

UNIDADE 1 – FUNDAMENTOS DA ARQUITETURA E REPRESENTAÇÃO DE DADOS NO SISTEMA OPERACIONAL

MÓDULO 1 – ARQUITETURAS BÁSICAS DE COMPUTADORES

01

1 - CONCEITO DE ARQUITETURA DE COMPUTADORES

Prezado(a) aluno(a) estamos começando nosso estudo sobre Arquitetura e Organização de Computadores (AOC) e nesse módulo veremos o conceito de Arquitetura de Computadores, faremos uma viagem no tempo para entender a evolução dos computadores através dos séculos (Histórico) e finalizaremos fazendo uma breve descrição dos principais componentes de um computador.

Vamos começar?

O estudo da matéria Arquitetura de Computadores é de suma importância para o entendimento dos aspectos da evolução tecnológica na área, de como é feita a montagem dos componentes de um computador, provendo assim uma visão e fundamentação teórica para que os profissionais da área tenham um desempenho melhor nas atividades da TI (Tecnologia da Informação). Alguns exemplos de projetos que exigem o conhecimento da Arquitetura de Computadores: projetos de redes de computadores, suporte e desenvolvimento de aplicações, automação de estabelecimentos comerciais, implantação de sistemas informatizados e outros.

Vamos então ver algumas definições de autores, estudiosos e profissionais da área da TI (Tecnologia da Informação) sobre o que seja um computador (máquina), quais são seus componentes, e com base nisso começaremos o entendimento do que seja **arquitetura** e como é a organização de um computador. Seguem abaixo algumas definições:

02

[Tanenbaum](#) define como Arquitetura de Computadores o estudo que trata do funcionamento e também do comportamento de um sistema computacional (computador).

Provavelmente você deve estar pensando: *computador tem comportamento?* Vamos esclarecer o que venha ser este “comportamento”. Quando falamos em comportamento estamos fazendo referência à forma como os componentes do computador interagem entre si.

Quando perguntamos para um profissional da programação de *software* (programador), como ele visualiza um sistema computacional, provavelmente ele dirá que o mais importante é: plataforma 32 ou 64 bits, frequência do clock etc. Isso acontece porque, para programar, não é primordial saber como é a estrutura interna, a parte que não é “visível” aos nossos olhos para o programador (exemplo: frequência do relógio ou tamanho da memória física).

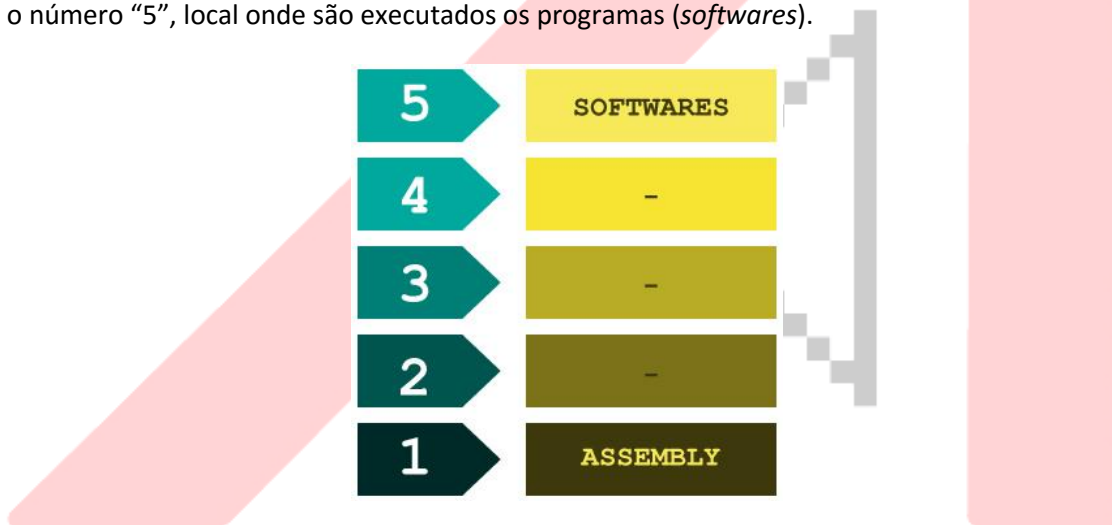
Tanenbaum

Andrew Stuart "Andy" Tanenbaum é o chefe do Departamento de sistemas de computação, na Universidade Vrije, Amsterdã nos Países Baixos. Ele é o autor do MINIX, um sistema operacional baseado no Unix com propósito educacional, e bastante conhecido por seus livros sobre ciência da computação.

03

Aproveitando a abordagem anterior em que falamos sobre os programadores, vamos também entender outro conceito, conhecido como **níveis** na arquitetura de computadores (programação de baixo e alto nível). A ideia que está por trás deste conceito é que existem muitos níveis por meio dos quais o computador pode ser “visualizado”.

Para tornar mais fácil o entendimento do conceito de nível, usaremos o seguinte exemplo: imagine uma sequência numérica em ordem crescente, começando de 0 até 5. Utilizando esse exemplo para fazer uma analogia com os **níveis** na Arquitetura de Computadores, podemos dizer que o nível “mais alto” seria o número “5”, local onde são executados os programas (*softwares*).



Continuando com o exemplo da sequência numérica, podemos dizer que “0” seria o nível “mais baixo”, é nível da programação em linguagem de máquina (*assembly*), ou o funcionamento e programação dos registradores (componente do processador).

04

No intuito de reforçar o conceito de **níveis** na arquitetura de computadores, vamos utilizar como exemplo um carro de Fórmula 1.

Vamos supor que você tenha a oportunidade de pilotar um carro de Fórmula 1. Ao entrar e sentar no “cockpit” você observa o volante e os vários instrumentos disponíveis no painel. Logo depois você sai para pista de corrida e sente a velocidade do carro.

Nesse estágio, em que você observa o painel e instrumentos, seria comparado com o nível mais alto na arquitetura de computadores, pois durante o tempo em que você “curtiu a velocidade” e todos os recursos disponíveis, provavelmente não ficou preocupado como o funcionamento do motor e dos outros componentes, uma vez que esta é uma preocupação dos mecânicos e engenheiros.

O nível do “motor e componentes” é comparado com o nível mais baixo nos “níveis de arquitetura de computadores”.



Em suma, a linguagem de baixo nível (assembly) ou **linguagem de máquina** é usada pelos engenheiros de hardware para criar instruções que permitam o melhor aproveitamento dos recursos de hardware (processador, memória e etc.).

05

A linguagem de alto nível está relacionada aos aplicativos, softwares e utilitários que não se preocupam com os recursos de hardware existente em um computador. Seguindo a comparação com o carro de Fórmula 1, seria o nosso nível, em que estamos preocupados apenas em utilizar o carro de Fórmula 1 e não nos preocupamos com o motor e outros dispositivos.

Vale destacar que este conceito não está relacionado a uma linguagem ser “melhor” ou “superior” pelo fato de ser de alto nível e o outro ser ruim por ser de baixo nível. Observe a figura abaixo para visualizar que, no nível “mais alto”, estão os aplicativos e utilitários e no nível “mais baixo” está a linguagem de máquina, microprogramação e os dispositivos físicos.



Para compreender melhor e aprofundar seus conhecimentos sobre os níveis de máquina, assista ao vídeo a seguir.

<https://www.youtube.com/watch?v=xFzwnXyt09Q>

Quando olhamos para um computador, o que nos parece é que é uma máquina formada por um conjunto de partes eletromecânicas e eletrônicas, que possuem a capacidade de armazenar, coletar, fornecer informações e manipular dados, tudo isso de forma automatizada. Também conseguimos visualizar o gabinete do computador e seus periféricos, isso é o que chamamos de **hardware**, que é uma palavra em inglês para nomear tudo aquilo que compõe fisicamente uma máquina (computador).

Quando fazemos uma pesquisa na internet sobre "arquitetura de computadores", descobrimos que esta expressão foi pela primeira vez utilizada pelos pesquisadores da IBM, Lyle R. Johnson, Muhammad Usman Khan e Frederick P. Brooks, Jr. Estes pesquisadores, em 1959, trabalhavam no departamento de organização de máquinas da IBM.

O pesquisador Johnson teve a iniciativa de fazer um relatório de pesquisa sobre o supercomputador, conhecido por **Stretch**, que foi desenvolvido pela IBM para atender a um pedido do laboratório Nacional de Los Alamos (Los Alamos Scientific Laboratory). Johnson, durante a confecção do relatório da pesquisa, chegou ao nível de detalhar formatos, tipos de instruções, parâmetros de *hardware* e as melhorias de velocidade de processamento. Ele então chegou à conclusão que havia detalhado a **arquitetura do sistema**.

Logo depois, outro pesquisador do Stretch, chamado Brooks, ao iniciar o segundo capítulo do livro chamado “Planning a Computer System: Project Stretch”, no ano de 1962, definiu, mesmo com limitações tecnológicas e econômicas:



"Arquitetura de computadores é a arte de determinar as necessidades do usuário de uma determinada estrutura".

Brooks queria dizer que as empresas, ao desenvolverem seus computadores, precisam estar atentas às necessidades dos usuários. Não basta criar um computador com “aparência bonita” é necessário que tenha componentes com qualidade e que sejam eficientes.

Brooks veio a desempenhar um papel no desenvolvimento do IBM System/360 que foi uma linha de computadores cuja "arquitetura" ganhou uma nova expressão, considerando sobretudo "o que o usuário precisa saber".

IBM

É a sigla de International Business Machines, que significa Máquinas de Negócio Internacionais, e é uma empresa americana que trabalha com produtos voltados para a área de informática, como computadores, hardwares e softwares. A IBM é uma das maiores empresas da área de TI (Tecnologia da Informação) no mundo, e foi fundada em 1888.

Stretch

O IBM 7030, também conhecido por Stretch, foi o primeiro supercomputador lançado na segunda geração, desenvolvido pela IBM. Seu tamanho era bem reduzido comparado com máquinas como o ENIAC, podendo ocupar somente uma sala comum. Ele era utilizado por grandes companhias, custando em torno de 13 milhões de dólares na época.

Outra maneira interessante para o entendimento do que seja arquitetura e organização é fazer uma separação dos conceitos. Essa divisão significa entender o que vem a ser o conceito de arquitetura e em seguida o que é organização.

Ao falarmos de **arquitetura** de um computador torna-se importante entender que estamos fazendo referência à aplicação teórica do que foi desenhado pelo fabricante do computador.

Seria algo semelhante a um engenheiro que faz um desenho de um avião e especifica na teoria como será seu funcionamento. Similarmente, os fabricantes de computadores fazem um projeto, planejam como será o “comportamento” da máquina e especificam como determinada instrução (linha de código) será executada. Um usuário mais avançado, como por exemplo, um programador de software, estará atento ao desempenho do computador, pois isso irá impactar diretamente na execução do programa que ele está desenvolvendo.



09

Vamos agora falar da **organização**. Podemos dizer que é a estrutura e organização do *hardware*, ou seja, faz referência a como é o funcionamento interno da máquina (computador), as **unidades operacionais** e suas conexões que implementam as especificações de sua arquitetura, ou seja, como serão as características da arquitetura implementada.

Ao falarmos em **organização** estamos fazendo referência à decisão e definição de como será implementada uma instrução (linha de código) por uma unidade específica de multiplicação.

Podemos usar como exemplo a chamada “família” de processadores Intel x86. Esses processadores compartilham a mesma arquitetura básica ou o mesmo “comportamento” básico. Entretanto, a organização é diferente entre uma versão e outra. Sendo assim, um processador **intel x86 core 2 duo** tem a mesma arquitetura básica que o *Intel core i7*, mas em matéria de execução e performance são diferentes.

A tecnologia está sempre em evolução, mas essa evolução é sempre um aperfeiçoamento do que já estava sendo utilizado.

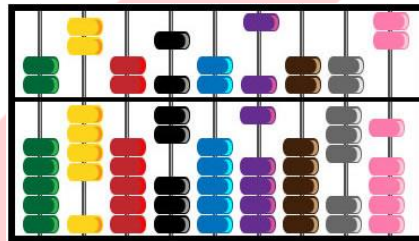
10

2 - HISTÓRIA DA EVOLUÇÃO DOS COMPUTADORES

Um pouco de história da evolução dos computadores, ao olharmos para o passado, pode nos fazer entender o presente e o que ainda estará por vir.

Caro estudante, a compreensão do processo evolutivo dos computadores e por consequência da tecnologia no decorrer dos anos, mostra o quanto as máquinas contribuíram com as mudanças na história das civilizações.

O ábaco, considerado a primeira calculadora conhecida, é um instrumento que foi e é ainda utilizado para cálculo. Alguns historiadores falam que o ábaco foi inventado, há mais de 5.000 anos, na Mesopotâmia, e depois os romanos e chineses foram os responsáveis pelo seu aperfeiçoamento. Com este instrumento é possível efetuar operações matemáticas básicas e principalmente os chineses o utilizam até hoje.



