



Apostila de Introdução à Programação **Prof. Eduardo Corrêa**

Capítulo II – Conceitos Básicos sobre Computação

Sumário

II. Conceitos Básicos sobre Computação	2
II.1 O que é um computador?	2
II.2. Arquitetura de computadores	2
II.3 O que é sistema operacional?.....	7
II.4 A Linguagem binária	8
II.5 Bits e bytes	10
II.6 Conversão de bases	10
Exercícios propostos.....	17



Escola Nacional de Ciências Estatísticas

II. Conceitos Básicos sobre Computação

Nesta unidade, faremos uma breve pausa na programação para apresentar os conceitos fundamentais de computação. Eles são importantes para ajudar os iniciantes a se tornarem melhores programadores.

II.1 O que é um computador?

Um computador é uma máquina capaz de **executar cálculos e tomar decisões lógicas** em uma velocidade da ordem de bilhões de vezes superior aos seres humanos. Observe os exemplos a seguir:

Exemplo 1 – Um programa executado em um computador pessoal¹ leva **menos de meio segundo** para determinar todos os números primos localizados na faixa entre 1 e 1.000.000 (a propósito, há 78.498 números primos menores do que 1.000.000!).

Pense nisto: quanto tempo você levaria para determinar o mesmo resultado utilizando calculadora, lápis e papel?

Exemplo 2 – Nos dias atuais, os supercomputadores mais rápidos podem executar dezenas de trilhões de adições por segundo.

Isto representa o mesmo número de cálculos que centenas de milhares de pessoas podem realizar, se trabalharem em conjunto, no período de um ano!

Desta forma, o computador pode ser considerado como um dispositivo criado para auxiliar o homem em seu trabalho, diminuindo esforços e economizando tempo.

II.2. Arquitetura de computadores

Embora a computação seja uma área onde o progresso é rápido e ideias novas surgem a todo momento, é curioso observar que a arquitetura básica dos computadores ainda segue o modelo conhecido como “Arquitetura de von Neumann”, proposta pela equipe do pesquisador de mesmo nome na distante década de 1940s. Esta arquitetura, com seus componentes básicos e a interligação entre os mesmos, é mostrada na Figura 1. É importante registrar que embora a arquitetura básica tenha se mantido, ocorreram notáveis aumentos de velocidade e capacidade de processamento e armazenamento ao longo das últimas décadas.

¹ Experimento realizado num PC com processador i5-8265U e 8GB de memória RAM.

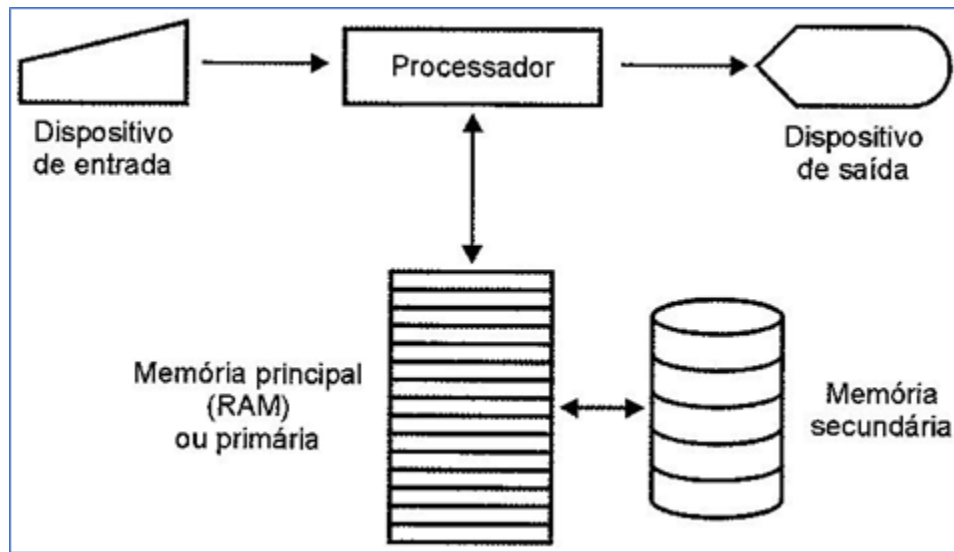


Figura 1. Componentes básicos de um computador²

Processador

Representa o componente vital do sistema de computação. Também conhecido como UCP (Unidade Central de Processamento) ou CPU (*Central Processing Unit*), é a unidade responsável pela atividade-fim do sistema: calcular, tomar decisões lógicas e processar. Devido a sua complexidade, para fins de estudo, a CPU é dividida em duas partes: Unidade Aritmética e Lógica (UAL) e Unidade de Controle (UC).

