

A02 Conceitos Básicos de Computação

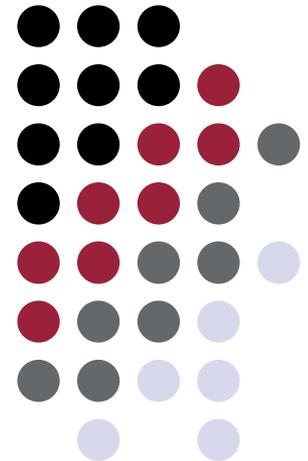


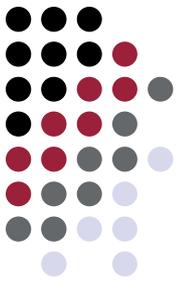
UFOP

Universidade Federal
de Ouro Preto

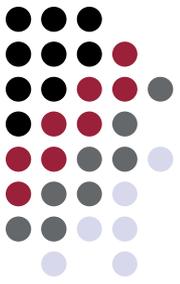
CSI030 – Programação de Computadores I

Prof. Dr. George H. G. Fonseca
Universidade Federal de Ouro Preto



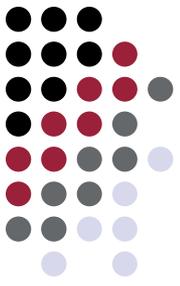


- O que é um computador?
- Como um computador é organizado internamente?
- Como funciona um computador?
- O que são algoritmos?
- O que são linguagens de programação?
- Como são criados e executados os programas?



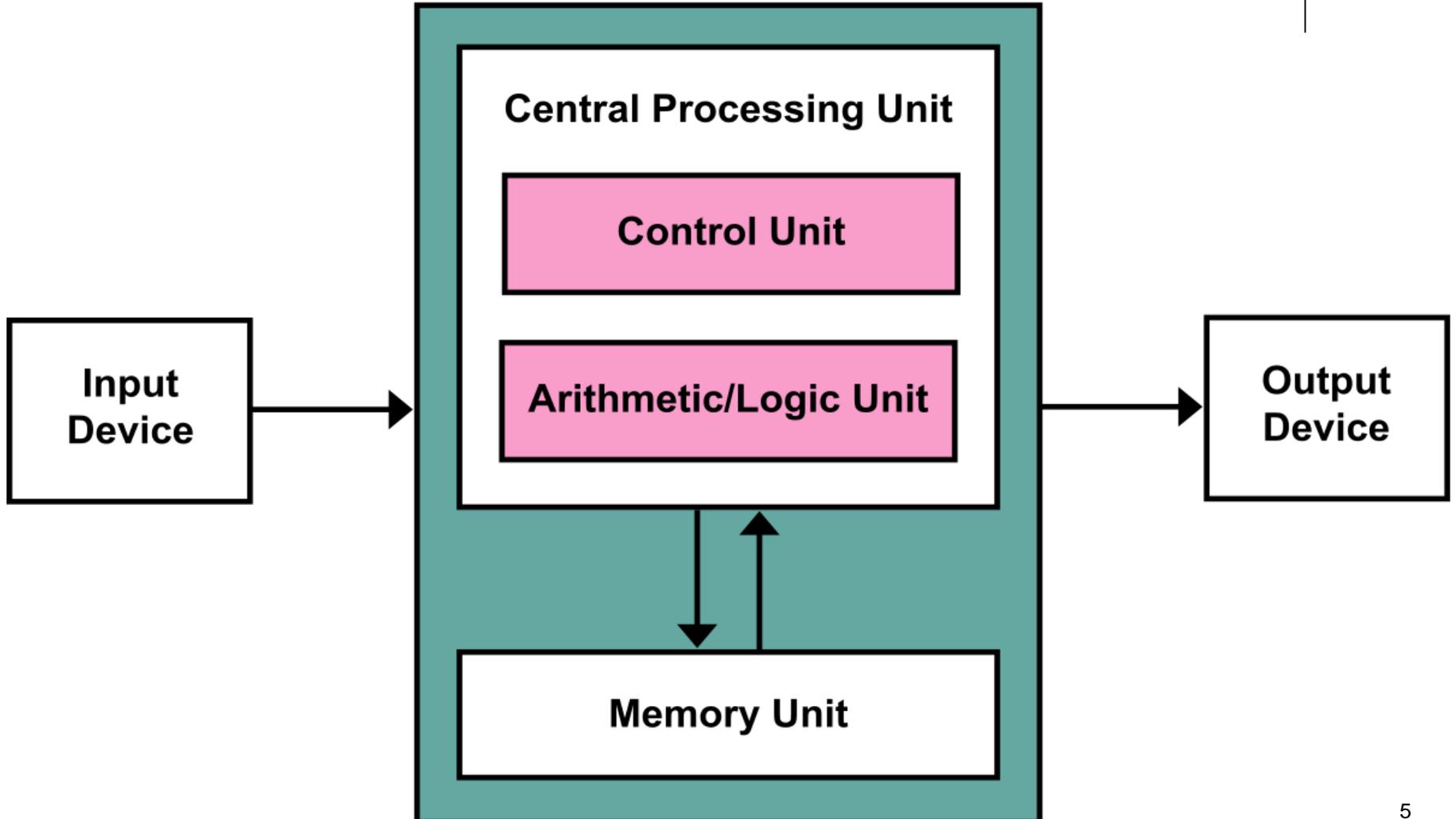
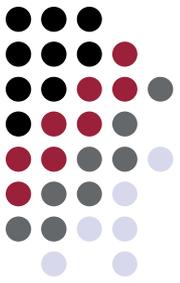
- Dispositivo capaz de realizar computações (cálculos) e tomar decisões lógicas muito mais rapidamente que nós humanos
- Composto de parte física (hardware) e parte lógica (software)
- Processa informações de acordo com um conjunto de comandos que formam um programa
- Esses comandos são escritos em uma linguagem de programação