11

O francês Blaise Pascal, que viveu no século XVII, foi o responsável pela construção da primeira **calculadora mecânica** capaz de efetuar somas e subtrações. Entretanto, há historiadores que contestam quem foi o responsável pela construção da primeira máquina de calcular mecânica. Durante vários anos nada era sabido sobre uma máquina criada por Wilhelm Schickard, que viveu entre o século XVI e XVII, a referida máquina subtraía, multiplicava e dividia. Ela foi perdida durante a Guerra dos Trinta Anos.



12

No século XIX, aproximadamente no ano de 1801, o inventor Joseph Marie Jacquard criou o que veio a ser conhecido **como o tear programável** mecânico que era dotado de uma leitora de cartões perfurados, os quais representavam certos modelos usados nos desenhos do tecido. Em virtude do sucesso desse evento e por funcionar de maneira eficiente o tear veio causar desemprego em virtude da automação.

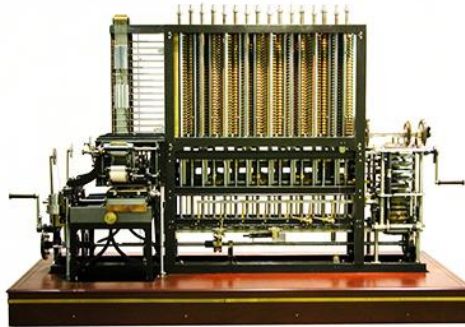


Tear programável de Joseph Marie Jacquard

13

Charles Babbage, que viveu no século XIX, ano de 1822, criou o que veio a ser chamado de **computador analítico**. Essa máquina dispunha de um dispositivo de entrada (semelhante ao tear de Jacquard) para a leitura de cartões perfurados com determinados dados e instruções. Estes dados seriam números e as instruções seria o que deveria ser feito com estes dados. O computador analítico dispunha de um

dispositivo de memória para guardar os dados (números). Esta “memória” ganhou o nome de **armazém**. Este “armazém” utilizava um banco com 1000 registradores com capacidade de armazenar um número de 50 dígitos.



O computador analítico

Também havia um dispositivo para imprimir os resultados obtidos. Babbage teve a notável ajuda da filha de Lord Byron, a matemática Ada Augusta Byron. Ada, que também era condessa, foi a responsável pelo desenvolvimento de uma série de instruções e por isso é considerada a primeira programadora do mundo. Charles Babbage é chamado de o “precursor do computador” e a condessa Ada é considerada a precursora do desenvolvimento de *software*.

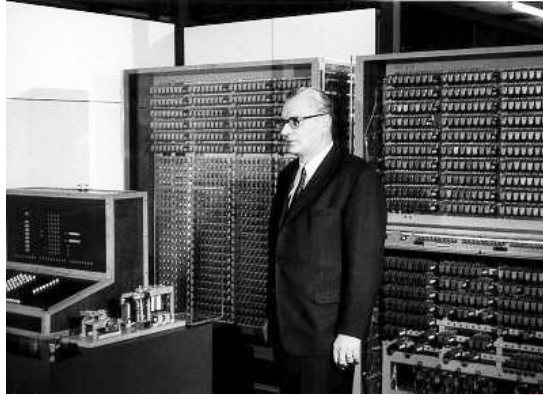
14

Herman Hollerith (1860-1929) também se inspirou nos cartões de Jacquard para criar uma máquina para acumular e classificar informações – a **tabuladora de censo**.



Tabuladora de censo

Em 1941, Konrad Zuse (Alemanha) criou o primeiro **computador digital**, automático, programável, de propósito geral, completamente funcional (eletromecânico).



15

Com a participação de Alan Turing, matemático e cientista da computação britânico, foi desenvolvido na Inglaterra no período de 1939 a 1943 (2ª guerra mundial) o Colossus criado com a intenção de quebrar o código da máquina de criptografia alemã conhecida com Enigma.



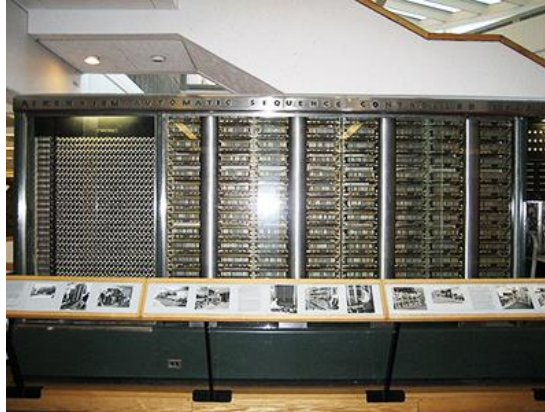
Colossus



Enigma

16

Em 1944 na Universidade de Harvard – EUA, o engenheiro Howard Aiken foi um pioneiro da computação, sendo o principal responsável pelo desenvolvimento do computador conhecido como Harvard Mark I, patrocinado pela IBM.



Harvard Mark I

17

Acompanhe, na linha do tempo, outros acontecimentos importantes sobre a evolução dos computadores:



No dia 5 de outubro de 1991 Linus Torvalds anunciou a primeira versão "oficial" do núcleo (kernel) do que veio a ser chamado de Linux, versão 0.02. Depois o kernel Linux foi utilizado no projeto capitaneado por Richard Stallman e passou a chamar GNU/Linux, junção do núcleo do sistema operacional (Kernel Linux) com a interface gráfica do projeto GNU ([GNOME](#)).

GNOME (acrônimo para GNU Network Object Model Environment) é um projeto de software livre abrangendo o Ambiente de Trabalho GNOME, para os usuários, e a Plataforma de Desenvolvimento GNOME, para os desenvolvedores. O projeto dá ênfase especial a usabilidade, acessibilidade e internacionalização.

1946

Em 1946 os cientistas norte-americanos Jonh Eckert e John Mauchly, ambos da Electronic Control Company, criaram o ENIAC (Electrical Numerical Integrator and Calculator). O ENIAC é considerado o primeiro computador digital eletrônico de grande escala. Uma das destinações deste computador foi para cálculos de trajetórias durante a Segunda Guerra Mundial.

1949

Em 1949 John Von Neumann participa da criação do computador EDSAC (Electronic Delay Storage Automatic Calculator), primeiro a utilizar fita magnética. Foi também, um dos primeiros computadores fabricados na Inglaterra.

1956

A IBM inova no mercado em 1956 com a criação do IBM 305 RAMAC (Random Access Method of Accounting and Control ou método de acesso aleatório para contabilidade e controle), sendo o primeiro a utilizar disco rígido magnético.

1964

Em 1964, a IBM (International Business Machines empresa americana) lançou o IBM System/360 (S/360) que se constitui numa família de mainframes, que foi a primeira família de computadores a fazer uma distinção clara entre a arquitetura e a implementação, permitindo que a IBM lançasse um conjunto de projetos compatíveis em várias faixas de preço.

1972

Em 1972 é lançado pela Intel o microprocessador 8008, considerado o primeiro microprocessador de 8 bits.

1976

No ano de 1976 é lançado o Apple I, computador pessoal simples, desenhado por Steve Wozniak para uso próprio. Steve Jobs teve a ideia de comercializar este computador.

1977

Em 1977 é lançado o Apple II. Este equipamento tinha processador MOS Technology 6502, clock de 1 MHz, 4kb de memória RAM.

1984

Em 1984 surge o Macintosh ou MAC, que é considerado o primeiro computador pessoal a popularizar a interface gráfica GUI (Graphical User Interface).

1985

Em 1985, Bill Gates, cria o sistema operacional windows 1.0, sendo uma interface do DOS.

18

Alguns estudiosos da área costumam classificar os computadores (máquinas) por geração. Esta classificação especifica determinadas características das máquinas naquele período (anos), tudo relacionado à história da evolução dos computadores. Vejamos:

**1ª geração**

As máquinas que são chamadas de Primeira Geração, período de 1930 a 1958, tinham por base a utilização de relés e válvulas eletrônicas para a realização de cálculos automaticamente, eram os computadores de primeira geração.

2ª geração

Os computadores de Segunda Geração, período de 1955 a 1965, tiveram como mola propulsora a invenção do transistor no ano de 1948, sendo assim, o mundo dos computadores foi tomado por uma onda de novos projetos que deram origem, na década de 1960 a empresas que em nossos dias são reconhecidas mundialmente. Exemplo: IBM.

3ª geração

Computadores da chamada Terceira Geração, período de 1965 a 1980, têm por característica a substituição dos transistores pela tecnologia dos circuitos integrados (transistores / componentes eletrônicos miniaturizados e montados numa única pastilha de silício = chip). No ano de 1961 os circuitos integrados tiveram sua inserção no mercado por meio da empresa americana “Fairchild Semiconductor” e também pela empresa americana “Texas Instruments”, ambas estavam localizadas no famoso e conhecido “Vale do Silício” que está situado na região de Palo Alto e Stanford, no estado da Califórnia, nos Estados Unidos.

4ª geração

Computadores chamados da Quarta Geração, período de 1980 a 1984, caracterizam-se pelo que foi chamado de LSI (Large Scale of Integration), que foi a integração em larga de escala, chegando a combinar até 65 mil componentes em apenas um chip (pastilha de silício).

5ª geração

A chamada Quinta Geração, período de 1984 a 1990, tem como característica o uso de “Integrated Circuit Very Large Scale Integration” (ICVLSI), em uma tradução simples seria “Circuitos Integrados em Escala Maior de Integração”. Os famosos chips começam a ter seu tamanho diminuído e tornando possível a criação de computadores cada vez menores (compactos).

6ª geração

Os computadores da chamada Sexta Geração, a partir de 1990, são os chamados computadores da “Era da computação quântica”. Já existem protótipos, mas ainda não viáveis, economicamente falando. O chamado CQ (Computador Quântico) utiliza as propriedades quânticas da matéria. De uma forma mais simples, podemos dizer que a computação quântica é a aplicação das teorias e propriedades da mecânica quântica ao computador. Em teoria, os computadores quânticos são muito mais rápidos do que os Desktops (PCs – Computador Pessoal) atuais, inclusive há limitadores na física quântica. Um CQ pode ser formado por fótons, nêutrons, prótons, elétrons e pósitrons. Em suma, em virtude de poder ser formado por partículas tão pequenas, é também possível imaginar que os processadores dos CQ, poderão ser bem pequenos.

19

Para distrair um pouco e aproveitar para recordar como ocorreu a história da evolução da computação, assista aos dois vídeos a seguir. Eles tratam da importância dos computadores em nossa sociedade.

Vamos lá?

Evolução da Informática Dos primeiros computadores à internet

<https://www.youtube.com/watch?v=SLsqeUPW-fs>

Historia do computador em minutos

<https://www.youtube.com/watch?v=F3qWg1JBPZg>

20

3 - OS COMPONENTES DO COMPUTADOR

Muito bem! Vamos agora começar a fazer uma descrição básica dos componentes de um computador. Podemos dizer de maneira simples que um computador é constituído por 3 elementos básicos que são: microprocessador (CPU ou UCP), memória e dispositivos de entrada e saída (E/S).



Quando falamos em memória é necessário haver a divisão em dois grupos, as memórias voláteis e as não voláteis ou também chamadas de armazenamento permanente. As memórias voláteis, ou memória RAM (Random Access Memory), necessitam de energia elétrica para manter as informações armazenadas. Quando desligamos o computador, as informações importantes são armazenadas nos dispositivos de memória não voláteis (como o disco rígido ou HD - Hard Drive). Os dispositivos de memória volátil são mais caros e de menor capacidade, porém são muito mais rápidos, tornando possível ao computador realizar o processamento de forma mais eficiente.

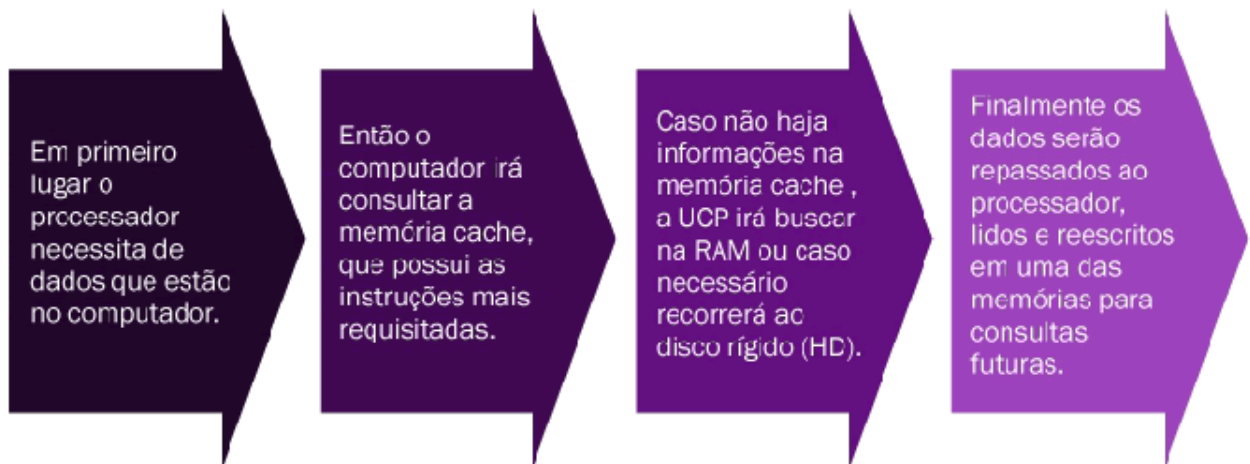
21

O computador é uma máquina poderosa de armazenamento de informações. Todo programa, página a qual se está navegando na internet ou algum arquivo que foi aberto, ficará registrado de alguma maneira, em um local do computador. Assim como em nós, usuários, a memória é uma das funções do sistema mais desenvolvida.

