

Programação Aplicada

Revisão de conceitos em linguagem C

Agostinho Brito

Departamento de Engenharia da Computação e Automação
Universidade Federal do Rio Grande do Norte

21 de abril de 2006

Revisão de conceitos em hardware/software

- Componentes de um computador e suas funções.
- Linguagens de programação.
- Como criar programas.
- Exemplo.

Componentes de um computador e suas funções

Computador: dispositivo que lê dados do mundo exterior, processa informações armazenadas em memória e apresenta resultados das tarefas realizadas.

Gabinete: Guarda as peças internas do computador.

Placa mãe: Interconecta as várias partes internas do computador e provê interfaces para meios de comunicação externos.

CPU: Processador central que comanda as operações da máquina.

ROM: Memória não volátil que armazena *software* básico para que a máquina consiga ligar e realizar a interpretação dos programas.

RAM: Memória volátil para uso geral dos programas em execução.

BIOS: Acorda o computador, conta memória, inicia dispositivos e dá início ao processo de boot.

Periféricos: teclado, mouse, impressora, monitor...

Disco rígido: Memória não volátil de maior capacidade de armazenamento.

Componentes de um computador e suas funções

Sistema Operacional: principal programa do sistema, que controla todos os recursos do computador (dispositivos físicos e funções de software).

- Cria abstração dos dispositivos e recursos do computador. Ex: Disco Rígido ↔ drive “c:”.
- Coordena recursos, controla interações de usuários, padroniza serviços
- (Multi/mono)programação, (Multi/mono)tarefa.
- Exemplos de sistemas operacionais: Linux, Windows, Solaris, MacOSX...

Componentes de um computador e suas funções

- Conjunto de códigos pré-determinados que permitem aos usuários especificar os passos devem ser seqüenciados para resolver um determinado problema (programa de computador).
- Baixo nível: hardware.
- Alto nível: usuário.
- Interpretação: código do programa é lido e suas instruções executadas de forma direta. Vantagem: rapidez de codificação/evolução.
Desvantagem: mais lento para executar
- Tradução: código do programa é convertido de linguagem de alto para baixo nível e, só então, executada. Vantagem: rapidez na execução.
Desvantagem: mais lento para evoluir a ferramenta.
- Algoritmo: conjunto de tarefas ou passos atômicos que leva à resolução de um problema. Implementados na forma de programas de computador.

Como criar programas

- Escrever o código em linguagem de programação de alto nível.
- Compilar o programa, gerando o executável.
- Executar o código gerado.
- Exemplo de programa compilado com o compilador **gcc**.

Listagem 1: helloworld.c

```
#include <stdio.h>

int main(void)
{
    printf("Hello, \world!\n");
    return 0;
}
```

```
$ gcc helloworld.c -o helloworld
$ ./helloworld
hello, world!
$
```