

MC-102 — Aula 02

Introdução à Programação de Computadores

Instituto de Computação – Unicamp

Primeiro Semestre de 2012

Roteiro

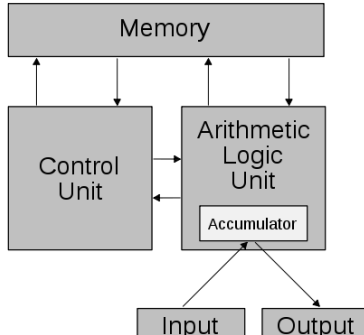
- 1 Hardware e Software
- 2 Organização de um ambiente computacional
- 3 Algoritmos
- 4 A linguagem C

O que é um computador?

- Computador: o que computa, calculador, calculista. (dicionário Houaiss).
- Os primeiros “computadores” eram humanos que calculavam tabelas de logaritmos ou trajetórias pra canhões, seguindo procedimentos bem definidos.
- Um computador é uma máquina que, a partir de uma entrada, realiza um número muito grande de cálculos matemáticos e lógicos, gerando uma saída.

Hardware e dispositivos

- Usualmente chamamos de *Hardware* todos os dispositivos físicos que compõem um computador.
- Temos por exemplo: CPU, Disco Rígido, Memória etc.
- Estes dispositivos seguem uma organização básica como na figura (Arq. de Von Neumann).



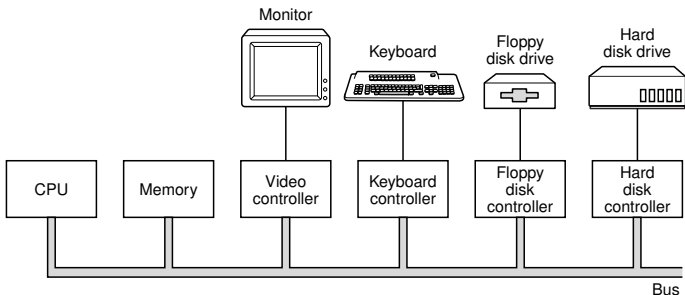
Hardware e dispositivos

Todo o hardware opera com sinais digitais: sem energia e com energia. Normalmente, usamos valores 0 e 1 (forma binária) para codificar numericamente estes sinais.

- Bit → Valores 0 ou 1.
- Byte → Agrupamento de 8 bits.
- Letras e símbolos são representados por números.

Hardware e dispositivos

- Para se comunicar os dispositivos utilizam canais denominados barramento.



Software

São os programas que executam tarefas utilizando o hardware de um computador.

- Os softwares são compostos por um conjunto de instruções que operam o hardware, além de operações lógicas e aritméticas.
- Temos abaixo, por exemplo, três instruções para um computador de 32 bits.
- Um software é composto por milhares de instruções deste tipo.

```
0100 0010 0011 0101 0101 0100 0011 0110
0100 1110 1100 1100 1001 0110 0110 1000
0000 0101 1111 1110 1101 0011 0000 1100
```

Organização básica de um ambiente computacional

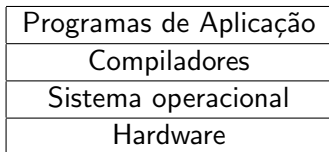
- Computadores realizam tarefas complexas por meio de um número enorme de operações simples.
- Para gerenciar a complexidade das soluções, um ambiente computacional é organizada como uma hierarquia de funções, onde cada uma é responsável por uma tarefa específica.

Programas de Aplicação	
Compiladores	Interpretadores
Sistema operacional	
Hardware	

Organização básica de um ambiente computacional

Programas de Aplicação.

- Como usuários, interagimos com os programas de aplicação.
- Neste curso iremos descer nesta hierarquia, para construirmos novos programas de aplicação.
- Para construir novos programas podemos escrever diretamente códigos digitais que serão executados por um computador.
- Uma maneira mais simples é usar um compilador para uma linguagem de programação específica.



Organização básica de um ambiente computacional

Compiladores e Linguagens de Programação.

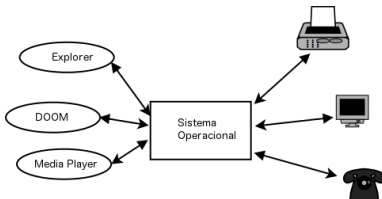
- Uma linguagem de programação é um conjunto de comandos que são mais “próximos” da linguagem humana do que os sinais digitais.
- Neste curso estamos interessados no estudo da *linguagem de programação C*.
- Um *compilador* é um programa que lê um código de uma linguagem de programação e o transforma em um programa executável.
- Na verdade o compilador realiza esta tarefa juntamente com um *assembler*.

```
for(i=0; i < 10; i++)      loop:  add c, a, b           0100 0010 0011 0101 0101
    c = a+b;                add i, i, 1          0110 0110 0111 0101 0101
                                bnq i, 10, loop        1111 0000 0111 0101 0101
```

Organização básica de um ambiente computacional

Sistema Operacional.