O que você provavelmente ainda não sabe é que a RAM é apenas um componente de uma série de outros que também são responsáveis por armazenar informações digitais sem perdê-las. Alguns desses componentes você irá observar que são muito utilizados no cotidiano da informática, até por usuários menos experientes. Alguns nomes são: memória cache, ROM, HD, BIOS. Cada uma possui um papel específico e um momento certo para ser utilizada.

Para entender um pouco melhor como é o funcionamento do seu PC, valerá a pena conhecer como é feito o caminho percorrido pelos dados e em quais memórias (volátil ou não) eles poderão ser armazenados no intuito de facilitar o trabalho do processador.

Você imagina como é intensa a troca e fluxo de informações no processador e memória? Pois é! O fluxo é realmente grande e muito intenso, mesmo quando você não está utilizando com tanta intensidade. Muitos dados “passam” pelo processador, que necessita rapidamente ficar “livre” deles para poder receber outros dados e não ficar sobrecarregado. A CPU dispõe de diferentes memórias, sempre trabalhando para trafegar os dados da maneira mais rápida possível, em um processo que ocorre em ciclos e que pode ser resumido da seguinte maneira:



22

Fazendo uma explicação básica sobre a UCP ou CPU, chamada de Unidade Central de Processamento ou simplesmente **processador**, podemos dizer que:

A CPU é a responsável em tornar possível a utilização (execução) dos programas que atenderão as tarefas que o usuário necessita.

Por meio de vários cálculos e consultas a dados, caberá à CPU tornar possível a utilização, por exemplo, do navegador Internet Explorer, Firefox, Chrome ou etc. Contudo, caso você deseje abrir novamente o mesmo programa? O que você notará? Será que é mais rápido do que foi na primeira vez? Sim, será mais rápido, sabe por quê? Isto ocorre em virtude de toda a informação está distribuída em “memórias” espalhadas pelo hardware, sendo que cada uma possui características e especificidades diferentes.

Uma dessas “memórias” é o CACHE. Ele é um tipo de memória importante no processador, cabe ao CACHE, em virtude de estar fisicamente próxima ao processador, ser mais rápido, guardar as informações temporárias que precisam ser processadas de maneira mais eficiente. Apesar da eficiência, o cache tem pequena capacidade de armazenamento, não excedendo KBs ou MBs. Atualmente a memória cache de um modelo básico de notebook é de aproximadamente 3Mb. Em virtude disso, sua principal função é guardar os resultados das operações do processador, que são fornecidos à medida do seu funcionamento.



A necessidade de sua criação surgiu justamente pela necessidade de guardar resultados sem gastar espaço de outras memórias e obter uma resposta imediatamente após requisitá-la, fazendo com que sua latência seja quase nula. Quando o processador deseja acessar uma informação, é no cache o primeiro local de busca. Caso o cache tiver entre seus arquivos (quando ocorre o chamado cache hit), a varredura termina e o dado é executado imediatamente.

23

O cache é dividido em níveis, de acordo com sua capacidade e proximidade com a CPU. Os dados do nível 1 (cache L1) são menores, porém de fácil acesso e até mesmo atingindo o dobro da velocidade da memória principal (MP). No nível 2 (cache L2) não há tanta rapidez, mas a transmissão ainda não é comprometida. As demais camadas seguem o mesmo princípio.

Todas elas estão em uso constante, mas nem sempre se encarregam de todo o armazenamento. Além de tudo isso, não é possível aumentar o tamanho da memória cache sem a substituição do processador.

A memória RAM, também chamada de memória de acesso aleatório, torna possível que possamos acessar os dados de forma não sequencial, além de ser relativamente rápida, cabe a ela guardar algumas informações do sistema operacional, de softwares em uso e os processos em atividade, uma quantidade de dados muito maior do que o cache poderia armazenar.

Quando você está jogando, por exemplo, cabe a RAM ser a responsável por acessar as texturas, animações e outros dados contidos para que o game rode (funcione) sem problemas.

A memória RAM e a memória CACHE necessitam de energia (elétrica) para manterem o armazenamento, sendo assim, é possível dizer que seu “ponto negativo” é ser volátil, pois quando o computador é desligado as informações armazenadas são perdidas.

Mesmo seu computador tendo uma capacidade razoável de memória RAM, isto não será suficiente se você utilizar seu computador para determinada atividade que exija muito processamento. Dependendo da situação, para a máquina não travar, ela buscará outras alternativas. Os dados que devem permanecer no sistema de qualquer maneira precisam de uma memória que não seja volátil, mas que consiga suportar um alto fluxo de informações. É aí que entra outra parte vital e importante do computador, o popular HD (disco rígido).

24

Armazenar memórias temporárias acaba sendo um trabalho para o HD em último caso, mesmo não sendo essa a sua principal função. Em formato de memória virtual (junção da MP e espaço em disco), ele

executa e registra os dados de forma mecânica, diferente do método eletrônico das demais (cache e RAM), tornando esse processo mais lento.

O HD é um “tipo de memória” em que estão guardados os arquivos que não são perdidos a cada boot (inicialização), como músicas e fotos. Eles são acessados com frequência pelo usuário, mas não pedem uma velocidade de leitura tão absurda como os processos do sistema.



Segundo o Terry's Computer Tips (site sobre TI – Tecnologia da Informação – página na língua Inglesa), para efeito de comparação, acessar um dado a partir do disco rígido leva cerca de 0,013 segundos, enquanto o mesmo dado a partir da RAM chega em 0,000.000.01 segundos. Para compensar, o HD possui a maior capacidade entre as memórias citadas anteriormente.

Então fique alerta ao perceber que seu computador está utilizando o HD (disco rígido) para armazenar memória (swapping ou paginação). Isto é um indicativo que seu PC já está saturado de arquivos, fazendo com que a máquina apresente lentidão e travamentos com certa regularidade.

25

4 - DISPOSITIVOS DE ENTRADA E DE SAÍDA

Agora iremos falar dos chamados **dispositivos de entrada e saída** que são os responsáveis por tornar possível a interação com o usuário ou com outras máquinas, por exemplo: teclado, mouse, monitor, modem, impressora, webcam, dispositivos USB etc.

Os dispositivos de entrada e saída (E/S) ou *input/output (I/O)* são também chamados de periféricos. Eles tornam possível uma melhor interação do processador com o homem, possibilitando a entrada e saída ou apenas a saída de dados.



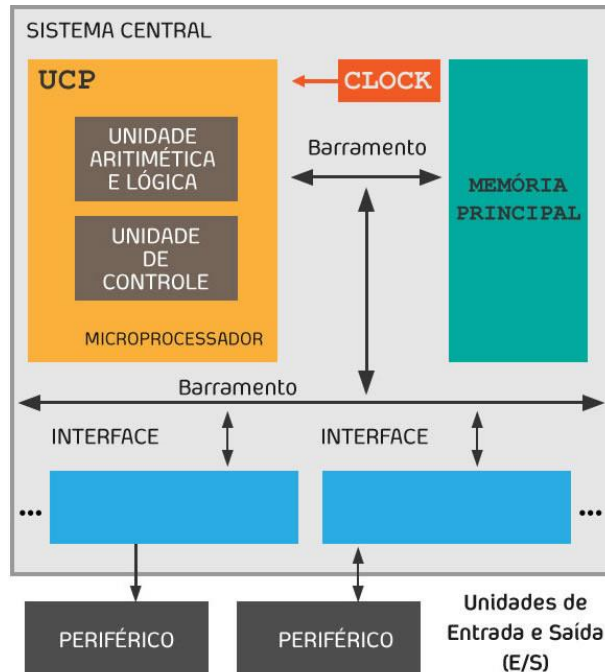
Há dispositivos que funcionam tanto para entrada como para saída de dados, por exemplo, o modem e o pendrive. Outro exemplo de dispositivo que funciona tanto para entrada como para saída de dados (dispositivo híbrido) é placa de rede do computador.

A comunicação dos dispositivos de E/S é estabelecido por meio de uma via de comunicação, conhecida como: **barramento**.

26

Você já deve estar acostumado ao uso destes dispositivos, mas com o intuito de padronização do conceito iremos estudar que os principais dispositivos de entrada e saída de informações. Os dispositivos mais conhecidos de entrada são: teclado, mouse, drive de CD / DVD-ROM, pen drive, scanner, microfone, joystick, câmera filmadora, câmera digital, tela sensível ao toque, mesa gráfica, caneta ótica e outros. Os principais dispositivos de saída de informações são: monitor de vídeo, drive de CD-ROM, caixa de som, impressora, sensores movimento, temperatura etc. e óculos (para realidade virtual).

No que tange os principais dispositivos que possuem a característica de ser tanto de entrada como de saída de informações, podemos exemplificar como: modem, drive de disquete, gravador de CD / DVD, placa de rede e disco rígido.



Organização Funcional de um Computador

27

Dando continuidade ao nosso estudo, vamos fazer conhecer alguns componentes de um computador que geralmente não são muito comentados. Estes componentes são: gabinete, fonte, placa-mãe, BIOS, clock, barramento, placa de extensão, chipset, MOS.

O chamado gabinete e o receptáculo, que contêm na sua parte interna a fonte, placa mãe, dispositivos de armazenamento, placas de expansão, memória etc. Existem vários modelos de desenho e tamanhos diferentes. Há a exigência de terem o tamanho adequado para serem compatíveis com determinados tipos de fonte e com as placas mãe.

Atualmente a maioria das “fontes” são bivolt (110 ou 220 volts) e os modelos mais conhecidos são AT e ATX.



AT: Possui uma chave liga/desliga e a saída é um conjunto de dois conectores semelhantes.



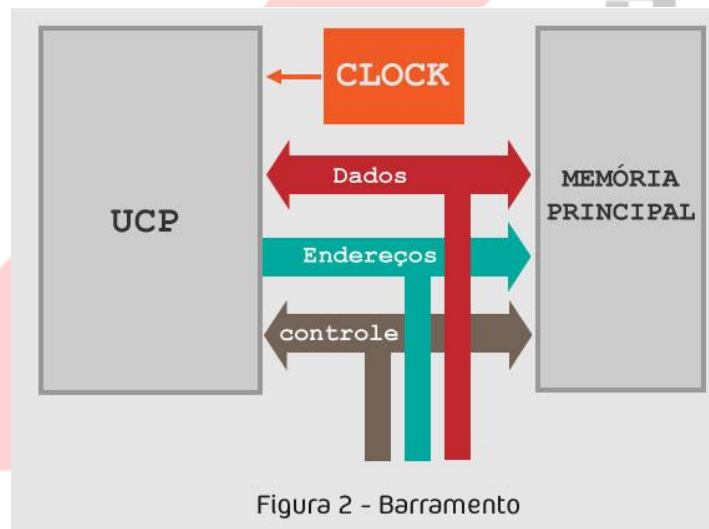
ATX: Também chamada de fonte inteligente, ela não possui uma chave liga/desliga (seu desligamento é realizado por um pulso enviado por meio da placa-mãe). Possui um único conector de saída.

A placa-mãe (*motherboard*) tem este nome por ser um componente físico (hardware) muito importante em um computador.

A placa-mãe é quem coordena toda a comunicação entre os seus componentes (filhos). É na placa-mãe que o processador, memória, placa de rede, chipset, slots e etc. estão fisicamente conectados.

Também é na placa-mãe que fica o barramento onde trafega os dados entre a CPU e os periféricos. Ao adquirirmos uma placa-mãe precisamos estar atentos a qual arquitetura ela foi desenvolvida para também sabermos quais e a compatibilidade dos componentes (Chipset, BIOS, Barramentos, Slots) adquiridos.

Na placa-mãe há um cristal piezoelétrico (ou um circuito integrado) que é o responsável por gerar os sinais de sincronismo e determinar a velocidade de processamento. O cristal, conhecido como CLOCK, fornece um pulso de alta precisão cuja frequência irá depender do uso do processador. Quanto mais rápido o clock, mais instruções uma CPU pode executar por segundo. A velocidade de clock é expressada em megahertz (MHz), 1MHz é igual a 1 milhão de ciclos por segundo.



O barramento também está localizado na placa-mãe.

Cabe ao barramento ser o “caminho” por onde trafega toda comunicação entre os componentes, tornando possível a conexão e transporte dos dados entre os componentes que necessitam desta comunicação.

Exemplo de componentes da placa-mãe: CPU, memória, placas de expansão, dispositivos de E/S e etc. O processador também estabelece comunicação com a memória principal por meio do barramento.

29

Cabe às **placas de extensão** tornar possível o acréscimo de novos recursos ao computador. Elas são conectadas à placa-mãe por meio *doss/ots* (termo em inglês para designar ranhura, fenda, conector, encaixe ou espaço.). Exemplos: placas de som, placa de vídeo, placa de rede etc.

