

Introdução aos Computadores e às Linguagens de Programação

Disciplina de Programação de Computadores I
Universidade Federal de Ouro Preto

Agenda

- 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?



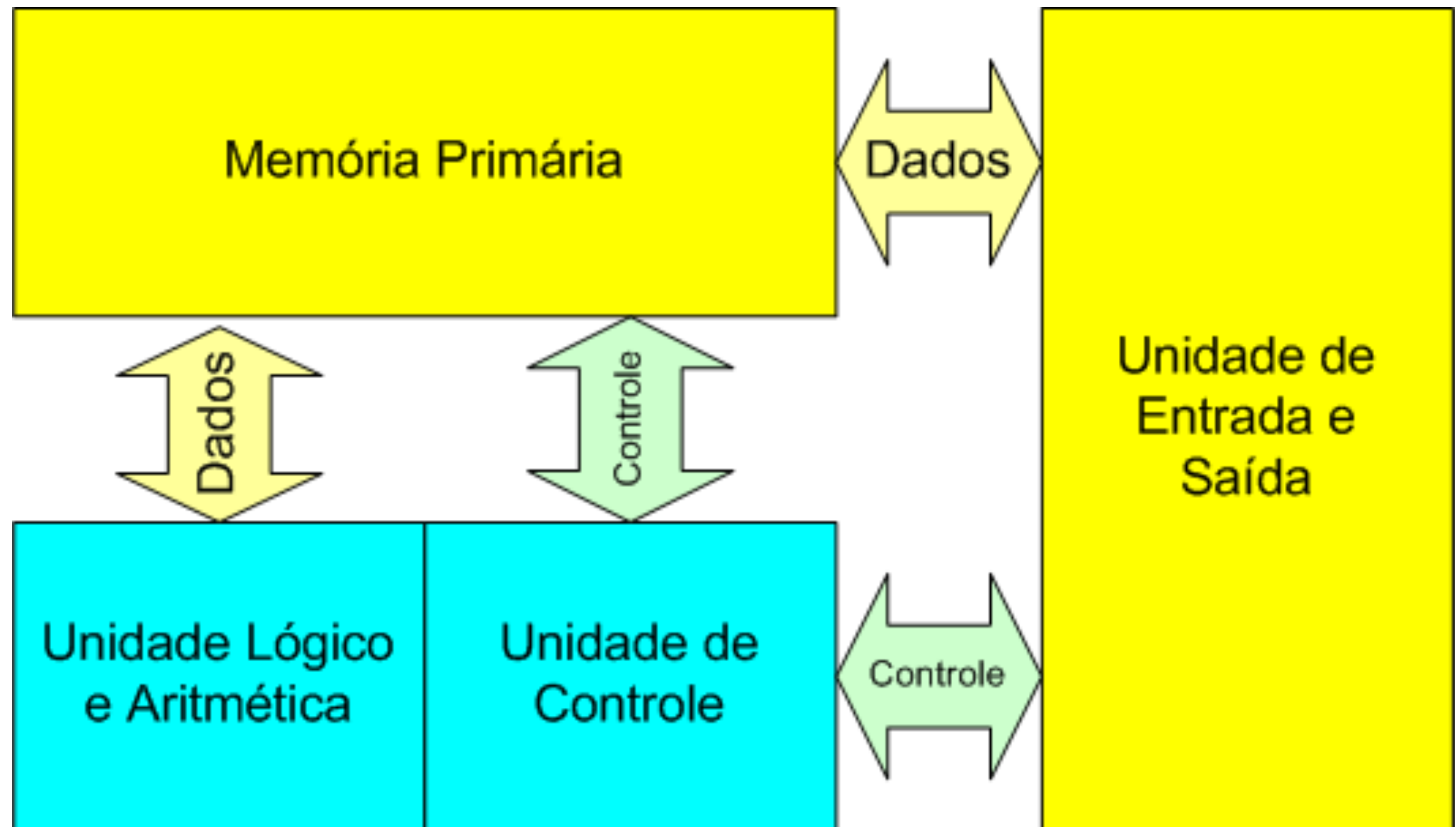
O que é um computador?

- É um dispositivo capaz de realizar computações (cálculos) e tomar decisões lógicas muito mais rapidamente que nós humanos.
- É composto de parte física (*hardware*) e parte lógica (*software*).
- Ele processa informações de acordo com um conjunto de comandos que formam um programa.
- Os comandos são escritos em uma linguagem de programação.

Organização do computador

- O hardware do computador é entendido pelo software em 6 partes:
 - Unidades de entrada de dados: teclado, mouse, etc.;
 - Unidades de saída de dados: monitor, impressora, rede, etc.;
 - Unidade de memória principal: memória RAM;
 - Unidade lógica e aritmética (ULA);
 - Unidade central de processamento (CPU);
 - Unidade de armazenamento secundário: Harddisk, DVD, pendrive.

Modelo lógico do computador



Como funciona o 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 de 0s e 1s como comandos.
- Aritmética binária inteira: cálculos são feitos com números inteiros representados na base 2.
- Bit: Menor unidade de informação (representa os dois estados - 0 e 1 - da lógica binária).

Representação binária

- O número 5 (base 10) equivale a 101 (base 2).
- O número 101 (base 2) pode ser representado como:
 - 101 (base 2): mínimo de bits necessário
 - 0101 (base 2): com 4 bits
 - 0000 0000 0000 0101 (base 2): com 16 bits
 - 0000 0000 0000 0000 0000 0000 0000 0101 (base 2): com 32 bits

Algoritmos

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

Algoritmo em português

Calcule a soma dos números 1234 e 456.

Escrever os números em um papel, um abaixo do outro, alinhados pelo dígito das unidades. Para cada coluna de 1 ou mais dígitos, somar os dígitos alinhados. Caso o valor da soma ultrapasse 9, adicionar 1 à coluna imediatamente à esquerda da coluna atual e anotar o dígito das unidades desta soma como o resultado da soma. Após executar todas as somas, ler o resultado final.

Algoritmo em Português Estruturado

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.

- Anote o dígito das unidades desta soma como o resultado da soma.

- Use a soma criada como resposta.

Exercício

- Dados N números, escrever um algoritmo em português estruturado que calcule a média destes números.

Linguagens de Programação

- São o mecanismo que utilizamos para dar ordens para os computadores.
- Podem ser de 3 tipos:
 - linguagem de máquina: diretamente entendidas pelos computadores;
 - linguagens de montagem: precisam de um programa montador para gerar linguagem de máquina
 - linguagens de alto nível: precisam de um compilador para traduzi-las para uma linguagem de montagem.

Linguagens de Máquina

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

Linguagem 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, Haskell, Prolog, 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 na linguagem C em um arquivo de texto, chamado de código fonte.
2. O compilador gera códigos objeto a partir da compilação dos códigos fonte.
3. Os códigos objeto de um ou mais códigos fonte são unidos pelo *linker*, gerando um programa executável.
4. O código executável é carregado na memória pelo *loader*.
5. A CPU executa as instruções carregadas.
6. Quando encontramos erros no funcionamento do programa, podemos depurá-lo com um depurador.

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 -o programa fonte.c
```

Para executar o programa no Linux, digitamos:

```
bash$ ./programa
```

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.

Referências Bibliográficas

- Material de aula do Prof. Ricardo Anido, da UNICAMP:
<http://www.ic.unicamp.br/~ranido/mc102/>
- Material de aula da Profa. Virgínia F. Mota:
<https://sites.google.com/site/virginiaferm/home/disciplinas>
- DEITEL, P; DEITEL, H. *C How to Program*. 6a Ed. Pearson, 2010.