Organização do Computador

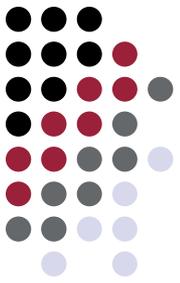


- O hardware do computador é organizado em 4 partes:
 - Unidades de saída de dados
 - Unidades de entrada de dados
 - Unidade de memória
 - Principal (RAM)
 - Secundária (HD, cartão de memória etc)
 - Unidade central de processamento
 - Unidade lógica e aritmética
 - Unidade de controle

Organização do Computador

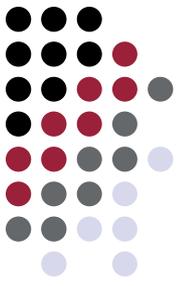


Como funciona um computador

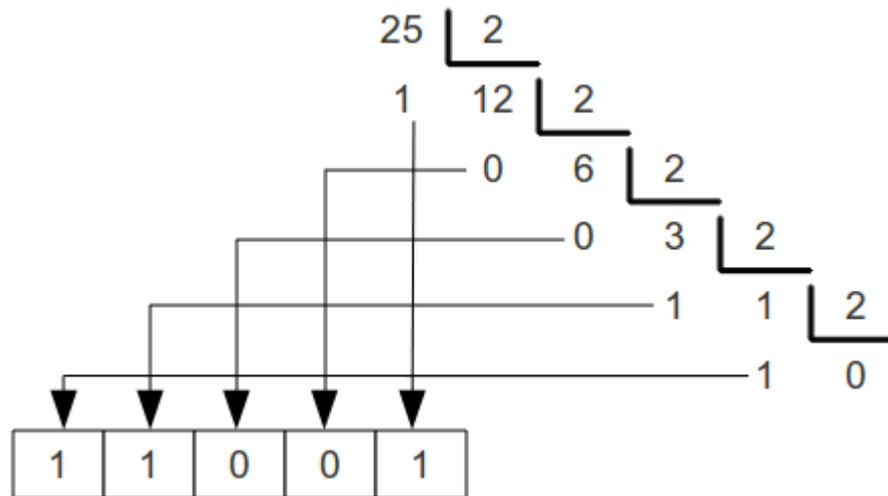


- Um programa é uma sequência de 0s e 1s armazenado na memória do computador
- O programa é executado pela CPU, que interpreta as sequências como comandos
- Aritmética binária inteira: cálculos são feitos com números inteiros representados na base 2 (binária)
- Bit: menor unidade de informação
 - Representa dois estados 0 ou 1 da lógica binária

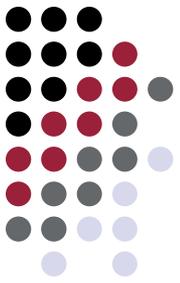
Representação Binária



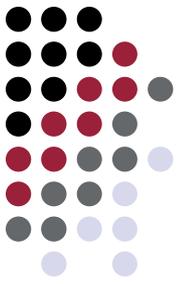
- O número 25 (base 10) equivale a 11001 (base 2).