O chip denominado CMOS é composto por um relógio eletrônico e memória 64 bytes de memória RAM, e nesta memória que estão armazenadas as informações relativas à configuração do hardware do computador.

O BIOS (Basic Input/Output System) é um pequeno programa armazenado em um chip de memória da placa de UCP. Ele é responsável por “despertar” o computador, sendo assim, ao ligar o computador o BIOS entra em ação, faz uma contagem e verifica a memória RAM, inicializa dispositivos, e principalmente, dá início ao processo de boot.

Boot é a operação de passagem do sistema operacional do disco rígido (HD) onde fica armazenado para a memória do computador.

Cabe ao chamado “chipset” ser o responsável pela gerência do funcionamento da placa-mãe. Ele faz o controle de memória cache, da memória RAM, do buffer de dados, da interface com a CPU etc.).

É responsável, também, pelas informações necessárias ao reconhecimento de hardware (armazenadas na sua memória ROM).

30

Vamos aproveitar e assistir a este vídeo para reforçar o entendimento acerca dos dispositivos de entrada e saída (I/O).

A interface homem-computador

https://www.youtube.com/watch?v=UZ14m_yVmO4

31

RESUMO

Durante o estudo deste módulo tivemos a oportunidade de entender que a arquitetura de computadores trata do funcionamento e também do comportamento de um sistema computacional (computador). Fica evidente a importância deste estudo para sua formação técnica. Vimos que atualmente as empresas, ao desenharem o projeto da Arquitetura de seus computadores, preocupam-se em entender as necessidades do seu público alvo.

É interessante destacar que, ao estudarmos a história da evolução dos computadores, estaremos entendendo a importância dos primeiros estudiosos do assunto. Graças às pesquisas destes obstinados pesquisadores é que podemos usufruir de tamanha evolução e também podemos traçar uma perspectiva do que o futuro nos reserva.

Vimos também que um sistema computacional é agrupado em 3 subsistemas básicos, conhecidos como **unidades operacionais**, que são a CPU, memória e periféricos. Vimos que um sistema computacional é um conjunto de circuitos eletrônicos conectados, constituído por processadores, registradores, memórias, dispositivos de entrada e saída. Cada componente tem sua importância e função na arquitetura de computadores e cabe a esses dispositivos a manipulação dos dados digitalmente, proporcionando de maneira confiável a transmissão e representação dos dados.

UNIDADE 1 – FUNDAMENTOS DA ARQUITETURA E REPRESENTAÇÃO DE DADOS NO SISTEMA OPERACIONAL

MÓDULO 2 – NÍVEIS E MÁQUINAS VIRTUAIS

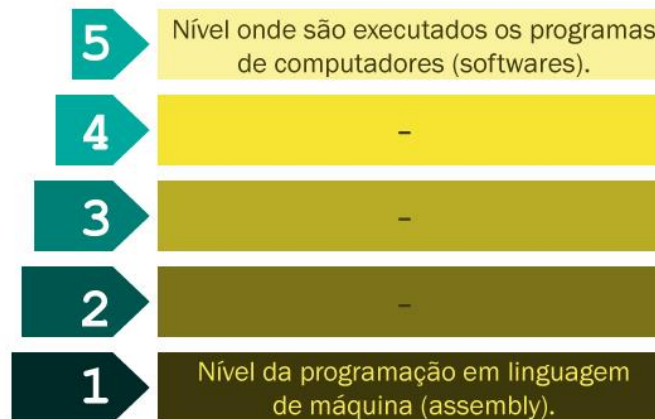
01

1 - REPRESENTAÇÃO DAS INFORMAÇÕES: O BIT, O BYTE E A PALAVRA

Durante o estudo deste módulo, teremos a oportunidade de entender como é feita a representação das informações, o que é bit, o byte e a palavra, aprenderemos a identificar e a entender o que são os níveis de máquina e falaremos também sobre máquinas virtuais e sua classificação.

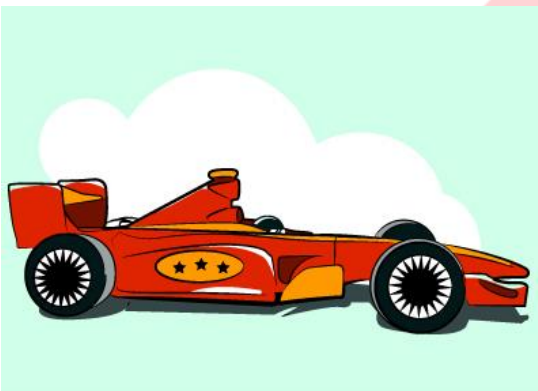
Cabe uma ressalva importante ao iniciarmos este módulo. Você deve estar lembrado de que no início do nosso estudo falamos sobre o entendimento do conceito de **níveis** na arquitetura de computadores (programação de baixo e alto nível). A ideia que está por trás deste conceito é que existem muitos níveis nos quais o computador pode ser “visualizado”.

Vamos lembrar o seguinte exemplo: imagine uma sequência numérica em ordem crescente, começando de 0 até 5. Utilizando este exemplo para fazer uma analogia com os NÍVEIS na Arquitetura de Computadores, podemos dizer que o nível “mais alto” seria o número “5”, local onde são executados os programas (*softwares*). Continuando com o exemplo, a sequência numérica, podemos dizer que “0” seria o nível “mais baixo”, é nível da programação em linguagem de máquina (*assembly*), ou o funcionamento e programação dos registradores (componente do processador).



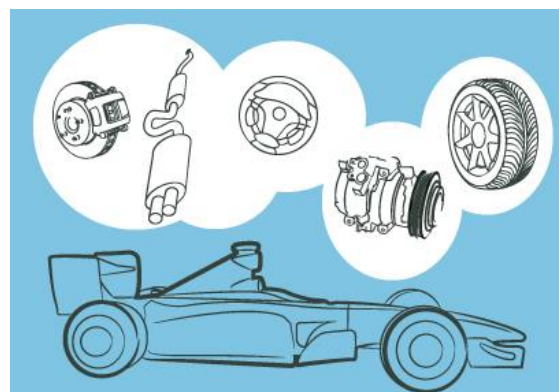
02

No intuito de reforçar o conceito de “níveis na arquitetura de computadores” vamos utilizar como exemplo um carro de Fórmula 1. Vamos supor que você tenha a oportunidade de pilotar um carro de Fórmula 1. Ao entrar e sentar no “cockpit” você observa o volante e os vários instrumentos disponíveis no painel. Logo depois você sai para pista de corrida e sente a velocidade do carro.



Neste nível em que você utiliza o carro e observa o painel e instrumentos, seria comparado com o nível mais alto na arquitetura de computadores, pois durante o tempo em que você “curtiu a velocidade” e todos os recursos disponíveis, provavelmente não ficou preocupado com o funcionamento do motor e dos outros componentes, que é uma preocupação dos mecânicos e engenheiros. O nível mais alto seria, então, o nosso nível, em que estamos preocupados apenas em utilizar o carro de Fórmula 1 e não nos preocupamos com o motor e outros dispositivos.

O nível do “motor e componentes” é comparado com o nível mais baixo nos “níveis de arquitetura de computadores”. Em suma, a linguagem de baixo nível (*assembly*) ou linguagem de máquina é aquela utilizada pelos engenheiros de hardware para criar instruções que permitam o melhor aproveitamento dos recursos de hardware (processador, memória e etc.). Os engenheiros de hardware podem ser comparados aos engenheiros e mecânicos de um carro de Fórmula 1.



Quando falamos em programação de alto nível (aplicativos) estamos falando dos softwares desenvolvidos por programadores. Esse nível está relacionado aos aplicativos, softwares e utilitários que não se preocupam com os recursos de hardware existentes em um computador.

03

Vale ressaltar que este conceito de nível não está relacionado a uma linguagem ser melhor ou superior pelo fato de ser de alto nível e o outro ser ruim por ser de baixo nível. Observe a figura abaixo que no nível “mais alto” estão os aplicativos e utilitários e no nível “mais baixo” está a linguagem de máquina, microprogramação e os dispositivos físicos.



O entendimento do conceito de “níveis” na arquitetura de computadores ajudará na compreensão que no “nível mais baixo” os componentes trocam informações utilizando bits e bytes. Quando digitamos “www.google.com” no nosso navegador (browser), que está no nível mais alto, transforma esta informação em “bits e bytes”, passa por vários níveis até chegar ao nível mais baixo e ser transformado em sinais elétricos e enviado via placa de rede e assim por diante.

Toda informação que será introduzida em um computador (sejam dados que serão processados ou instruções de um programa) necessita ser entendida pela máquina, para que possa interpretá-la e processá-la. Quando você lê esta aula, as informações apresentadas em forma de caracteres são entendidas pelo leitor porque ele conhece o formato e o significado dos símbolos que representam os caracteres alfabéticos, os numéricos (algarismos) e os sinais de pontuação ou matemáticos (+, -, X, /, >, <, = etc.).

Para que seu computador pudesse representar eletricamente todos os símbolos utilizados na linguagem humana, seriam necessários mais de 100 diferentes valores de voltagem (ou corrente). Tal máquina seria inviável economicamente e certamente seria difícil de ser construída para fins comerciais.

Saiba +

O computador ou um sistema computacional, sendo formado por circuitos eletrônicos interligados, troca e armazena informações (dados) internamente sob a forma eletrônica (sinais elétricos). Esta troca de informações é feita por alteração no valor de voltagem ou de corrente que está passando no barramento (linhas de comunicação que tornam possível a interligação entre os dispositivos de um sistema de computação). Cabe fazer uma ressalva quanto ao HD (disco rígido), pois ele armazena informações sobre a forma magnética ou CD/DVD/BlueRay que armazena informações sob forma ótica.

04

Considerando a necessidade de ser mais simples e confiável, optou-se em projetar um circuito capaz de gerar e manipular o menor número possível de valores distintos. Então surgiu o que é chamado de “**máquina binária**”, isto é, capaz de entender apenas dois valores diferentes: 0 e 1.

Com base nesta base binária fica muito mais simples o emprego da lógica booleana (SIM/NÃO, ABERTO/FECHADO, ACIMA/ABAIXO, LIGADO/DESLIGADO). Dessa maneira, os computadores digitais são totalmente binários. Toda informação introduzida em um computador é convertida para a forma binária, por meio do emprego de um código qualquer de armazenamento, como será visto mais adiante.

**05**

A menor unidade de informação armazenável em um computador ou sistema computacional é o algarismo binário ou dígito binário, conhecido como bit (inglês= **binary digit**).

O bit pode ter, então, somente dois valores: 0 ou 1. Isto quer dizer, que ele é representado numericamente somente pelo numeral 1 ou numeral 0.

Consequentemente, em virtude de ter possibilidades tão limitadas, o bit pouco pode representar isoladamente; em virtude disso, as informações manipuladas por um computador são codificadas em grupos ordenados de bits, de modo a terem um significado útil. O menor grupo ordenado de bits representando uma informação útil e compreensível para nós, seres humanos, é o byte (leia-se "baite").

Coube a IBM a primeira definição para "byte". Segundo a IBM, "byte" é um grupo ordenado de bits, que tem por função facilitar e tornar mais eficiente a manipulação interna (computador). Um byte é definido como um grupo ordenado de 8 **bits**, tratados de forma individual, como unidade de armazenamento e transferência.

Em virtude dos principais códigos de representação de caracteres utilizarem grupos de oito bits por "caractere", os conceitos de **byte** e **caractere** passam a ser semelhantes e as palavras, praticamente sinônimas.

Costumeiramente, no mercado atual, as memórias são construídas com a capacidade de acessar, armazenar e recuperar informações byte a byte.

Como vimos anteriormente, os computadores são binários e, por consequência, todas as indicações numéricas referem-se a potências de 2.

Para facilitar a representação, adotaram-se letras para corresponder a determinados valores, por exemplo, a letra "K" é a representação de 1.024 unidades (décima potência de 2 ou $2^{10}=1024$) e a letra "M" (Mega) representa o valor de 1.048.576 unidades (valor igual a 1.024×1.024 ou $2^{10} \times 2^{10}=2^{20}$).

06

Quando falamos que determinada máquina tem capacidade de 64 Kbytes (pronuncia-se "sessenta e quatro k baites") estamos fazendo referência a um valor de $64 \times 1.024 = 65.536$ bytes, enquanto 2M bytes (pronuncia-se "dois mega baites") corresponde a $2 \times 1.024 \times 1.024 = 1.048.576$ bytes.

Com a evolução constante e o aumento vertiginoso da capacidade dos dispositivos de armazenamento dos computadores, criou-se um terceiro elemento para abreviar valores mais elevados: trata-se do giga, que é representado pela letra "G". Ele representa um valor igual a 1.024 mega ou 1.048.576 K ou $2^{30}=2^{10} \times 2^{10} \times 2^{10}=1.073.741.824$ unidades. Atualmente, já é possível ouvir o termo "terá", para a representação de um valor igual a 2^{40} ou 1.024 G. Há também o termo "peta" que é representado pela expressão 2^{50} ou 1.024 teras. Enfim, a evolução não para e provavelmente em um futuro próximo, estaremos utilizando outras abreviaturas.