O processador é o responsável por entender e executar as instruções que formam os programas. Esta função é chamada de **função de processamento**. Quanto mais rápido o processador executar estas instruções, mais rápida será a execução dos programas. A UAL é a principal parte da CPU (e do sistema de computação em si) envolvida nessa função.

Outra responsabilidade da CPU é controlar todo o funcionamento do sistema (esta função é chamada de **função de controle**). Em decorrência da interpretação de uma determinada instrução, ela emite os sinais de controle para os demais componentes do computador agirem e realizarem alguma tarefa. Por exemplo, se o programa tem uma instrução que solicita a gravação de dados em um *pen-drive*, a CPU deve enviar sinais de controle para este dispositivo. A UC é a parte da CPU responsável pela função de controle.

A CPU executa instruções dentro de intervalos denominados **ciclos de relógio**. O relógio é um dispositivo gerador de pulsos cuja duração é chamada de ciclo. A unidade de medida utilizada para a frequência de relógios é o hertz (Hz), que significa 1 ciclo por segundo. As frequências dos computadores atuais são muito elevadas, da ordem dos bilhões de hertz (GHz). A frequência representa uma importante medida de desempenho para uma CPU (embora não seja a única medida utilizada). Para compreender esta afirmação imagine um computador que tem frequência igual a 1 GHz. Isto significa que seu relógio vibra 1

² Figura retirada do livro “Introdução à Organização de Computadores”, de M. Monteiro, Ed. LTC, 5ª Edição (2012).



Escola Nacional de Ciências Estatísticas

bilhão de vezes em um segundo. Então, se a CPU for capaz de executar uma instrução a cada ciclo, o número total de operações por segundo será igual a 1.000.000.000 (um bilhão).

CPUs Multicore

No passado, as CPU's dos computadores eram formadas por um grande número de chips, distribuídos ao longo de uma ou diversas placas. Com o passar do tempo, graças ao desenvolvimento da microeletrônica, começou a ser possível construir toda uma CPU em um único **chip**, denominado **microprocessador**.

A partir de 2006, houve uma evolução ainda mais notável: as companhias começaram a desenvolver microprocessadores com **múltiplos processadores** por chip. Com isto, as tarefas a serem realizadas por um programa podem ser divididas em subtarefas e as múltiplas CPUs as podem processar em paralelo, isto é, ao mesmo tempo (contanto que o programa tenha sido desenvolvido de modo a tirar proveito da possibilidade da execução paralela).

Para reduzir a confusão entre as palavras “processador” e “microprocessador”, as companhias referem-se aos processadores como **cores** (núcleos) e os microprocessadores com múltiplos processadores são chamados de **microprocessadores multicore** (ou CPU's multicore). Por exemplo, os microprocessadores da família Intel i5, dependendo do modelo, são dual-core (dois núcleos) ou quad-core (quatro núcleos).

Uma comparação entre os microprocessadores da família Intel pode ser obtida em:
<https://olhardigital.com.br/noticia/i3-i5-e-i7-entenda-de-uma-vez-a-diferenca-entre-os-processadores-da-intel/66293>

Memória Principal

Conceitualmente a memória principal (MP) é um componente muito simples: representa um **depósito** onde **informações são guardadas** e transmitidas para outras unidades, especialmente para a CPU. Existem dois tipos diferentes:

- **RAM** (Random Access Memory): volátil, seu conteúdo é apagado quando o computador é desligado.
- **ROM** (Read Only Memory): fixa, só pode ser lida pelo computador. Utilizada para gravar de forma permanente tabelas e programas fixos, que possuem finalidades especiais em computadores (ex: programa do SETUP nos PC's).

Devido a sua maior relevância, a **memória RAM** será a única abordada na apostila. A importância da memória RAM é justificada pelo seguinte fato: para que qualquer programa possa ser executado (ex.: editor de texto, jogo, planilha, um programa Python, etc.), ele precisa inicialmente ser carregado na memória RAM. E os dados que esses programas manipulam (ex.: textos e imagens) também precisam estar na RAM. O conteúdo de variáveis utilizadas em programas de computadores também é armazenado na memória RAM. Cada informação é guardada numa posição ou lugar específico da memória, chamado **endereço** de memória (Figura 2).



E	000000
N	000001
C	000002
E	000003
...	
	FFFFFFE
	FFFFFFF

Figura 2. Endereços de Memória

Observe que cada endereço possui um número único. No exemplo acima podem ser vistos os seguintes números de endereço: 000000, 000001, 000002, 000003, FFFFFE e FFFFFFF (letras como “E” e “F” aparecem nos números dos endereços porque estes são representados em notação **hexadecimal** – assunto que ainda será tratado ainda neste capítulo). De maneira simplificada, pode-se dizer que, dentro de cada endereço de memória, é possível armazenar **um byte**. Tipicamente, um **caractere** (ex.: uma letra, dígito ou símbolo de pontuação) representa um byte³.

