# INTRODUÇÃO À PROGRAMAÇÃO

Aula 01 - Introdução

Prof. Me. Sérgio Carlos Portari Júnior www.sergioportari.com.br sergio.junior@uemg.br

# TÓPICOS ABORDADOS

- Comparação homem x computador
- Entendendo o computador
- Linguagens de Programação
- Tipos de traduções nas Linguagens de Programações

#### Comparação homem x computador

#### PESSOA

- Baixa velocidade
- Cansa rapidamente e torna-se sujeito a erro quando o trabalho é repetitivo
- Muito flexível, opera com instruções imprecisas
- Memória limitada e com o tempo de acesso elevado
- Dificuldade de consultar grandes volumes de informações.

#### COMPUTADOR

- Alta velocidade
- Não cansa e é extremamente preciso
- Totalmente estruturado: opera somente com instruções rigorosamente precisas
- Memória praticamente ilimitada, pequeno tempo de acesso
- Facilidade de consultar grandes volumes de informações

#### Comparação homem x computador Tomada de decisões

#### Pessoa

- Podem criar conceitos gerais e empregam o raciocínio para tirar conclusões
- Nem sempre optam pela melhor solução por causa do grande nº de alternativas
- Podem decidir mesmo se o conjunto de dados for impreciso
- Definem prioridades, objetivos e valores
- Podem visualizar novas oportunidades

#### Computador

- Não pode operar nestas condições
- Podem adotar a melhor solução, desde que tenham dados suficientes
- Normalmente, exigem um conjunto completo das informações
- Exigem que sejam determinadas as prioridades, objetivos e valores
- Não tem essa capacidade

# Computador

- Características principais
  - Altíssima velocidade de cálculo
  - Não comete erros em cálculos
  - Desempenho constante, com mesmo nível de excelência em qualquer hora do dia

- Maior deficiência
  - Inteligência limitadíssima

# Exemplo de aplicações com computadores

- Calcular e imprimir cheques de pagamento
- Controlar o estoque de uma drogaria com 14000 itens no estoque
- Executar cálculos para construção de uma ponte
- Emitir e controlar os pagamentos das contas de luz da cidade de São Paulo

# Exemplo de aplicações com computadores

- Para que o computador execute estas funções, necessitamos de roteiros ou receitas que expliquem ao computador como processar a função desejada.
- Esse roteiro ou receita em Sistemas de Informações é chamado de PROGRAMA.
- Tais programas devem ser escritos respeitando-se as limitações do computador.

 O computador é projetado para executar a atividade de processamento de dados.
 Basicamente, este processamento é composto de três etapas:



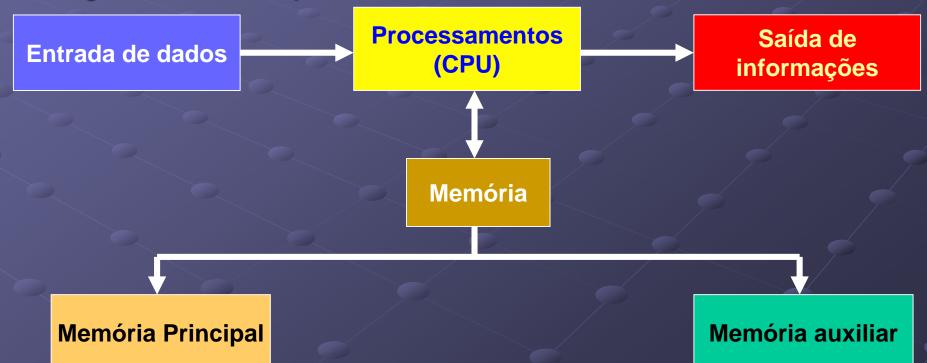
#### • Exemplos:

- Emitir um pedido para um fornecedor:
  - Dados: Nome do cliente, endereço, data da compra, data da entrega, produto, quantidade, custo unitário.
  - Processamento: Multiplicar a quantidade comprada pelo preço unitário obtendo o custo total do pedido, acumulando os valores para emissão do valor total ao final.
  - Saída: Impressão do pedido e do valor total.