Dessa maneira, os valores que são utilizados em computação para indicar capacidade de memória são normalmente compostos de um número (sempre entre 0 e 999) e umas das abreviaturas citadas. Além do bit e do byte, há outro conceito, também importante, a **palavra**. Ela (palavra) está relacionada com o armazenamento e a transferência de informações entre MP (Memória Principal) e UCP e especialmente relacionado ao processamento de dados pela CPU.

Primeiramente, poderemos dizer que a “**palavra**” é a forma de chamar um conjunto de bits capaz de representar uma informação. Consequentemente, uma “palavra” está associada ao tipo de interação entre a memória principal (MP) e o processador (CPU).

Sendo assim, a CPU processa instrução por instrução (cada uma estaria associada a uma palavra), armazena ou recupera número a número (cada um estaria associado a uma palavra), e assim por diante.

Entretanto, alguns fabricantes diferem quanto ao conceito de palavra. Para as empresas IBM e DIGITAL o tamanho dos registradores internos da CPU igual ao da palavra, mas para a empresa Control Data, a palavra tem características mais abrangentes.

Palavra (inglês: word)

É a unidade natural de informação usada pelos computadores. É uma sequência de bits com tamanho fixo e que é processado em conjunto em uma determinada máquina. O número de bits em uma palavra (tamanho / comprimento da palavra) é uma característica preponderante da arquitetura de um computador. Esta característica ecoa em vários aspectos de sua estrutura, da sua operação e na indicação da unidade de transferência entre a UCP e MP (memória principal).

07

No que se refere à unidade de armazenamento, considera-se mais importante a quantidade de bits que são recuperados em um acesso, em geral de tamanho igual a um byte. Esse valor de bits é pequeno demais para representar um número ou uma instrução de máquina e, então, não pode ser aceitável para o tamanho de uma palavra.

Geralmente, são utilizados dois valores diferentes: um relacionado à unidade de armazenamento, o “byte”, e outro para indicar a unidade de transferência e processamento, a “palavra”.

A UCP faz o processamento de valores representados por uma quantidade de “bits” igual à da “palavra”, sendo um indicando da capacidade de processamento do sistema. A empresa americana IBM, por exemplo, define a “palavra” para alguns de seus computadores como tendo um valor igual a 32 bits (4 bytes), enquanto a unidade de armazenamento é o byte.



Atualmente, a maioria dos computadores utiliza a “palavra” com 64 bits, inclusive os supercomputadores que já se utilizam deste tamanho de palavra há mais tempo.

08

2 - ARQUIVOS E REGISTROS

Agora vamos começar a conceituar o que são **arquivos** e **registros**. Para isso é preciso entender que todo processamento em um computador consiste, como já mencionado, na manipulação de dados segundo um conjunto de instruções que são, geralmente, chamados de programas.

Para tornar possível individualizar grupos diferentes de informações, os sistemas operacionais (programas que controlam o armazenamento e recuperação dessas informações para entrada, saída ou guarda em memória secundária) estruturam esses grupos de dados sob uma forma denominada arquivo.

Um arquivo de informações (ou dados) é um conjunto formado por dados (ou informações) de um mesmo tipo ou para uma mesma aplicação. Por exemplo, podemos ter um arquivo de alunos de uma turma (contendo informações sobre cada aluno individualmente), ou um arquivo contendo as instruções de um programa. Cada arquivo é constituído por itens individuais de informação (cada aluno, no nosso exemplo) chamados registros.

Assim, um arquivo de uma turma de 70 alunos possui um total de 70 registros; um arquivo com informações sobre 7 mil empregados de uma organização possui 7 mil registros.

Para entendermos melhor o conceito de armazenamento e recuperação de informações sob a forma de arquivos, podemos fazer uma analogia com um sistema semelhante, porém manual. Vamos supor que a empresa XYZ, fabricante de calçados, tenha um total de 500 empregados. Esta empresa precisa manipular diariamente um estoque de material de consumo com cerca de 9212 itens e que, por incrível que possa parecer, ainda não possui um sistema que automatize todo este controle, que não utilize um sistema informatizado.



09

Vamos continuar a utilizar o exemplo da empresa XYZ. Tomaremos como exemplo a gerência de recursos humanos. No RH estão seguitas informações sobre os funcionários:

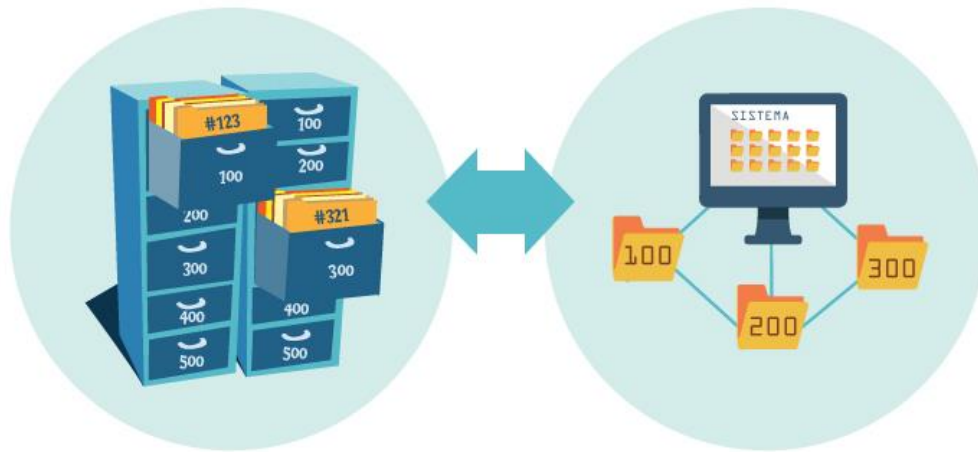


Agora vamos ver como seria a gerência de material. No caso da empresa XYZ o controle dos itens do estoque de material são semelhantes ao do RH, sendo que, há uma ficha para cada item de estoque, contendo as informações pertinentes a cada um e estas fichas são armazenadas em pequenas caixas metálicas, chamadas arquivos portáteis.

10

Muito bem, agora vamos fazer uma analogia entre formulários, pastas, gavetas e etc. No exemplo da empresa XYZ, o armário de aço e a caixa metálica constituem arquivos, com função semelhante aos

arquivos de dados em uma máquina (computador), sendo que, cada pasta ou ficha constitui um registro, respectivamente, do funcionário ou do material de estoque.



A maneira de como é estruturado o armazenamento e recuperação de informações no HD de um computador é projetada conforme o conceito de arquivos e registros. Isso ocorre no HD em virtude do sistema operacional poder guardar informações em grupos para ter maior eficiência na transferência para a MP.

11

3 - REPRESENTAÇÃO DE DADOS: OS CÓDIGOS ASCII, EBCDIC E UNICÓDIGO (UNICODE)

Até agora estávamos estudando como o computador utiliza byts, bytes e palavras. Agora vamos entender que há determinados códigos que são responsáveis em fazer a representação de dados.

Sabemos que um caractere é a unidade básica de armazenamento na maioria dos sistemas computacionais. O armazenamento de caracteres (letras, números e outros símbolos) é possível por meio de codificação onde, por convenção, certos conjuntos de bits representam determinados caracteres.

Há três códigos mais conhecidos para a representação de caracteres, são eles:

- ASCII (American Standard Code for Information Interchange),
- EBCDIC (Extended Binary Coded Decimal Interchange Code) e
- UNICÓDIGO (unicode).

Veremos cada um desses tipos de códigos a seguir.

12

O **ASCII** (American Standard Code for Information Interchange), que em uma tradução para português significa "Código Padrão Americano para o Intercâmbio de Informação", é uma codificação de caracteres de sete bits baseada no alfabeto inglês.

Cada sequência de códigos na tabela ASCII corresponde a um caractere, no geral são representados pelos 8 bits (equivalente a um byte), sendo que o oitavo bit (da direita para a esquerda) serve como um bit de paridade. O bit de paridade é utilizado para detecção de algum tipo de erro.

Para facilitar o entendimento, vamos supor que houve o envio do bit "10110", sendo que o primeiro "1" como o bit de paridade, porém o receptor recebe o seguinte código: "00110", ou seja, o circuito de verificação de paridade percebe que há dois "1", isto é par, então é detectado o erro. Não segredo algum sendo fácil de entender o bit de paridade. Esta técnica consiste em adicionar, pelo transmissor, um bit de redundância ou também chamado de bit de paridade. Este bit é inserido logo após ou antes da sequência de bits enviados. A regra é a seguinte: a) Será o bit "1" quando for ímpar o número de vezes que apareceu o "1", exemplo: 1101101 paridade = 1. b) Na aparição do bit "1" em quantidade par de vezes será adicionado 0, exemplo: 010101010, paridade = 0.

Os códigos ASCII começaram seu desenvolvimento a partir do ano de 1960, e a maioria das codificações de caracteres atuais tem por herança esta base.

A codificação faz a definição de 128 caracteres, preenchendo completamente os sete bits disponíveis nas ($2^7=128$) sequências possíveis. Dessas codificações, o total de trinta e três não são passíveis de serem imprimidas, como caracteres de controle atualmente não utilizáveis para edição de texto, porém amplamente utilizados em dispositivos de comunicação, que afetam o processamento do texto. Veja a tabela abaixo com exemplo:

| Binário | Decimal | Hexa | Glifo |
|-----------|---------|------|-------|
| 0100 0000 | 40 | 64 | @ |

13

A codificação conhecida como **EBCDIC** (Extended Binary Coded Decimal Interchange Code) é uma codificação que utiliza 8 bits de caracteres, que tem descendência direta do código chamado de BCD, que possui 6 bits. Foi criada pela empresa IBM no ano de 1960 e utilizado no computador IBM/360.

Como consequência desta descendência, o código EBCDIC pode até ser modificado para 6 bit e dar origem a um código BCD. Esta iniciativa representa uma normalização, pois na década de 1960, o governo americano havia normalizado o ASCII.

No EBCDIC é feita pela primeira vez a utilização de 8 bits (1 byte) para a codificação de um determinado estado, tornando possível a codificação de 256 diferentes estados. Observe a tabela abaixo e veja como seria a representação em ASCII e EBCDIC das letras A e Z:

| Caracteres | EBCDIC | ASCII |
|------------|-----------|----------|
| A | 1100 0001 | 10100001 |
| Z | 1110 1001 | 10111010 |

14

Na codificação chamada de **UNICÓDIGO** (ou Unicode), são utilizados dois bytes para a representação de mais de 65.000 caracteres ou símbolos. O UNICODE permite fazer o intercâmbio de dados e programas em línguas diferentes.

| Char | Code | Char | Code | Char | Code | Char | Code | Char | Code | Char | Code | Char | Code |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 160 | ı | 161 | ğ | 162 | £ | 163 | ¤ | 164 | ¥ | 165 | | 166 |
| ˆ | 168 | © | 169 | ª | 170 | « | 171 | ¬ | 172 | | 173 | ® | 174 |
| ° | 176 | ± | 177 | ² | 178 | ³ | 179 | ´ | 180 | µ | 181 | ¶ | 182 |
| ¸ | 184 | ¹ | 185 | º | 186 | » | 187 | ¼ | 188 | ½ | 189 | ¾ | 190 |
| À | 192 | - | 193 | Â | 194 | Ã | 195 | Ä | 196 | Å | 197 | Æ | 198 |
| È | 200 | É | 201 | Ê | 202 | Ë | 203 | Ì | 204 | Í | 205 | Î | 206 |
| Ð | 208 | Ñ | 209 | Ò | 210 | Ó | 211 | Ô | 212 | Õ | 213 | Ö | 214 |
| Ø | 216 | Ù | 217 | Ú | 218 | Û | 219 | Ü | 220 | Ý | 221 | Þ | 222 |
| à | 224 | á | 225 | â | 226 | ã | 227 | ä | 228 | å | 229 | æ | 230 |
| è | 232 | é | 233 | ê | 234 | ë | 235 | ì | 236 | í | 237 | î | 238 |
| ð | 240 | ñ | 241 | ò | 242 | ó | 243 | ô | 244 | õ | 245 | ö | 246 |
| ø | 248 | ù | 249 | ú | 250 | û | 251 | ü | 252 | ý | 253 | þ | 254 |
| | | | | | | | | | | | | ÿ | 255 |

15

4 - MÁQUINAS VIRTUAIS

Vamos agora começar o estudo sobre o que são as chamadas “máquinas virtuais”. Em virtude da constante evolução da TI (Tecnologia da Informação) está proporcionando, cada vez mais surgem opções de plataformas diferentes para computadores. Cada uma destas plataformas apresenta características próprias e peculiaridades de funcionamento que impossibilitam migrar arquivos executáveis (programas) de uma plataforma para outra. Um exemplo seria que determinados softwares funcionam apenas em determinadas plataformas, exemplo: plataforma Microsoft Windows, plataforma GNU/Linux ou na plataforma IOS da Apple. Sendo assim, caso código fonte de um programa, escrito em qualquer linguagem de programação, é necessário compilá-lo para determinada plataforma, pois o código executável deve ser compatível com determinada plataforma.