Retornando ao exemplo da Figura 2, veja que o caractere “E” está armazenado no endereço de número 000000. Dizemos então que o conteúdo do endereço 000000 é “E”. Da mesma forma, o conteúdo armazenado no endereço 000001 é “N”. Note ainda que, no exemplo, os endereços FFFFFE e FFFFFFF encontram-se com o conteúdo vazio.

A quantidade de informação que pode ser armazenada na memória é chamada de **capacidade** ou **tamanho** da memória. Esta capacidade é usualmente expressa quantidade de **bytes**. Mais precisamente, o tamanho da memória costuma ser apresentado em algum múltiplo de 2^{10} bytes, conforme indicado a seguir:

- 1 Kbyte (1 kilobyte) = 2^{10} bytes = 1024 bytes
- 1 Mbyte (1 megabyte) = 1024Kbytes = 2^{20} bytes.
- 1 Gbyte (1 gigabyte) = 1024Mbytes = 2^{30} bytes.
- 1 Tbyte (1 terabyte) = 2^{40} bytes.
- 1Pbyte (1 petabyte) = 2^{50} bytes.
- 1Exbyte (1 exabyte) = 2^{60} bytes.
- 1Zbyte (1 zetabyte) = 2^{70} bytes.
- 1Ybyte (1 yottabyte) = 2^{80} bytes.

³ Não é exatamente assim, pois muitos caracteres acentuados e símbolos ocupam mais de 1 byte em esquemas de representação modernos, como o UTF-8. Porém, para efeito de simplificação, nesta disciplina consideraremos que um caractere ocupa 1 byte.



Escola Nacional de Ciências Estatísticas

Atualmente, é comum encontrar computadores pessoais (PC's) equipados com memória RAM com capacidades como 4GB, 8GB, 16GB e até acima destes valores. Ainda nesta unidade será explicado o motivo pelo qual os computadores utilizam potências de 2 ao invés de potências de 10 em seu sistema métrico. Prosseguindo com o estudo sobre o conceito de capacidade de memória, analise o exemplo a seguir.

Capacidade de Memória

Considere um computador antigo, que possuía memória RAM com capacidade igual a 256 KB. Qual a quantidade máxima de caracteres que esta memória podia armazenar?

Resposta: 256 KB representa 256×2^{10} bytes = 256×1024 bytes = 262.144 bytes. Como 1 caractere ocupa 1 byte, então a memória do computador antigo podia armazenar 262.144 caracteres.

Conceitualmente, a RAM é um componente muito simples que aceita apenas duas **operações**: READ (leitura) e WRITE (escrita).

- **READ:** recuperar uma informação num local (endereço) da memória.
- **WRITE:** gravar uma informação num local (endereço) da memória.

Para facilitar o entendimento considere a analogia com uma biblioteca. Na biblioteca a operação WRITE significaria guardar um livro na estante/prateleira adequada (ENDEREÇO). A operação READ, por sua vez, significaria pegar um livro emprestado. No entanto há uma diferença importante: nos computadores a ESCRITA é uma **operação destrutiva**: quando o WRITE é feito em um endereço de memória que está preenchido, o conteúdo anterior é apagado e substituído por um novo conteúdo.

Além da capacidade, o **tempo de acesso** representa outra importante medida de desempenho para as memórias de computador. Essa medida indica quanto tempo a memória gasta para ler ou escrever uma informação em um endereço. Melhor explicando: trata-se do período de tempo decorrido desde o instante em que foi iniciada a operação de acesso (quando a origem – em geral a CPU – passou o endereço de acesso para o sistema de memória), até que a informação requerida (instrução ou dado) tenha sido efetivamente transferida. Nos PC's atuais, as memórias possuem tempo de acesso da ordem de 50-70ns (nanossegundos).

Memória Secundária

A memória secundária representa a seção de armazenamento de “grande capacidade” e longo prazo do computador. Ao contrário da memória RAM, os dados gravados em memória secundária não são voláteis, ou seja, eles permanecem gravados no computador mesmo quando ele é desligado.



Escola Nacional de Ciências Estatísticas

Os conceitos de endereço, capacidade e tempo de acesso também são válidos para a memória secundária. Neste sentido, este tipo de memória caracteriza-se por ter baixa velocidade (tempo de acesso alto) e grande capacidade de armazenamento (além de ser bem mais barata que a memória RAM). As duas principais tecnologias utilizadas nos PCs para implementar a memória secundária são os tradicionais discos magnéticos (HD – *hard disk*) e os mais modernos Solid State Drives (SSD).