- Formas mais comuns de armazenamento
 - 00000000 00000000 00000000 00011001: 32 bits
 - 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00011001: 64 bits

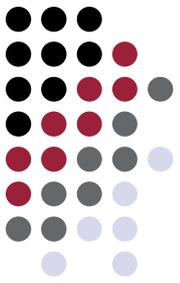


- Sequência finita de passos, precisos e bem definidos que descrevem como realizar uma tarefa
- Podem ser especificados em português, português estruturado, fluxogramas, linguagens de programação, etc



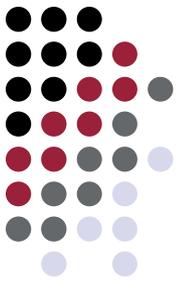
- Calcule a soma dos números 1234 e 456
 - Escreva os números em um papel, um abaixo do outro, alinhados pelo dígito das unidades. Para cada coluna de 1 ou mais dígitos, some os dígitos alinhados. Caso o valor da soma ultrapasse 9, adicione 1 à coluna imediatamente à esquerda da coluna atual e anote o dígito das unidades desta soma como o resultado da soma. Após executar todas as somas, leia o resultado final.

Algoritmo em Português Estruturado

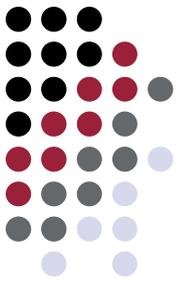


- Calcule a soma dos números 1234 e 456
 - Escreva os números em um papel, um abaixo do outro, alinhados pelo dígito das unidades.
 - Para cada coluna de 1 ou mais dígitos:
 - Some os dígitos alinhados.
 - Se o valor da soma ultrapassar 9:
 - Adicione 1 à coluna imediatamente à esquerda da coluna atual
 - Fim-se
 - Anote o dígito das unidades desta soma como o resultado da soma
 - Fim-para
 - Use a soma criada como resposta

Exercício

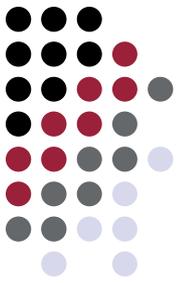


- Dados N números, escrever um algoritmo em português estruturado calcule a média desses números



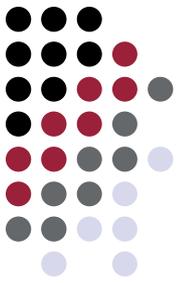
- Dados N números, escrever um algoritmo em português estruturado calcule a média desses números
 - Crie soma = 0
 - Para cada número em N
 - Faça soma = soma + número
 - Fim-para
 - Faça média = soma / quantidade de números
 - Imprima a média

Linguagens de Programação



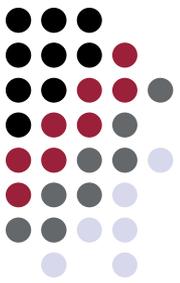
- Mecanismos utilizados para dar ordens aos computadores
- Podem ser de três tipos
 - Linguagem de máquina: diretamente entendidas pelos computadores
 - Linguagem de montagem: precisam de um programa montador para gerar linguagem de máquina
 - **Linguagem de alto nível**: precisam de um compilador para traduzi-las para linguagem de montagem

Linguagens de Máquina



- Sequências de 0s e 1s que pode ser executada por um computador
- Cada processador tem uma linguagem de máquina própria, ou seja, entende essas sequências como comandos diferentes

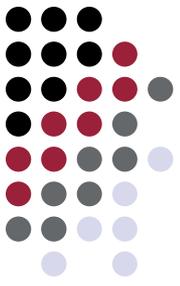
Linguagens de Montagem



- Linguagem que utiliza siglas para representar comandos em sequências de 0s e 1s. Ex: Assembly.
- Um montador traduz este código para linguagem de máquina.