Muito bem, esta necessidade de gerar arquivos executáveis para cada plataforma deu origem à ideia de criar uma ferramenta que tenha a capacidade de separar o desenvolvimento do programa com a sua execução em determinada plataforma ou sistema operacional ou computador. A solução para essa ideia

é a utilização do que é chamado de “máquina virtual”. Agora o que é uma máquina virtual? Para respondermos o que é uma máquina virtual é necessário o entendimento do que seja uma máquina real. Sabemos que você entende que sua máquina ou mesmo o computador que você utiliza são reais, mas no caso de nosso estudo vamos aprender que conceitualmente há uma diferenciação entre máquina real e virtual.

16

Um computador é uma máquina constituída de dois componentes principais dispostos hierarquicamente conforme expresso na figura abaixo.



A partir deste escopo, observando a figura acima, a máquina virtual parte da ideia de acrescentar mais uma camada acima do sistema operacional e da CPU, de tal forma que os programas serão executados independentes da plataforma existente. Sendo assim, ao invés de um código fonte ser compilado em separado para cada plataforma que desejamos trabalhar, agora ele será compilado somente uma vez para a máquina virtual. Caberá a ela tratar a comunicação com as camadas inferiores, fazendo o programa ser executado, possibilitando a portabilidade de programas entre diferentes plataformas.

Sistema Operacional

A camada superior é o Sistema Operacional. Sistema Operacional é o software responsável pela gerência da memória, por coordenar o acesso aos periféricos e também administra todos os processos que serão executados em uma máquina, sendo a interface entre um programa e a CPU. Exemplo de sistemas operacionais: Android, Microsoft Windows, GNU/Linux e IOS da Apple. O sistema operacional, bem como, a UCP, exige que os programas a serem instalados, sigam suas especificações para serem executados.

CPU

A camada inferior é o centro da parte física do computador e corresponde à CPU (Unidade Central de Processamento). Como já estudamos anteriormente, cabe a CPU ser a responsável pela realização de todas as operações que um programa necessite executar. Algo importante de ser observado é a característica de sua arquitetura, isto é, a especificação e implementação de um conjunto de instruções que serão suportadas por tal componente. Sendo assim, para que determinado arquivo seja executável em uma determinada UCP, há necessidade de representar o código fonte que o originou através de instruções codificadas para ter o funcionamento correto. A codificação das instruções também faz parte do projeto da arquitetura da CPU e determina o que cada sequência de bits irá representar.

17

Sendo assim, para a implementação de uma máquina virtual é necessária a criação de uma infraestrutura básica, para a simulação de máquina real, independente da plataforma usada. Mas para que esse uso multiplataforma seja atendido, torna-se necessário delimitar cuidadosamente o funcionamento da máquina virtual, evitando e protegendo o *hardware* de operações perigosas, como por exemplo, acesso a regiões protegidas da memória.

Em primeiro lugar é necessário especificar um conjunto de instruções que determinem como os programas serão executados. Esse conjunto de instruções é determinado e codificado da mesma maneira que se faz para uma UCP. Desse conjunto de instruções surgirá a necessidade de implementar compiladores que gerem códigos com essas instruções a partir dos códigos fonte do programa, porém, estes compiladores extrapolam as fronteiras exigidas para uma máquina virtual e estão vinculados às linguagens de programação que usufruirão desta máquina virtual, o que indica que ela não está vinculada a uma ou outra linguagem de programação.

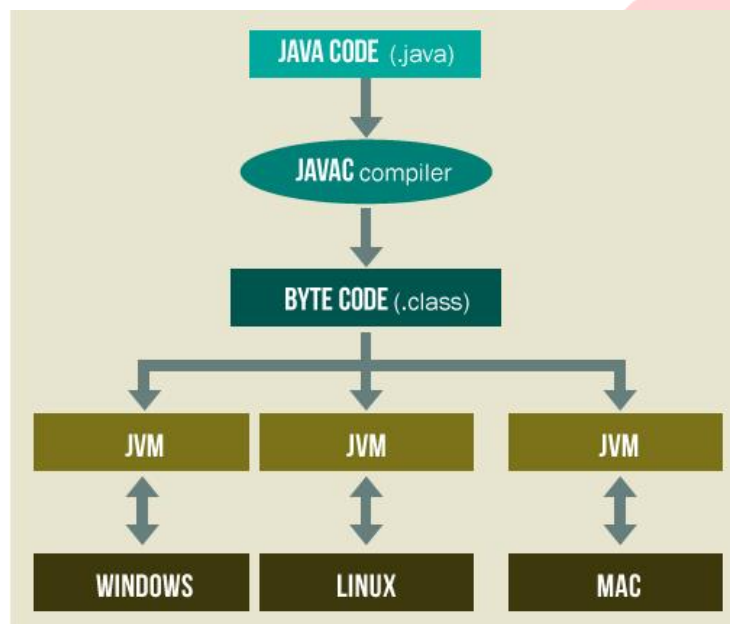
Consequentemente, ao compilarmos determinado programa, que tem por destino uma máquina virtual, será obtido um arquivo com a sequência de instruções suportadas pela máquina virtual. Este arquivo é chamado de **bytecode**. O *bytecode*, por sua vez, será tratado pela máquina virtual como se fosse um arquivo executável, usando áreas de memória do mesmo modo que ocorre em uma execução. Entretanto, ao invés da execução, a máquina virtual transformará as suas instruções em instruções suportadas pela arquitetura da máquina real. Será nesse momento que a máquina virtual tratará as peculiaridades da plataforma em que está trabalhando e é por isso que existe uma máquina virtual para cada plataforma.

18

A transformação dos *bytecodes* em instruções reais é feita de modo interpretado, isto é, da mesma maneira que ocorre com linguagens interpretadas. Os *bytecodes* são lidos em uma sequência e, de forma isolada, associados a uma instrução real. Finalmente, tendo o que pode ser chamado de o

“verdadeiro arquivo executável”, o programa passa para o SO (Sistema Operacional) em que ocorre o processo normal de uma máquina real.

Para facilitar este processo observe a figura abaixo. Nesta figura vemos um programa (software) feito em “.java”, logo depois utilizando um compilador (compiler), que é o programa responsável em traduzir o código fonte de uma linguagem de programação de alto nível (java) para uma linguagem de programação de baixo nível (linguagem que o computador entende). O compilador cria o “.class” que será utilizado pelo JVM (Máquina Virtual Java). O JVM irá criar um ambiente virtual multiplataforma, isto é, poderá funcionar (rodar) em qualquer sistema operacional (Windows, GNU/Linux, MacOS).



19

Atualmente, o conceito de máquina virtual está ganhando novas fronteiras e mercados, com isso, uma alternativa que está sendo muito interessante é sua implementação em hardware, no formato de microchips, por exemplo, o UltraJAVA.

Destaca-se uma nova tendência, o uso de máquina virtual em celulares e em PDAs, tornando possível o reuso de aplicativos de PCs (computadores pessoais – desktops – laptops – notebooks), como ocorre com os jogos feitos em Java, e que estão sendo portados para celulares. Você já deve ter utilizado algum jogo para celular, não é? O jogo “Angry Birds” foi inicialmente feito para os iPhones, mas atualmente foi portado (compilado) para diversas plataformas, podendo ser jogado no Android, por exemplo.



20

5 - VIRTUALIZAÇÃO DE MÁQUINAS

Vamos agora falar do que venha a ser virtualização de máquinas. Há *softwares* de mercado como o VirtualBox da empresa americana Sun Microsystems, o VMWare da empresa também chamada VMware.Inc e outros que possibilitam a criação de uma máquina virtual por meio de um *software* que emula um ambiente computacional, tornando possível a instalação de um sistema operacional ou programa.

A virtualização de máquinas, portanto, é o nome dado ao processo de criação de uma máquina, criada por meio de um software, que executa programas como se fosse um computador real.

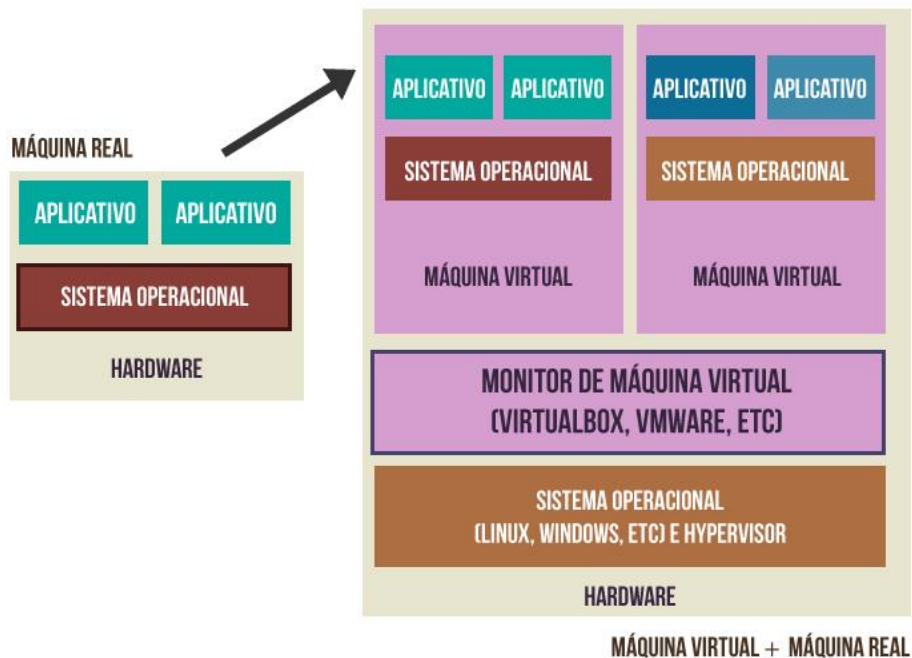
De maneira sucinta e simples, podemos falar que virtualização de máquina funciona como ter um “computador dentro do outro (computador)”. Parece estranho dizer “computador dentro do outro”, mas o software de virtualização (VMWare, VirtualBox) cria um ambiente de sistema computacional que torna possível instalar “um computador dentro do outro”.

O software de virtualização cria todo o ambiente básico de um sistema computacional (memória, processador e dispositivos de entrada e saída) tornando possível a instalação do sistema operacional que você deseja.

21

Para facilitar o entendimento do conceito de máquina virtual, observe a figura abaixo. Note que há uma máquina real (física) que é composta por memória, processador e dispositivos de entrada e saída (sistema computacional) e também pelo sistema operacional. Na máquina real está instalado o software de virtualização (VMWare, VirtualBox). O software de virtualização, tendo por base o sistema computacional da máquina real, criará o ambiente (sistema computacional) que tornará possível a instalação de outro sistema operacional. Veja que na figura há duas “máquinas virtuais” e na base delas está o software de virtualização, que é o responsável em interagir com o sistema operacional

“hospedeiro” e utilizar o recurso de hardware disponível. Em suma, podemos ver na figura abaixo uma representação do conceito “computador dentro do outro”.



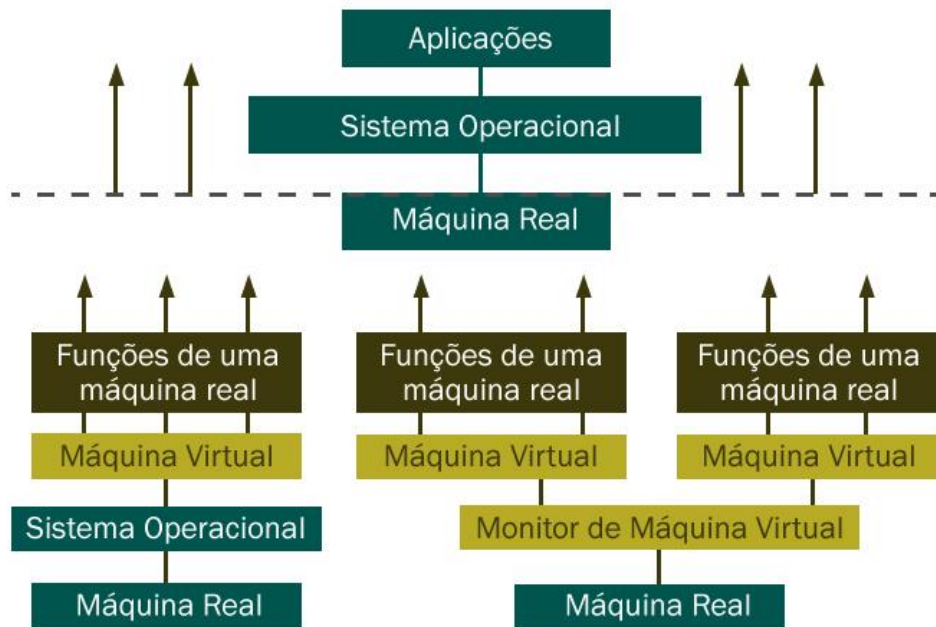
Mas como isso pode ser feito? Como funciona? Hoje em dia, criar uma máquina virtual é algo muito simples. Para isso, é necessária apenas a instalação de um programa específico no seu computador, da mesma maneira que você já instalou outro qualquer. Será por meio desse programa (software) que você poderá criar os componentes básicos para o funcionamento de uma máquina real, exemplo: um HD (disco rígido) virtual, memória RAM virtual, placa de rede virtual. Tudo isso tornará possível executar um sistema operacional inteiro a partir dele.