- Os programas possuem instruções que são executados no hardware.
- Mas o acesso ao hardware, como disco, memória, processador é controlado por um software especial conhecido como *sistema operacional*.
- Exemplos de sistema operacional: *Windows, Mac OS, Linux, Solaris*.



Algoritmos

Ao criarmos um programa para realizar uma determinada tarefa devemos ser capazes de construir algoritmos.

- Algoritmo: Seqüência de passos, precisos e bem definidos, para a realização de uma tarefa.
- Algoritmos podem ser especificados de várias formas, inclusive em português.

Exemplo de algoritmo

Como calcular 2345×4567 usando lápis, papel e uma tabuada?

Algoritmos

Resposta

- 1 Utilize o lapis e anote o número 2345 (n_1) no papel;
- 2 Utilize o lapis e anote o número 4567 (n_2) no papel sob o número 2345;
- 3 Para cada posição dos números faça:
- 4 some o dígito da posição atual de n_1 com o dígito da mesma posição de n_2 utilizando a tabuada;
- 5 anote o resultado da soma no papel utilizando lápis;

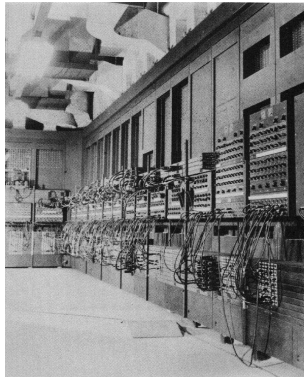
De algoritmos a programas

- Como transformar um algoritmo em linguagem que o computador entenda?
- Deve ser capaz de expressar tudo o que o computador pode fazer.
- Não pode ser ambígua.

Um pouco de história - Primórdios

Programação em código absoluto ou binário (apenas 0s e 1s).

ENIAC



Um pouco de história - Assembly

Uma melhoria: A Linguagem Assembly.

- Cria-se uma linguagem de baixo nível (Linguagem Assembly) para representar as instruções em código binário.
- Um programa, chamado montador ou assembler, faz a transformação em código absoluto.

```
LOOP:  MOV A, 3  
       INC A  
       JMP LOOP
```


Um Pouco de História - Linguagens de alto nível

- Mais distantes da máquina e mais próximas de linguagens naturais (inglês, português, etc.).
- Mesmo mais compreensíveis, elas não são ambíguas.
- Um **compilador** as transforma em código executável.

Exemplos de linguagens

- C
- Pascal
- Java

Primeiro programa em C

Um programa em C é um arquivo texto, contendo declarações e operações da linguagem. Isto é chamado de **código fonte**.

```
#include <stdio.h>

main() {
    printf("Hello, world!\n");
}
```

Como executar este programa

Para executar um programa a partir do seu código fonte é necessário compilá-lo, gerando **código binário** ou **executável**. Este pode ser executado como qualquer outro programa de aplicação.

```
$ gcc hello.c -o hello
$ hello
Hello, world!
```

O que são erros de compilação?

Caso o programa não esteja de acordo com as regras da linguagem, erros de compilação ocorrerão. Ler e entender estes erros é muito importante.

```
#include <stdio.h>
main() {
    printf("Hello, world!\n");
```

```
$ gcc hello.c -o hello
hello.c: In function 'main':
hello.c:5: error: syntax error at end of input
```

O que são erros de execução?

Acontecem quando o comportamento do programa diverge do esperado e podem acontecer mesmo quando o programa compila corretamente.

```
#include <stdio.h>
main() {
    printf("Hello, world! $#%#%#\n");
}
```

```
$ gcc hello.c -o hello
$ hello
Hello, world! $#%#%#
```

O que é um depurador?

- Ferramenta que executa um programa passo a passo.
- Ajuda a encontrar erros de execução (bugs).

Exemplo

- gdb

Um exemplo mais complexo

```
#include <stdio.h>
main() {
    int x, y;
    printf("x: ");
    scanf("%d", &x);
    printf("y: ");
    scanf("%d", &y);
    if (x > y)
        printf ("O maior número é x = %d\n", x);
    else
        printf ("O maior número é y = %d\n", y);
}
```

Resumo da Aula

- Hardware e Software.
- Pilha de um ambiente computacional: Programas Aplicações, Compilador, Sistema Operacional, Hardware.
- Código Binário, Assembly, Linguagem Alto Nível.
- Algoritmos.
- Erros de Compilação, Erros de execução e Depurador.

OBRIGADO!