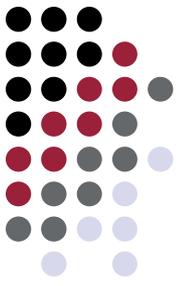
```
1  TITLE Hello                (...this is a comment area...good to give name of program)
2
3  ; This program puts basic output on the screen
4  ; Last update: 8/9/02
5
6  INCLUDE Irvine32.inc       ;libraries being called put here
7
8  .data
9  str1 byte "hello",0        ;word hello stored in the variable of type byte called str1
10 .code
11 main PROC ;program begins
12     mov edx, offset str1    ;message moved into a register where strings can be held
13     call WriteString        ;executes display of the string
14     call Crlf               ;gives a line feed
15     exit                    ;program exits here
16 main ENDP                  ;end of main function
17 END main
```

Linguagens de Programação (de Alto Nível)



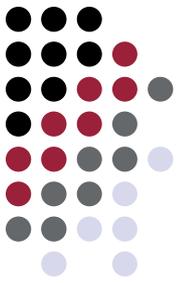
- São linguagens cujos comandos são mais próximos da linguagem humana.
 - Ex: C, C++, Java, Python, PHP, etc.
- O compilador é responsável por ler um código nesta linguagem e traduzi-lo para uma linguagem de montagem. Ex: GCC, Visual Studio, etc.
- O montador, por sua vez, gera um código executável em um determinado computador.

Do algoritmo para um programa em C



1. O programador escreve o algoritmo em um arquivo de texto, chamado código fonte
2. O compilador gera códigos objeto a partir da compilação dos códigos fonte
3. Os códigos objeto são unidos por um *linker*, gerando um programa executável
4. O código executável é carregado na memória pelo loader
5. A CPU executa as instruções carregadas

Escrevendo um programa em C



- Suponha o arquivo *fonte.c* com o conteúdo a seguir:

```
#include <stdio.h>

// Isto é um comentário!

int main( void ) {

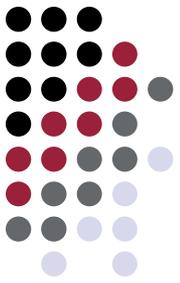
    int resultado = 1234+456;

    printf("1234 + 456 = %d \n" , resultado);

    return 0;

}
```

Estrutura de um programa em C



Declaração de Bibliotecas Utilizadas

```
// Comentários
```

```
/* Comentários */
```

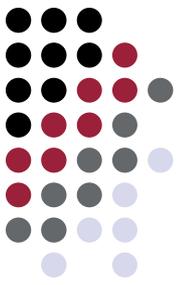
```
int main( void ) {
```

Declaração de Variáveis

Comandos

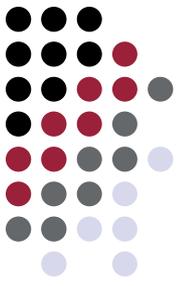
```
}
```

Compilando e executando o programa em C

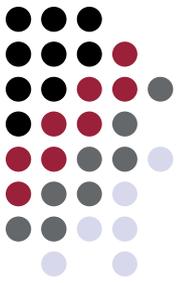


- Para compilar o programa, usamos o compilador GCC:
 - `bash$ gcc -c codigoFonte.c -o arquivoObjeto.o`
 - `bash$ gcc -o arquivoObjeto.o arquivoExecutavel.exe`
- A saída do programa exibe:
 - `1234 + 456 = 1690`

Identificando e corrigindo erros: Depuração



- Os programas podem possuir erros:
 - que os impeçam de ser compilados;
 - na implementação do algoritmo ou em sua lógica, produzindo respostas erradas.
- Para identificar o segundo tipo de erros utiliza-se um depurador de código. Ex: GDB
- O depurador permite executar o programa passo a passo, inspecionando a memória durante a execução.



- Anido, R., Notas de aula. UNICAMP, disponível em <http://www.ic.unicamp.br/~ranido/mc102/>, acessado em Maio de 2015.
- Mota, V. F., Notas de aula. UFMG. Disponível em <https://sites.google.com/site/virginiaferm/home/disciplinas> acessado em Maio de 2015.
- Deitel, P, Deitel, H. C How to Program. 6a Ed. Pearson, 2010.