O quadro a seguir apresenta um comparativo entre as propriedades da memória principal e da memória secundária.

Memória Principal - RAM

Capacidade Típica nos PC's: 2GB a 64GB

Tempo de Acesso: 50 a 70ns (10^{-9} s).

Preço de 1GB \approx R\$ 35,00

Memória Secundária

Capacidade Típica nos PC's: 500GB ou mais

Principais tipos: **Disco Magnético** ou **SSD**.

Tempo de Acesso: 5 a 20ms (10^{-3} s) para o disco magnético e 0,1 a 0,2ms para SSD.

Preço de 1GB \approx R\$ 0,40 (disco magnético) e R\$ 0,80 (SSD)

Unidade de Entrada

É a “seção receptora” do computador. Ela obtém informações (dados e programas) oriundas dos dispositivos ou **periféricos de entrada** (ex.: teclado, mouse, scanner, câmera, leitora de código de barras) e coloca estas informações disponíveis para outras unidades (como CPU e memória), de forma que as informações possam ser processadas.

Unidade de Saída

Esta é a “seção de expedição” do computador. Ela pega as informações que foram processadas por um programa e as coloca em algum **periférico de saída**, para torná-las disponíveis aos usuários. Alguns exemplos de dispositivos de saída: monitor de vídeo, impressora, caixa de som, entre outros.

II.3 O que é sistema operacional?

Um sistema operacional (SO) é um programa especial que tem o propósito de fornecer um ambiente no qual o usuário possa executar programas. Os SO's mais conhecidos e



Escola Nacional de Ciências Estatísticas

utilizados atualmente são Windows e Linux. O SO é o programa que está sempre executando no computador, todo resto consistindo em programas aplicativos. Ele atua de forma semelhante a um **governo** ou, utilizando uma expressão mais técnica, atua como um **alocador de recursos**. Um sistema de computação possui muitos recursos de hardware que podem ser necessários para resolver um problema: CPU, espaço em memória, dispositivos de entrada e saída. O sistema operacional atua como gerente desses recursos e os aloca a programa e usuários específicos, conforme necessário, para a execução de tarefas. Como pode haver muitos pedidos de recurso, possivelmente conflitantes entre si, o sistema operacional deve decidir em que pedidos serão alocados recursos para que ele possa operar o sistema de computação de forma eficiente e justa.

A Figura 3 apresenta um exemplo de atuação do sistema operacional. Considere um usuário que esteja executando três programas aplicativos simultaneamente: Excel, Word e Chrome. O sistema operacional é o responsável por diversas tarefas, entre elas, a alocação do espaço na memória RAM para estes três programas, a interação do Chrome com os dispositivos de rede, a ativação e desativação de uma janela de acordo com a conveniência do usuário, entre outras.



Figura 3. SO gerenciando a execução de três aplicativos

II.4 A Linguagem binária

Quando estamos realizando a leitura de um texto em Português conseguimos entender as informações apresentadas porque conhecemos o formato e o significado de **símbolos** como letras, números, sinais de pontuação, sinais matemáticos, entre outros. Temos 26 letras (sem contar caracteres acentuados), 10 algarismos (0, 1, ... 9) e diversos sinais (+, -, *, %, etc.). Além disso, sabemos as formas como as letras se unem para formar as palavras e conhecemos diversas **regras** para a construção de frases em Português (por exemplo, regras de sintaxe). Observe agora o texto reproduzido na Figura 4. Ele está escrito no idioma Grego. Os símbolos (letras) são inteiramente diferentes daqueles utilizados por nosso idioma. Com isto, é praticamente impossível para nós decodificar qualquer informação.



Escola Nacional de Ciências Estatísticas

Ὅτι οἱ ἄνθρωποι γεννιούνται ἐλεύθεροι καὶ ἴσοι στὴν αξιοπρέπεια καὶ τὰ δικαιώματα. Εἶναι προικισμένοι με ῥησική καὶ συνείδηση, καὶ οφείλουν νὰ συμπεριφέρονται μεταξύ τους με πνεῦμα ἀδελφικότητος.

Figura 4. Texto na língua Grega⁴

A única maneira de compreendê-lo é solicitando uma tradução para uma linguagem que possamos **entender diretamente**. Esta tradução é apresentada na Figura 5. Veja que agora torna-se possível saber que o texto representa uma transcrição do Artigo I da Declaração Universal dos Direitos Humanos.

“Todos os seres humanos nascem livres e iguais em dignidade e direitos. São dotados de razão e consciência e devem agir em relação uns aos outros com espírito de fraternidade”.