#### • Exemplos:

- Cadastrar um novo funcionário
  - Dados: Os dados dos funcionário transcritos em uma ficha (como nome, data de nascimento, documentos, endereço, telefone, e-mail, etc).
  - Processamento: Digitação dos dados.
  - Saída: gravação dos dados em um "banco de dados".

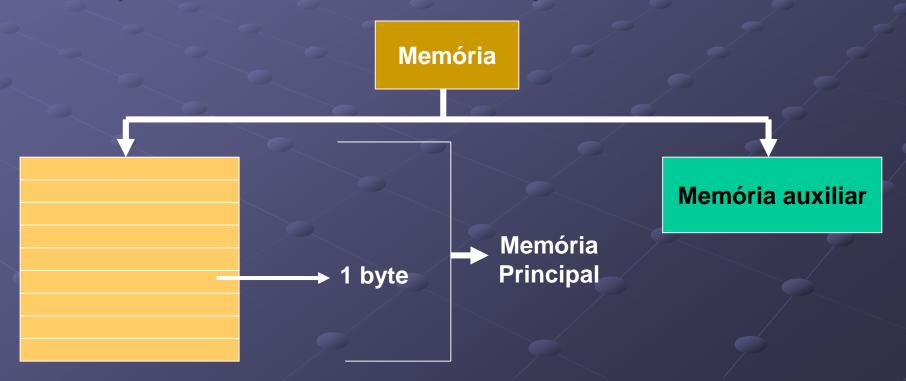
• Uma máquina projetada para esses processamentos terão necessariamente sempre estes três módulos, mais um quarto módulo: a memória, que é subdividida em memória principal (ou eletrônica) e auxiliar. Veja no gráfico o exemplo:



- A memória principal existe fisicamente, a partir recursos de eletrônica. Basicamente, ela permite-nos criar dois estados mutuamente exclusivos: LIGADO ou DESLIGADO, como se fosse uma lâmpada elétrica.
- Se associarmos o número 1 (um) para LIGADO e 0 (zero) para desligado, podemos simular o sistema binário de numeração.
- Na computação, um circuito que permite representar esses estados (ligado e desligado, ou 1 e 0) é chamado de BIT (BInary digiT).

- Um bit não é suficiente para representarmos os dados usados no dia a dia.
- Esta situação é equivalente a uma calculadora de seis dígitos. Se você precisar digitar um número com oito dígitos terá que comprar uma calculadora com visor maior.
- Para melhorar a capacidade de representação dos bits no dia a dia precisamos agrupá-los. Do ponto de vista prático, mostrou-se adequado para representar os dados do dia a dia (letras, números, caracteres especiais, etc) um conjunto de 8 bits. Esse conjunto recebeu o nome de Byte.

- Para quem usa computadores, a unidade de medida básica é o byte. Bit é utilizado na maioria das vezes por engenheiros eletrônicos.
- A memória principal é composta por um conjunto de caixas empilhadas, cada uma chamada de byte.



Nessas condições, dizer que um computador de 512 MB de memória significa dizer que ele possui aproximadamente 512 Milhões de caixas empilhadas, cada uma com 1 byte de capacidade, para armazenamento de números no formato binário.

# Linguagens de Programação

- Conforme dito anteriormente, o computador tem uma "inteligência limitadíssima", portanto é necessário darmos roteiros de trabalho para que ele atenda nossa necessidade.
- Esses roteiros tem que chegar ao computador com uma linguagem que ele entende, que é chamada de "linguagem de máquina".
- Essa linguagem é basicamente binária e seria impossível uma pessoa "normal" desenvolver programas complexos com seqüências de zeros e uns.
- Para resolver este "problema de comunicação" foram criados programas capazes de traduzir uma sintaxe que as pessoas entendam para uma linguagem que a máquina entenda. Estes programas são chamados "linguagens de programação".

