo fato de que são necessários até 04 dígitos para representar o valor = = = Tirando a prova: Decimal Base Binário = 513. Para converter os valores de binário para hexadecimal deve-se utilizar uma cadeia de 04 dígitos (da esquerda para direita) atribuindo seu respectivo número em hexadecimal. Vejamos o valor 234 como exemplo: Em binário: Para conversão em hexadecimal converte o binário em cadeias de 04 dígitos e em seguida para o respectivo código em hexadecimal em decimal E A em hexadecimal Tirando a Prova Decimal Base Hexad E A E  $(14) \times 16 + 10 = 224 + 10 = 234$  A utilização dos sistemas decimal, binário e hexadecimal, baseia-se no fato de que os usuários só entendem em decimal, portanto os valores tem que ser apresentados em decimal (apresentação), só processam e armazenam em binário (processamento e armazenamento) pois o computador só conhece os impulsos elétricos (ligados ou desligados) e para a otimização de códigos em programação utiliza-se o código em hexadecimal.

ASCII, EBCDIC e UNICODE Sabemos que o computador só processa e armazena dígitos binários. O dígito binário é a menor unidade de informação que encontramos nos computadores. É formado como vimos anteriormente pelos valores 0 (status desligado) e 1 (status ligado). Imaginemos então que para todo e qualquer leitura de dados o computador precisasse fazer algum tipo de conversão para a geração de valores binários. Seria praticamente impossível pois todo o processamento estaria comprometido com esta conversão. Para tanto utilizam-se tabelas de conversão que fazem automaticamente a conversão dos caracteres em dígitos binários. ASCII O ASCII (American Standard Code for Information Interchange Código Padrão Americano para Troca de Informações) é utilizado na maioria dos computadores pessoais e desenvolvidos para disponibilizar informações via internet. No início de 07 bits para representar as letras e caracteres o que permitia a representação de um total de 128 caracteres destes os valores de 0-9 e as letras de A-Z e alguns símbolos, o suficiente para o padrão da língua inglesa que não utiliza acentuação. A Apple e a IBM resolveram expandir o código para 08 bits pensando no mercado mundial dos PCs. O código passou a utilizar então uma cadeia de 08 bits ou como dizemos

01 byte. Independente desta convergência os códigos do IBM e do Apple não são padronizados e por este motivo apresentam erros caso você tentar abrir arquivos do Apple no IBM. O mesmo não acontece no Apple por este já possuir um software de conversão. EBCDIC O EBCDIC (Extended Binary Coded Decimal Interchange Code Codificação Binária Estendida para Troca de Códigos Decimais), também utiliza uma cadeia de 08 bits pelos mesmos motivos do ASCII, mas com valores diferenciados e é usado principalmente por alguns computadores de grande porte chamados mainframes. UNICODE É um padrão para codificação de caracteres que utiliza uma cadeia de 16 bits o que possibilita representar mais de caracteres e isto possibilita que sejam contemplados idiomas não possíveis pela limitação do ASCII e do EBCDIC.

## EXEMPLOS DE SISTEMA DE NUMERAÇÃO OU TRANSFORMAÇÃO DE BASE

**TABELA 01: Tabelas de Valores**

N.º Decimal 10	N.º Binário 2	N.º Hexadecimal 16	N.º Octal 8
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13

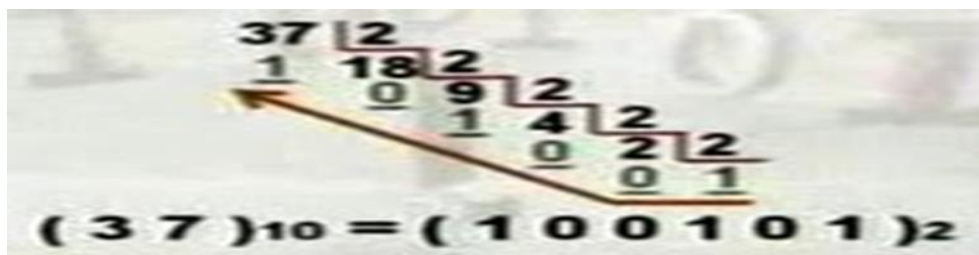
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Fonte: Autor/DACC

## • De Decimal para Binário/Octal/Hexadecimal

Nas conversões da base 10 pra as bases 2, 8 e 16, utiliza-se o mesmo método, tanto pra a parte inteira, como para a parte fracionária.

- **Parte Inteira:** São feitas divisões sucessivas pela base até que o quociente seja zero(0) (ou o dividendo menor que o divisor), tomando-se os restos na ordem inversa.



## • Binário para Hexadecimal

- **Parte Inteira:** Formar grupos de "4" dígitos à esquerda, complementando-se com zeros.

Ex:  $11001011101,1111011_2 = ?_{16}$

$\underbrace{0110}_6 \underbrace{0101}_5 \underbrace{1101}_D \underbrace{1111}_F \underbrace{0110}_6 = 65DF6_{16}$

## • Binário para Octal

- **Parte Inteira:** Formar grupos de "3" dígitos à esquerda, complementando-se com zeros.

Ex:  $010101,010100_2 = ?_8$

$\underbrace{010}_2 \underbrace{101}_5 \underbrace{010}_2 \underbrace{100}_4 = 25,24_8$

## • Para Base 10 (Decimal)

Nas conversões para a base 10, basta usar o **TEOREMA FUNDAMENTAL DA NUMERAÇÃO (TFN)**:

$$N_{10} = \sum_{i=-m}^{n-1} d_i \times b^i$$

onde:

N = número equivalente na base 10;

d = dígito;

b = base (em outro sistema de numeração);

i = índice do dígito ou expoente da base "b";

m = quantidade de dígitos à direita da vírgula;

n = quantidade de dígitos à esquerda da vírgula

## Binário para Decimal

$101,01_2$

$b=2$   
 $m=2$   
 $n=3$

$$N_{10} = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$
$$= 4 + 0 + 1 + 0 + 0,25$$
$$= 5,25$$

## Octal para Decimal

$1011_8$

$b=8$   
 $n=4$

$$N_{10} = 1 \times 8^3 + 0 \times 8^2 + 1 \times 8^1 + 1 \times 8^0$$
$$= 512 + 0 + 8 + 1$$
$$= 521$$

## Hexadecimal para Decimal

$1111_{16}$

$b=16$   
 $n=4$

$$N_{10} = 1 \times 16^3 + 0 \times 16^2 + 1 \times 16^1 + 1 \times 16^0$$
$$= 4096 + 0 + 16 + 1$$
$$= 4113$$

### Referência

UNIVERSIDADE DO CONTESTADO / UnC CAMPUS CONCÓRDIA/SC CURSO DE SISTEMAS DE INFORMAÇÃO Fundamentos em Informática (Sistemas de Numeração e Representação de Dados) (Apostila da disciplina elaborada pelo prof. MOACIR KICHEL, M.Sc. e gentilmente cedida para uso no curso de Sistemas de Informação).

## **EXERCÍCIOS PROPOSTOS**

- 1) 1990 - Transformar número da base decimal para base binária
- 2) 10101010 – Transformar número da base binária para base decimal, octal e hexadecimal
- 3) AB2,C – Transformar número da base hexadecimal para base decimal e base binária
- 4) 40,25 – Transformar número decimal para base Hexadecimal
- 5) 541 – Transformar número decimal para base binário
- 6) F8 – Transformar número hexadecimal para número na base decimal e octal