Figura 5. Texto traduzido para a língua Portuguesa

Um computador também possui a sua “linguagem natural”, ou seja, a linguagem que ele consegue entender diretamente. Esta linguagem é denominada **linguagem de máquina** ou **linguagem binária**. De maneira bastante diferente do Português ou do Grego, a linguagem de máquina possui apenas **dois símbolos: 0** (zero) e **1** (um). Qualquer informação que necessite ser armazenada em memória ou processada pela CPU, sejam dados ou programas, **precisa que ser traduzida** para a linguagem de máquina. Veja os exemplos relacionados abaixo:

- **Programas de computador:** um programa de computador escrito em uma linguagem como Python precisa ter cada uma de suas linhas traduzidas para a linguagem de máquina para que o computador possa entender suas instruções. No caso do Python, um software **interpretador** (Interpretador Python) é o responsável pela realização desta tarefa.
- **Representação de Números:** para que o computador possa efetuar operações aritméticas sobre dados numéricos de entrada, é preciso antes traduzir os números para o formato binário. Se um programa solicita, por exemplo, ao computador a execução da soma $5 + 8$, os números 5 e 8 precisam ser convertidos do **formato decimal** para o **formato binário** antes da conta ser realizada. Felizmente, esta conversão é realizada de forma transparente (sem que seja preciso a intervenção do usuário).

⁴ Texto retirado de <http://www.omniglot.com/writing/greek.htm> (Greek alphabet, pronunciation and language)



Escola Nacional de Ciências Estatísticas

- **Representação de Caracteres:** os caracteres digitados via teclado também precisam ser convertidos para o formato binário antes de serem armazenados em memória. Para isto, o computador utiliza uma tabela de conversão interna que mapeia cada caractere do teclado para um código binário. Ou seja, a letra 'A' possui um código binário, 'B' possui outro código, assim como 'C', 'D', e as demais letras. Dessa forma, a conversão também é realizada de forma transparente.

Em resumo, os exemplos acima indicam que toda informação que pode ser compreendida e manipulada por um computador (sejam números ou caracteres) é sempre representada por uma sequência de 0's e 1's. É importante justificar que a linguagem do computador possui esse número tão reduzido de símbolos pelo fato de que, do ponto de vista da eletrônica, é mais confiável projetar uma máquina com este tipo de característica, em que os circuitos internos precisem lidar com apenas dois valores de tensão, por exemplo 0V e +3V.

II.5 Bits e bytes

A menor unidade de informação armazenável em um computador é chamada de **bit** (*binary digit* ou dígito binário). O bit pode ter apenas dois valores: 0 ou 1. Da mesma forma que numa linguagem humana as letras são reunidas para formar palavras, dentro do computador os bits são reunidos para que possam ter algum significado útil. Um conjunto de 8 bits reunidos é chamado de **byte**. O conceito de byte foi criado com o objetivo de servir como referência para a construção e funcionamento de dispositivos de armazenamento, como a memória RAM, por exemplo. Para efeito de simplificação, podemos considerar que cada caractere ocupa, tipicamente, um byte na memória do computador. O conceito de byte também é utilizado como referência para processos de transferência de dados entre periféricos (como uma impressora) e a CPU ou memória.

Como os computadores são máquinas binárias, todas as indicações numéricas referem-se a **potências de 2** e não a potências de 10 como no sistema métrico. Por esta razão, 1Kbyte representa 1024 bytes ao invés de 1000 bytes ($2^{10} = 1024$). Da mesma forma, 1Mbyte representa 1.048.576 bytes ao invés de 1.000.000 de bytes ($1\text{Mbyte} = 2^{10} \times 2^{10} = 2^{20}$ bytes). De maneira análoga 1Gbyte representa 1024 Mbytes (2^{30} bytes).

II.6 Conversão de bases

A seguir, mostraremos como é possível realizar a conversão de números inteiros positivos do formato utilizado pelos seres humanos (decimal) para o formato utilizado pelo computador (binário). A representação de inteiros negativos e de números reais não é tratada ao longo do texto desta apostila, devido a sua maior complexidade.



II.6.1 - Notação Posicional – Base Decimal

O **sistema de numeração** que usamos é o sistema decimal, que, naturalmente é composto por dez algarismos: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Por sua vez, o sistema de numeração utilizado pelo computador é o sistema binário, formado por apenas dois algarismos: 0 e 1. A quantidade de algarismos disponíveis em um dado sistema de numeração é chamada de **base**. Os humanos representam números na **base 10**, enquanto os computadores representam números na **base 2**. No entanto um número pode ser representado em qualquer base (ex.: 5, 8, 16, etc.).