22

A virtualização de máquinas é extremamente útil nas tarefas do dia a dia, pois torna possível ao usuário executar (rodar) e testar outros sistemas operacionais, tendo acesso a todos os *softwares* que necessita. Você poderia utilizar um *software*, por exemplo o VirtualBox, para fazer um *test-drive* de um novo sistema operacional, sem necessidade de reinstalar seu computador, evitando diversos transtornos.

Durante a virtualização de máquina você irá determinar o quanto será alocado de memória RAM, emulando um ambiente de computação física, quais serão as especificações de CPU, memória, disco rígido, rede e outros recursos de *hardware*. Tudo isso será gerenciado pelo *software* de virtualização, em uma camada chamada “camada de virtualização”, que faz a tradução das solicitações da máquina virtual para o *hardware* presente na máquina real.

As máquinas virtuais tornam possível a instalação dos mesmos programas que você utiliza em seu computador, esses programas entendem que estão sendo executados diretamente no *hardware* físico, e não dentro de uma simulação. Peço que observe a figura abaixo para entender como e onde a máquina virtual ou virtualização de máquina intermedia a comunicação com a máquina real.



23

RESUMO

Durante o estudo deste módulo, tivemos a oportunidade de entender como é feita a representação das informações. Entendemos que toda informação que é introduzida em um computador (sejam dados ou instruções de um programa) necessita ser entendida pela máquina, para que possa interpretá-la e processá-la. Estudamos que a menor unidade de informação armazenável em um computador ou sistema computacional é o algarismo binário ou dígito binário, conhecido como bit (inglês= **binary digit**). Entendemos que um byte é definido como um grupo ordenado de 8 **bits**, tratados de forma individual, como unidade de armazenamento e transferência. Compreendemos que “palavra” está relacionada com o armazenamento e a transferência de informações entre MP (Memória Principal) e UCP e especialmente relacionado ao processamento de dados pela CPU. Vimos a importância do entendimento de “níveis de máquina” em virtude que neste módulo falarmos do nível “mais baixo” que é como é feita a representação dos dados. Também estudamos que virtualização de máquina é o nome dado ao processo de criação de uma máquina, criada por meio de um software, que executa programas como se fosse um computador real.

UNIDADE 1 – FUNDAMENTOS DA ARQUITETURA E REPRESENTAÇÃO DE DADOS NO SISTEMA OPERACIONAL

MÓDULO 3 – NOTAÇÃO POSICIONAL

01

1 - SISTEMA DE NUMERAÇÃO DECIMAL, BINÁRIO E HEXADECIMAL.

Nesse módulo iremos estudar os Sistemas de Numeração, os conceitos, as bases de numeração, número e numeral. Necessidade de representação em outras bases de numeração. Sistemas Decimal, binário, octal e hexadecimal. Conversão entre bases: da base 10 para uma base qualquer; de uma base qualquer para base 10.

Desde que as primeiras civilizações buscando formas de perpetuar conhecimento por meio da escrita, o ser humano vem adotando maneiras e métodos específicos para representar números, possibilitando, com eles, contar objetos e efetuar operações aritméticas (soma, subtração, multiplicação, etc.). A forma mais empregada de representação numérica é a chamada notação posicional. Nela, os algarismos componentes de um número assumem valores diferentes, dependendo de sua posição relativa no número. O valor total do número é a soma dos valores relativos de cada algarismo. Desse modo, é a posição do algarismo ou dígito que determina seu valor.

A formação de números e as operações com eles efetuadas dependem, nos sistemas posicionais, da quantidade de algarismos diferentes disponíveis no referido sistema. Há muito tempo, nós do mundo ocidental, adotamos um sistema de numeração que possui dez diferentes algarismos - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 - e, por essa razão, foi chamado de sistema decimal. A quantidade de algarismos disponíveis em um dado sistema de numeração é chamada de base; a base serve para contarmos grandezas maiores, indicando a noção de grupamento. O sistema de dez algarismos, acima mencionado, tem base 10; um outro sistema que possua apenas dois algarismos diferentes (0 e 1) é de base 2, o que possui 16 algarismos diferentes é o de base 16, e assim por diante.

02

Generalizando, num sistema qualquer de numeração posicional, um número N é expresso da seguinte forma:

$$N = (d_{n-1} d_{n-2} d_{n-3} \dots d_1 d_0)_b$$

onde:

d indica cada algarismo do número;

n-1, n-2, 1, 0 índices, indicam a posição de cada algarismo;

b indica a base de numeração;

n indica o número de dígitos inteiros.

O valor do número pod e ser obtido do seguinte somatório:

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

Agora vamos utilizar como exemplo o número 3748, estando na base 10, podemos representar da seguinte forma:

N = 3748

onde:

N **4** (quatro dígitos inteiros).

Utilizando a fórmula indicada acima:

$$d_{n-1} = 3 \text{ ou } d_3 = 3 / d_2 = 7 / d_1 = 4 / d_0 = 8$$

ou obtendo seu valor de acordo com a fórmula mostrada acima:

$$N = 3 \times 10^3 + 7 \times 10^2 + 4 \times 10^1 + 8 \times 10^0$$

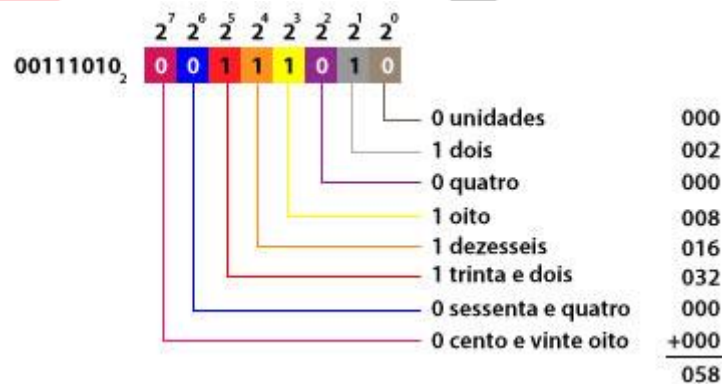
$$= 3000 + 700 + 40 + 8 = 374810$$

03

O sistema de numeração é uma forma sistêmica (metódica, coerente) de utilização de numerais (conjuntos de um ou mais algarismos) para a representação dos números. Como vimos anteriormente, o sistema numérico mais conhecido é o decimal, chamado de “base 10”. Entretanto, existem outros sistemas numéricos, são eles:

- Binário (base 2)
- Octal (base 8)
- Hexadecimal (base 16)

A figura abaixo representa um número na base binária (base 2).



04

Após esta pequena introdução sobre o que venha a ser a notação posicional e bases numéricas, gostaríamos que você respondesse a seguinte pergunta:

Em sua opinião, qual o sistema numérico utilizado nos computadores?



Caso você tenha respondido o sistema binário, parabéns, você acertou!

Vamos aproveitar que você está indo muito bem e vamos fazer outra pergunta:

Por que é utilizado o sistema binário e não o decimal nos computadores, pois diariamente utilizamos o sistema decimal?



Caso você tenha respondido que o sistema decimal (base 10) iria tornar os circuitos digitais inviáveis economicamente e seriam difíceis de implementar em relação aos com circuitos digitais (ligado/desligado), você está de parabéns, acertou.

Caso você pare para pensar um pouco e veja que ao juntarmos os dedos da mão direita e esquerda teremos dez dedos, o que torna mais fácil o ensino das operações básicas da matemática. Todos nós já recorremos às nossas mãos para fazer contas, não é mesmo?



05

2 - SISTEMAS NUMÉRICOS BASE 10

Como falamos anteriormente e com o intuito de reforçar o conceito, quando dizemos Sistema Numérico base 10 é quando utilizamos 10 algarismos (símbolos) para representar qualquer quantidade (0 a 9).

Exemplo:

0 1 2 3 4 5 6 7 8 9

Matematicamente falando, quando usamos a base é 10, entendemos que para cada posição à esquerda, o peso será 10 vezes maior do que a posição à direita. Para entendermos melhor veja este exemplo abaixo:

$$(5248)_{10} = 5 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 8 \times 10^0$$

Você pode notar que o número 5 é o que está mais à esquerda, na casa decimal milhar, contando a partir do número 8, temos três casas. Com o intuito de facilitar o entendimento colocamos desta forma:

$$5 \times 10^3$$

Para os números 2, 4 e 8 segue-se o mesmo raciocínio.

06

Base 2

A base do sistema binário é 2 e, conseqüentemente, qualquer número, quando representado nesse sistema, consiste exclusivamente em dígitos 0 e 1. O termo dígito binário é chamado bit, contração do termo inglês binary digit.

Por exemplo, o número binário **11011** possui cinco dígitos, ou algarismos binários. Diz-se que o referido número é constituído de 5 bits. Ao falarmos do Sistema numérico base 2, sistema usado pelos computadores, observamos que este sistema utiliza somente dois algarismos para representar qualquer quantidade. A origem do nome é por ser apenas dois, então se chama binário. São eles:

0 1

Aproveitamos a base 2 para falarmos também do termo bit. Este termo vem das palavras *binary digit* $(01001)_2$

07

Base 16

Em bases de valor superior a 10, usam-se letras do alfabeto para a representação de algarismos maiores que 9. Uma dessas bases é especialmente importante em computação; trata-se da base 16 ou hexadecimal, por ser de valor múltiplo de 2 (como a base 8).

O sistema hexadecimal, também chamado de sistema de base 16 é utilizado nos projetos computacionais (hardware e software), também utilizado por ser mais próximo da base 10, é de fácil conversão para sistema binário. Como próprio nome diz, ele é composto por 16 símbolos, representados da seguinte forma:

0 1 2 3 4 5 6 7 8 9 A B C D E F

Nessa base, os “algarismos” A, B, C, D, E e F representam, respectivamente, os valores (da base 10): 10, 11, 12, 13, 14 e 15.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| | | | | | | | | | | 10 | 11 | 12 | 13 | 14 | 15 |

Um número nessa base é representado na forma da expressão:

(1A7B)₁₆

O seu valor na base 10 será obtido usando-se a expressão:

$$1 \times 16^3 + 10 \times 16^2 + 7 \times 16^1 + 11 \times 16^0 = 4096 + 2560 + 112 + 11 = 6779_{10}$$

Observemos que na fórmula foram usados os valores 10 (para o algarismo A) e 11 (para o algarismo B) para multiplicar as potências de 16. Por isso, obtivemos o valor do número na base 10.

08

Em outras palavras, utilizamos valores e regras de aritmética da base 10 e, por isso, o resultado encontrado é um valor decimal. A tabela abaixo mostra a representação de números nas bases 2, 8, 10 e 16.

| Base 2 | Base 8 | Base 10 | Base 16 |
|--------|--------|---------|---------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 10 | 2 | 2 | 2 |
| 11 | 3 | 3 | 3 |
| 100 | 4 | 4 | 4 |
| 101 | 5 | 5 | 5 |
| 110 | 6 | 6 | 6 |
| 111 | 7 | 7 | 7 |
| 1000 | 10 | 8 | 8 |
| 1001 | 11 | 9 | 9 |
| 1010 | 12 | 10 | A |
| 1011 | 13 | 11 | B |
| 1100 | 14 | 12 | C |
| 1101 | 15 | 13 | D |
| 1110 | 16 | 14 | E |
| 1111 | 17 | 15 | F |

09

3 - CONVERSÃO DE BASES

Uma vez entendido como representar números em notação posicional, e como esta notação é aplicável em qualquer base inteira, podemos exercitar a conversão de números de uma base para outra. Interessa-nos, principalmente, verificar o processo de conversão entre bases múltiplas de 2, e entre estas e a base 10, e vice-versa.

Conversão entre Bases Potência de 2

• Entre as Bases 2 e 8

Em virtude do numeral oito ser igual ao numeral dois elevado a três ($8 = 2^3$), um número binário (base 2) pode ser facilmente convertido para o seu valor equivalente na base 8 (octal). Se o número binário for inteiro, basta dividi-lo, da direita para a esquerda, em grupos de 3 bits (o último grupo não sendo múltiplo de 3, preenche-se com zeros à esquerda – exemplo 2). Então, para cada grupo, acha-se o algarismo octal equivalente. A conversão de números da base 8 para a 2 é realizada de forma semelhante, no sentido inverso; substitui-se cada algarismo octal pelos seus 3 bits correspondentes.