#### Linguagens de Programação

- Evolução das Linguagens:
  - 1ª Geração: correspondem às linguagens de máquina.
  - 2ª Geração: Linguagem Assembler (anos 1950) instruções de máquinas agrupadas em comandos de 3 letras, como RCL, ADD, STO, JMP, etc.
  - 3ª Geração: Sintaxes direcionadas para determinados tipos de aplicações. Criam-se seqüências de comandos com palavras geralmente em inglês, facilmente traduzidas, que são traduzidas por programas especiais para linguagem de máquina. Exemplo: Cobol, Fortran, Algol, C, Java, Pascal, Delphi, Dbase, Clipper, etc.
  - 4ª Geração: Linguagens muito parecidas com as que utilizamos no dia a dia por pessoas que falam inglês. Podem ser utilizadas por qualquer pessoa, seguindo suas regras. Exemplo: SQL

# Linguagens de Programação

O programa escrito, em qualquer uma das linguagens acima citadas, são chamados de "programas fontes".

Já os programas prontos para serem utilizados pelo computador, que resultam do programa fonte, são chamados "programas executáveis".

#### Tipos de tradução de programas

- Existem basicamente dois tipos de tradução de programas:
  - COMPILAÇÃO
  - INTERPRETAÇÃO
- Veremos as diferenças entre os dois processos

- A tradução pelo processo de compilação é o processo onde o programa fonte é traduzido para a linguagem de máquina por outro programa (chamado de compilador).
- Após a tradução o usuário poderá executar o programa sem depender de nenhum outro aplicativo.
- Exemplos de linguagens compiladoras: Cobol, Fortran, Pascal, Algol, Clipper, C, Delphi, Visual Basic, entre outras.

Neste processo de tradução o programa tradutor (linguagem de programação) vê o programa fonte em nível de linha e traduz para linguagem de máquina para a CPU ativar sua execução posteriormente.

# # include<stdio.h> float main() { float salario, acres,total; total=salario \* acres; printf("%f - Total",total); } COMPILADOR Programa de tradução Programa fonte escrito em linguagem de máquina

- O programa começa a ser traduzido para linguagem de máquina de cima para baixo, da esquerda para a direita, ou seja, linha a linha. Cada linha traduzida é gravada em um outro arquivo (objeto).
- Terminada a linha, o compilador passa para a linha seguinte e assim sucessivamente até o fim do programa fonte.
- Quando terminada a tradução, o programa objeto passa pelo processo de linkedição (ou linkagem), onde o arquivo objeto é escrito para a linguagem da máquina correspondente (ou seja, para aquele Sistema Operacional).
- O arquivo resultante da linkedição é o Programa Executável.

#### Interpretação

- Na tradução por interpretação o programa tradutor vê o programa fonte linha a linha, da esquerda para a direita e vai executando as funções imediatamente, traduzindo para a linguagem de máquina.
- Neste processo, não é gerado nenhum arquivo objeto ou programa executável.
- Exemplos de linguagens que são interpretadas: dBase, Batch, ASP, PHP, SQL, dentre outras.

#### Interpretação

Nestas condições, podemos dizer que a tradução e a execução são processos quase paralelos. As linhas são traduzidas todas as vezes que você executar o programa.



# Comparação de desempenho

- Quando comparamos a velocidade de execução, qualquer programa que foi compilado é mais rápido que seu equivalente interpretado.
- Ainda existem inúmeras análises que comparam as velocidades de execução entre os programas compilados e interpretados, porém mesmo sendo mais rápidos os compilados, a diferença de velocidade nem sempre é muito perceptível (geralmente entre décimos e milésimos de segundos).

#### **VANTAGENS**

- Numa venda de software, os fontes não precisam ser fornecidos;
- Na venda, o cliente não necessita de comprar o compilador;
- Velocidade de execução é maior do que a versão interpretada equivalente.

#### **DESVANTAGENS**

- Não existência de processos iterativos (o cliente não pode moldar o software à sua vontade depois de ter o executável);
- Maior dificuldade na depuração de erros no processo de desenvolvimento.

# Interpretação

#### **VANTAGENS**

- Existência de processos iterativos (o cliente pode mudar o programa à sua vontade);
- Facilidade de depuração dos erros no desenvolvimento.

#### **DESVANTAGENS**

- Na venda de um programa o cliente receberá o programa fonte;
- O cliente deverá ter a linguagem interpretadora;
- Baixa velocidade se comparado ao seu equivalente compilado.