A forma mais empregada para a representação numérica é chamada de **posicional**. Nela, os algarismos componentes de um número assumem valores diferentes, dependendo de sua posição relativa no número. Observe o seguinte exemplo:

$(1973)_{10}$

Como se trata da base decimal, que é aquela utilizada pelos humanos, o indicador da base pode ser dispensado:

1973

No exemplo, o número é composto por quatro algarismos: 1, 9, 7 e 3. Note que o valor deste número pode ser obtido da seguinte forma:

$$1 \times 10^3 + 9 \times 10^2 + 7 \times 10^1 + 3 \times 10^0$$

O primeiro algarismo (mais à direita) representa 3 unidades, ou seja, 3×10^0 . O segundo algarismo mais à direita representa 7 dezenas, ou seja, 7×10^1 . O terceiro algarismo 9 centenas (9×10^2) e o quarto um milhar (1×10^3). Generalizando, num sistema qualquer de numeração posicional, um número N é expresso da seguinte forma:

$$N = (d_{n-1} d_{n-2} \dots d_1 d_0)_b$$

Onde:

d indica cada algarismo do número.

n-1 n-2, ..., 1, 0 (índice) indicam a posição de cada algarismo.

b indica a base da numeração

n indica o número de dígitos inteiros.

O valor do número pode ser obtido pelo seguinte somatório:

$$N = d_{n-1} \times b^{n-1} + d_{n-2} \times b^{n-2} + \dots + d_1 \times b^1 + d_0 \times b^0$$



Escola Nacional de Ciências Estatísticas

II.6.2 - Outras Bases de Numeração – Conversão para a Base 10

O sistema posicional é utilizado para representar números expressos em qualquer base, conforme ilustrado a seguir:

$(00010101)_2$ – número na base 2.

$(304)_5$ – número na base 5.

$(1671)_8$ – número na base 8.

$(10)_2$ – número na base 2.

Números escritos em qualquer base podem ser convertidos para a base 10 de maneira bastante simples. Observe os seguintes exemplos:

Exemplo II.1 – Conversão da Base 2 para a Base 10

Seja o seguinte número na base 2: $(1010)_2$. Para obter seu valor na base 10, basta aplicar a seguinte equação:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (10)_{10}$$

Exemplo II.2 – Conversão da Base 8 para a Base 10

Seja o seguinte número na base 8: $(62)_8$. Seu valor na base 10 é:

$$6 \times 8^1 + 2 \times 8^0 = (50)_{10}$$

Em resumo:

- 1) O número máximo de algarismos diferentes de uma base é igual ao valor da base:
 - Na base 10 existem 10 algarismos: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.
 - Na base 8 existem 8 algarismos: 0, 1, 2, 3, 4, 5, 6 e 7.
 - Na base 5 existem 5 algarismos: 0, 1, 2, 3 e 4.
 - Na base 2 existem 2 algarismos: 0 e 1.
- 2) A conversão de um número expresso em qualquer base para a base 10 é bastante simples.
 - Os exemplos II.1 e II.2 ilustraram este fato.
 - O valor, na base 10, de um algarismo **mais à esquerda** (mais significativo), de um número de n algarismos inteiros, é obtido pelo produto do seu valor absoluto pela base elevada a potência, ou seja: $d_{n-1} \times b^{n-1}$.



II.6.3 - Base 2 (Binária)

A base 2 é a de maior interesse para esta disciplina, pelo fato de ser utilizada internamente pelo computador. A Tabela 1 apresenta a correspondência entre valores nas bases 10 e 2:

Tabela 1 –Correspondência: Base 10 x Base 2

Número na Base 10	Número na Base 2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
100	1100100
1000	1111101000

Para converter um número da base 2 para a base 10, basta empregar o método recém-apresentado na Subseção II.6.2. A conversão no sentido inverso será apresentada na Subseção II.6.5.

II.6.4 - Base 16 (Hexadecimal)

Observando a Tabela 1, é possível perceber que os números representados em base 2 são muito extensos. Veja que o número $(1000)_{10}$ é representado por $(1111101000)_2$, que possui 10 algarismos! Quanto menor a base de numeração, maior é a quantidade de algarismos necessários para indicar um dado valor. Note que a manipulação visual destes números por parte de um ser humano torna-se bastante dificultada.