Preste atenção nos seguintes exemplos:

$$1) (111\ 010\ 111)_2 = ()$$

$$(111)\ (010)\ (111)_2 = (727)_8$$

$$7\quad 2\quad 7$$

$$2) (1\ 010\ 011\ 111)_2 = ()_8$$

$$(001)\ (010)\ (011)\ (111)_2 = (1237)_8$$

$$1\quad 2\quad 3\quad 7$$

$$3) (327)_8 = ()_2$$

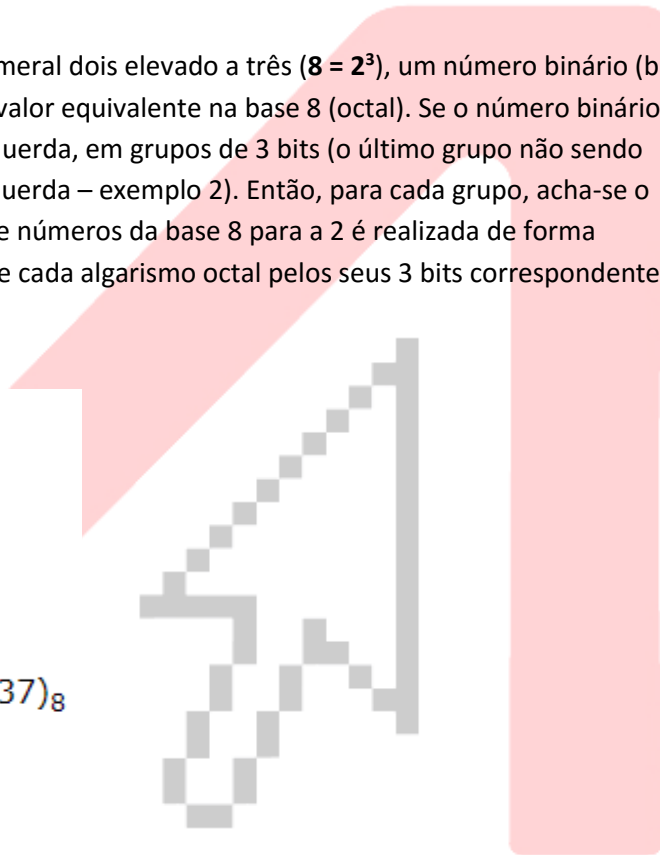
$$(011)\ (010)\ (111)_2 = (11010111)_2$$

$$3\quad 2\quad 7$$

$$4) (673)_8 = ()_2$$

$$(110)\ (111)\ (011)_2 = (110111011)_2$$

$$6\quad 7\quad 3$$



• Conversão entre as Bases 2 e 16

O procedimento de conversão entre números binários e hexadecimais (base 16) é idêntico ao da conversão entre as bases 2 e 8, exceto que, neste caso, a relação é $16 = 2^4$.

Desse modo, um algarismo hexadecimal é representado por 4 bits; converte-se um número binário em hexadecimal, dividindo-se este número em grupos de 4 bits da direita para a esquerda. A conversão de hexadecimal para binário é obtida substituindo-se o algarismo hexadecimal pelos 4 bits correspondentes. Caso o último bit não tenha o total de 4 bits, preenche-se com zeros à esquerda.

Observe a seguinte tabela para auxiliar na resolução e compreensão dos exercícios abaixo:

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Preste atenção nos seguintes exemplos:

1) Converta da base 2 para base 16.
 $(10\ 1101\ 1011)_2 = ()_{16}$

$(0010)\ (1101)\ (1011)_2 = (2DB)_{16}$
 2 D B

Resolução do exemplo 1

2) Converta da base 2 para base 16
 $(10011100101101)_2 =$

$(0010)\ (0111)\ (0010)\ (1101)_2 = (272D)_{16}$
 2 7 2 D

3) Converta da base 16 para base 2
 $(306)_{16} = ()_2$

$(0011)\ (0000)\ (0110)_2 = (1100000110)_2$
 3 0 6

4) Converta da base 16 para base 2 $(F50)_{16} = ()_2$

$(1111)\ (0101)\ (0000)_2 = (111101010000)_2$
 F 5 0

Resolução do exemplo 1

Observando da esquerda para a direita, temos (0010). O bit que está “ligado”, o bit 1, está na posição do 2¹, que é igual a 2 // por isso a resposta é 2. No segundo, (1101), temos o 1º, o 2º e o 4º bit ligado, então somando $2^3 + 2^2 + 2^0$ temos o número 13, que corresponde na tabela hexadecimal a D. No terceiro, (1011), temos o 1º, o 3º e o 4º bit ligado, então somando $2^3 + 2^1 + 2^0$ temos o número 11, que corresponde na tabela hexadecimal a B.

| | | |
|-------------------------|-------------------------|-------------------------|
| (0010) | (1101) | (1011) |
| $2_3 \ 2_2 \ 2_1 \ 2_0$ | $2_3 \ 2_2 \ 2_1 \ 2_0$ | $2_3 \ 2_2 \ 2_1 \ 2_0$ |

11**4 - CONVERSÃO DE NÚMEROS DE UMA BASE B PARA A BASE 10**

A conversão de um número, representado em uma base B qualquer, para seu correspondente valor na base 10 é realizada empregando-se a fórmula. A melhor maneira de compreender o processo de conversão consiste na realização de alguns exemplos práticos, onde se indica, detalhadamente, a aplicação da referida fórmula.

Os exemplos apresentados referem-se apenas a números inteiros.

1) $(101101)_2 = ()_{10}$

Substituindo, na expressão, as letras pelos valores do exemplo, teremos:

$b = 2$ (a base origem do número a ser convertido)

$n = 6$ (6 algarismos);

$n - 1 = 5$ (expoente do W produto mais à esquerda)

$d_1 = 1$

1º produto: $d_{n-1} \times b^{n-1} = 1 \times 2^5$

Os demais produtos seguem a sequenciada expressão, resultando em:

$$(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

$$= 32 + 0 + 8 + 4 + 0 + 1 = (45)_{10}$$

Exemplo 2

Exemplo 3

Exemplo 4

Exemplo 2

$$(27)_8 = ()_{10}$$

Da mesma maneira, substitui-se na expressão:

$$b=8$$

$$n=2$$

$$n-1=1$$

$$d_{n-1}=2$$

Valor total:

$$2 \times 8^1 + 7 \times 8^0 = 16 + 7 = (23)_{10}$$

Exemplo 3

$$(2A5)_{16} = ()_{10}$$

$$2 \times 16^2 + 10 \times 16^1 + 5 \times 16^0$$

$$= 512 + 160 + 5$$

$$= (677)_{10}$$

Observação: Lembre-se que A representa 10 em hexadecimal.

Exemplo 4

$$(6734)_8 = ()_{10}$$

$$6 \times 8^3 + 7 \times 8^2 + 3 \times 8^1 + 4 \times 8^0$$

$$= 3072 + 448 + 24 + 4 = (3548)_{10}$$

12

5 - CONVERSÕES DE NÚMEROS DECIMAIS PARA UMA BASE B

A conversão de números, representados na base 10, para seus valores equivalentes em uma base B qualquer é efetuada através de um processo inverso ao do subitem anterior (base B para base 10).

A conversão é obtida dividindo-se o número decimal pelo valor da base desejada; o resto encontrado é o algarismo menos significativo do valor na base B (mais à direita). Em seguida, divide-se o quociente

encontrado pela base B; o resto é o outro algarismo (à esquerda); e assim, sucessivamente, vão-se dividindo os quocientes pelo valor da base até se obter quociente de valor zero. Em cada divisão, o resto encontrado é um algarismo significativo do número na nova base; o primeiro resto encontrado é o valor do algarismo menos significativo e o último resto será o algarismo mais significativo (mais à esquerda).

Na realidade, o algoritmo de conversão pode ser definido com vários critérios de parada, tais como:

a) Enquanto quociente for diferente de zero:

- dividir dividendo por divisor
- extrair resto como algarismo e colocá-lo à esquerda do anterior
- repetir

b) Quando quociente for igual a zero, parar.

c) Enquanto dividendo for maior que divisor:

- dividir dividendo por divisor
- extrair resto como algarismo e colocá-lo à esquerda do anterior
- repetir

Usar o dividendo (que agora é menor que o divisor) como último algarismo à esquerda (algarismo mais significativo).

Exemplo

Exemplo

$(3964)_{10} = ()_8$
 $3964/8 = 495 \Rightarrow \text{resto } 0 = 4$
 (algarismo menos significativo)
 $495/8 = 61 \Rightarrow \text{resto } = 7$
 $61/8 = 7 \Rightarrow \text{resto } 2 = 5$
 $7/8 = 0 \Rightarrow \text{resto } 3 = 7$ (algarismo mais significativo)
 O número é, então, $(7574)_8$

13

Exercício 02:

$(2293)_{10} = ()_8$

O resultado será: **4365**

$$\begin{array}{r|l} 2293 & 8 \\ \hline 5 & 286 & 8 \\ \hline 6 & 35 & 8 \\ \hline 3 & 4 & 8 \\ \hline 4 & & 0 \end{array}$$

Veja a figura abaixo para compreender melhor essa conversão:

CONVERSÃO DE BASES:

Base 10 para base 8:

Exemplo: $(2293)_{10} = (4365)_8$

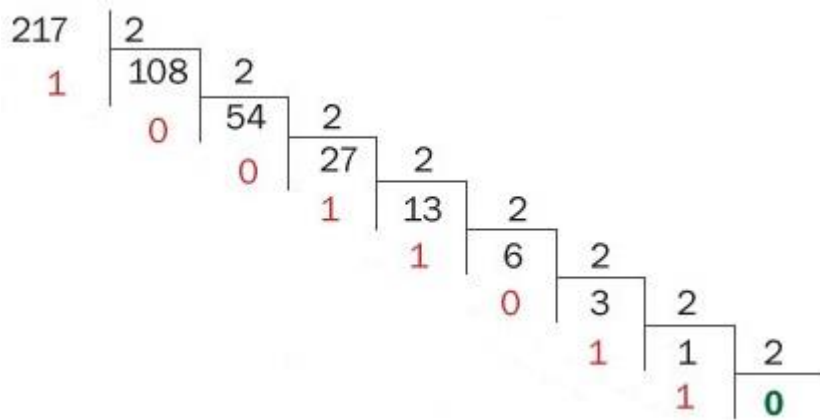
| | | | | | |
|------|-----|----|---|---|---|
| 2293 | 8 | 8 | 8 | 8 | 8 |
| | 286 | 35 | 4 | | 0 |
| 5 | 6 | 3 | 4 | | |

Divide-se o número sucessivamente por oito até obter-se o quociente **ZERO**.
O resultado é obtido tomando-se os restos das divisões ordenados do último para o primeiro.

14

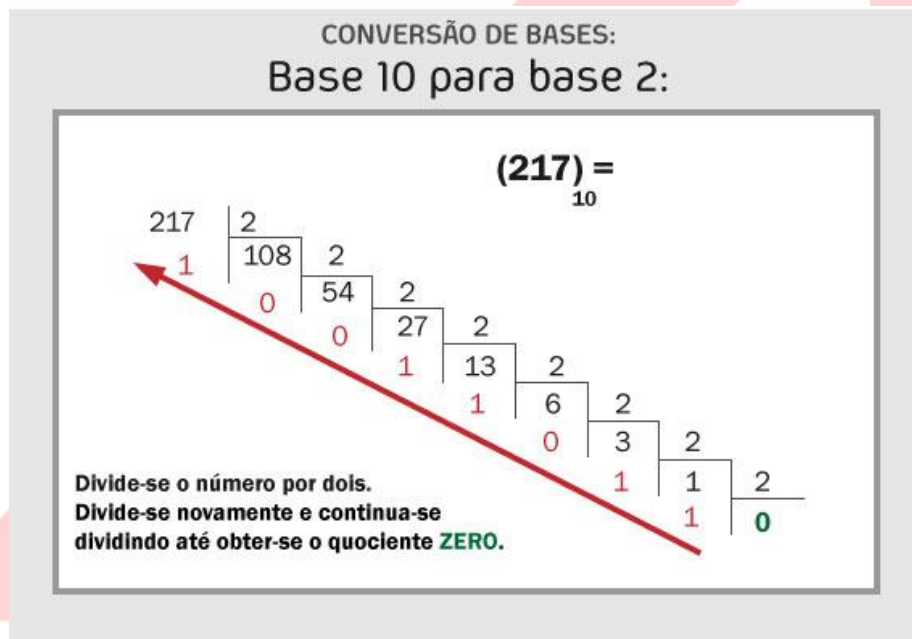
Exercício 03:

$$(217)_{10} = ()_2$$



O resultado será: **11011001**

Veja a figura abaixo para entender esta conversão:



15

Para ajudar na compreensão de conversão de base, assista ao vídeo a seguir.

1º Aula de Eletrônica Digital - Bases Numéricas - Conversão de Bases

https://www.youtube.com/watch?v=loD6Br_cggc

16

RESUMO

Durante o estudo deste módulo tivemos a oportunidade de conhecer o que são os sistemas de numeração decimal. Vimos que nós, do mundo ocidental, adotamos um sistema de numeração que possui dez diferentes algarismos - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 - e, por essa razão, foi chamado de sistema decimal. O sistema de numeração é uma forma sistêmica (metódica, coerente) de utilização de numerais (conjuntos de um ou mais algarismos) para a representação dos números. Como vimos anteriormente, o sistema numérico mais conhecido é o decimal, chamado de “base 10”. Contudo, há outros sistemas numéricos, como o binário (base 2), Octal (base 8) e o Hexadecimal (base 16). Finalmente, uma vez que entendemos como é a representação dos números, fizemos vários exercícios sobre a conversão de números de uma base para outra com o intuito de fixar o entendimento do assunto.