Por este motivo, em textos técnicos, programas e manuais de componentes de computador, muitas vezes é comum expressar números binários na **base 16** ou **base hexadecimal**. Os números de endereços de memória RAM, por exemplo, são sempre referenciados na base 16 (como fizemos na Figura 2, memória RAM). Na base 16 existem, obviamente, 16 algarismos. As letras “A”, “B”, “C”, “D”, “E” e “F” representam, respectivamente, os valores (da base 10): 10, 11, 12, 13, 14 e 15. Observe o exemplo a seguir:

$(A3F)_{16}$



Escola Nacional de Ciências Estatísticas

Para converter este número para a base 10, basta utilizar o método já apresentado. Note no exemplo abaixo que o “algarismo” A tem que ser substituído pelo valor 10 no momento do cálculo. Da mesma forma o “algarismo” F precisa ser substituído por 15:

$$(A3F)_{16} = 10 \times 16^2 + 3 \times 16^1 + 15 \times 16^0 = (2623)_{10}$$

Observe a Tabela 2 e perceba como a notação hexadecimal é bem mais compacta e agradável aos olhos dos seres humanos quando comparada à base 2. Como veremos em breve, na Subseção II.6.7, a conversão de um número na base 16 para a base 2 (a base “natural” das CPUs) e vice-versa é extremamente simples, pois 16 é uma potência de 2 ($16 = 2^4$). É exatamente devido a esse fato e por causa de sua notação compacta, que a base hexadecimal costuma ser adotada em algumas situações práticas.

Tabela 2 – Correspondência: Base 10 x Base 16 x Base 2

Número na Base 10	Número na Base 16	Número na Base 2
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
100	64	1100100
1000	3E8	1111101000

II.6.5 - Conversão da Base 10 para outras Bases

Essa subseção apresenta a “receita” para a conversão de números na base 10 para outras bases. Basta utilizar o seguinte método:

- A conversão de números representados na base 10, para seus valores em uma base B qualquer, é obtida dividindo-se o número decimal pelo valor da base desejada; o **resto** encontrado é o valor do algarismo menos significativo na base B (dígito mais à direita). Em seguida divide-se o quociente encontrado pela base B; o valor encontrado no resto é o algarismo seguinte. O processo continua **até que o quociente seja igual a zero**. Para compreender este método, observe os exemplos a seguir:

Exemplo II.3 – Conversão da Base 10 para a Base 2

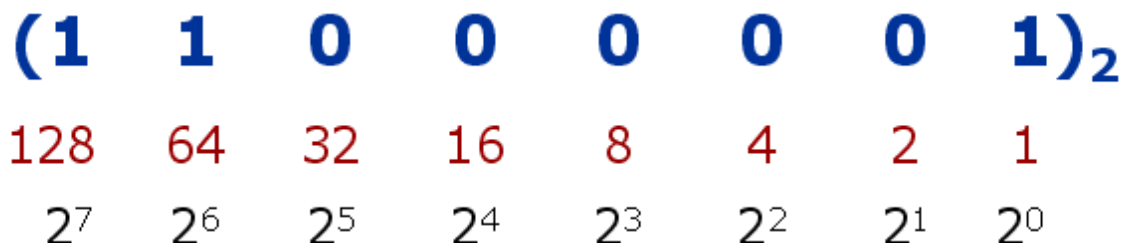
$$(45)_{10} = (?)_2$$

$45 \div 2 = 22$	resto = 1 (algarismo mais à direita).
$22 \div 2 = 11$	resto = 0 (segundo algarismo mais à direita).
$11 \div 2 = 5$	resto = 1 (terceiro algarismo mais à direita).
$5 \div 2 = 2$	resto = 1 (quarto algarismo mais à direita).
$2 \div 2 = 1$	resto = 0 (quinto algarismo mais à direita).
$1 \div 2 = 0$	resto = 1 (FIM DO PROCESSO - algarismo mais significativo encontrado).

$$(45)_{10} = (101101)_2$$


$$(64)_{10} = (?)_2$$
$$(64)_{10} = (1000000)_2$$
$$(490)_{10} = (?)_{16}$$
$$(490)_{10} = (1EA)_{16}$$

A conversão da base 2 para a base 10, introduzida na Subseção II.6.2, também pode ser realizada com o uso de um “macete” muito prático. Considere o seguinte exemplo: a conversão do número $(11000001)_2$ para a base 10. A forma mais rápida de realizar o processo consiste em utilizar o método apresentado na Figura 6.



15



Escola Nacional de Ciências Estatísticas

Descrição do Método:

- Escreva o número 1 (equivalente a 2^0) embaixo do dígito mais à direita (dígito menos significativo) do número binário que você deseja converter.
- Embaixo do segundo dígito menos significativo, escreva 2 (equivalente a 2^1). Embaixo do terceiro dígito menos significativo, escreva 4 (equivalente a 2^2) e assim sucessivamente, até chegar ao dígito mais significativo.
- Ao terminar, some todos os números que estejam escritos embaixo dos dígitos '1'. O resultado obtido corresponderá ao valor do número na base 10. No exemplo da Figura 21 esta soma fica:

$$128 + 64 + 1 = 193$$

Desta forma, tem-se que $(11000001)_2 = (193)_{10}$.

II.6.7 - Conversão Direta entre as Bases 2 e 16

O mesmo “macete” é aplicado para conversões entre estas duas bases. Como $16 = 2^4$, um número binário inteiro pode ser facilmente convertido para o seu valor equivalente na base 16. Basta dividi-lo, da esquerda para a direita em grupos de 4 bits e converter cada grupo:

Exemplo II.6 – Conversão da Base 2 para a Base 16

$$(11000001)_2 = (?)_{16}$$

$$(1100) (0001)$$

$$\text{C} \quad 1$$

$$(11000001)_2 = (\text{C}1)_{16}$$

Exemplo II.7 – Conversão da Base 2 para a Base 16

$$(10101)_2 = (?)_{16}$$

$$(0001) (0101)$$

$$1 \quad 5$$

$$(10101)_2 = (15)_{16}$$

OBS: Veja que, neste exemplo complementamos o número $(10101)_2$ com três zeros à esquerda para facilitar a sua conversão: $(00010101)_2$. Assim foi possível dividi-lo em dois grupos de 4 bits.



Escola Nacional de Ciências Estatísticas

A conversão da base 16 para a base 2 é realizada de forma semelhante, no sentido inverso: substitui-se cada algarismo em hexadecimal por seus 4 bits correspondentes.

Exemplo II.8 – Conversão da Base 16 para a Base 2

$$(A5)_{16} = (?)_2$$

$$(1010) (0101)$$

$$A \quad 5$$

$$(A5)_{16} = (10100101)_2$$

Exercícios propostos

(1) Considere um PC atual, dotado de memória RAM com capacidade igual a 4 GB. Supondo que um caractere ocupa 1 byte na memória, qual a quantidade máxima de caracteres que esta memória pode armazenar?

(2) O microprocessadores da família i5 são produzidos em diferentes modelos, cada um deles operando com uma frequência distinta. O modelo mais básico possuía frequência de 2.66GHz, enquanto outros mais sofisticados operam a 4.80GHz. Suponha que, independentemente do modelo, o i5 seja capaz de realizar uma operação de adição a cada ciclo⁵. Determine: o número total de operações de adição que os modelos que operam com frequências de 2.66 GHz , 3.60 GHz e 4.80 GHz podem realizar em um segundo

(3) Uma loja do centro da cidade comercializa um módulo de memória RAM para PC's, com capacidade de 8GB, pelo preço de R\$ 260,00. Calcule o custo por byte desta memória.

(4) Considere uma CPU antiga com frequência de 800MHz. Supondo que essa CPU seja capaz de realizar uma operação de multiplicação a cada **dois ciclos**, qual o número total de multiplicações que ela realiza em um segundo?

(5) Complete as seguintes expressões:

a) 65536 bytes = _____ Kbytes

d) 512 Mbytes = _____ bits

b) 131072 bytes = _____ Kbytes

e) 131072 bits = _____ Kbytes

c) 7 Mbytes = _____ bytes

f) 4 Gbytes = _____ Kbytes

⁵ Trata-se apenas de uma suposição. Na vida real o processador não funciona exatamente assim, além do fato de ele possuir mais de um núcleo.



Escola Nacional de Ciências Estatísticas

(6) Converter os valores decimais em valores binários equivalentes.

- | | |
|----------|---------|
| a) 64 | d) 1024 |
| b) 581 | e) 197 |
| c) 65536 | f) 99 |

(7) Converter os valores binários em decimais equivalentes.

- | | |
|---------------|-----------------|
| a) 1011100 | c) 100000 |
| b) 1111101001 | d) 101010101010 |

(8) Converter os valores decimais para hexadecimais equivalentes.

- | | |
|----------|---------|
| a) 917 | d) 622 |
| b) 65536 | e) 121 |
| c) 1024 | f) 1325 |

(9) Converter os valores hexadecimais para decimais equivalentes.

- | | |
|---------|----------|
| a) A0 | d) 800 |
| b) 3E9 | e) FFFF |
| c) 1024 | f) 14A3B |

(10) Efetue as seguintes conversões:

- a) $(110011110101)_2 = (?)_{16}$
- b) $(FEBF)_{16} = (?)_2$
- c) $(101010)_2 = (?)_{16}$
- d) $(EBA)_{16} = (?)_2$

(11) A partir do valor binário $(110011)_2$, escreva os 3 números que vem em sequência.

(12) A partir do valor $(CEF)_{16}$, escreva os 4 números subsequentes.