MC102 - Algoritmos e Progração de Computador

Prof. Alexandre Xavier Falcão

1° Aula: Introdução à Computação

1 Organização do computador

O computador é uma máquina que funciona mediante instruções enviadas pelo ser humano ou por outra máquina, executando tarefas e resolvendo problemas tais como: cálculos complexos, geração de relatórios, comando de outras máquinas (e.g. robô), controle de contas bancárias, comunicação de informações, etc. Uma visão simplificada do computador é ilustrada na Figura 1.

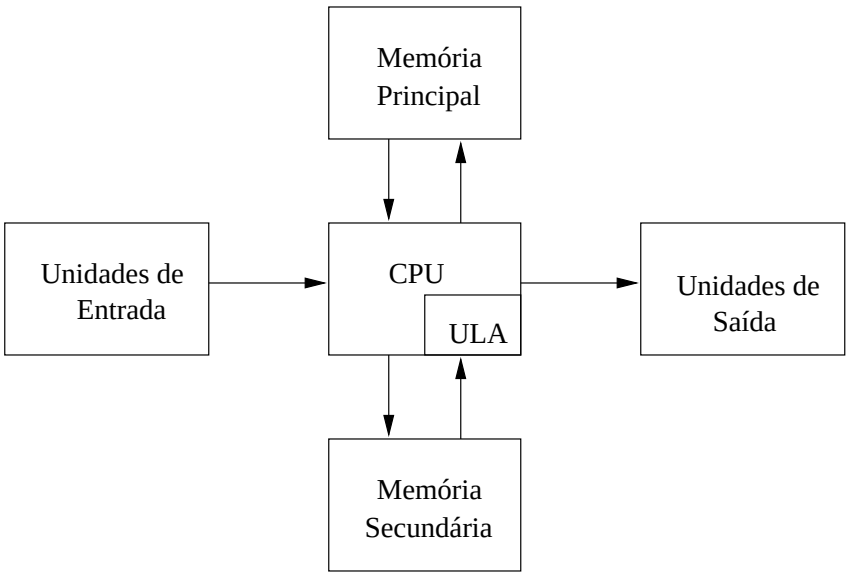

Figura 1: Organização básica de um computador

- **Unidades de Entrada:** Usadas pelo computador para receber instruções ou informações externas. Ex: teclado, *mouse*, câmera de vídeo, etc.
- **Unidades de Saída:** Usadas pelo computador para exibir os resultados da computação. Ex: monitor, impressora, etc.
- **Unidade Central de Processamento (***Central Processing Unit* **- CPU):** Responsável pelo gerenciamento do sistema como um todo, incluindo as memórias e as unidades de entrada e saída.

- **Unidade Lógica e Aritmética (ULA):** Responsável pelos cálculos matemáticos. Alguns computadores têm esta unidade separada da CPU. Também chamada de có-processador matemático.
- **Memória Principal:** Usada pela CPU para armazenar instruções e informações enquanto o computador está ligado. Também conhecida como memória RAM (*Random Access Memory*).
- **Memória Secundária:** Usada pelo computador para armazenar instruções e informações por prazo indeterminado, independente do estado do computador (ligado ou desligado). Em geral com capacidade de armazenamento bem maior do que a memória RAM, mas de acesso mais lento. Ex: discos rígidos (*winchester*), disquetes, fitas magnéticas, etc.

Observação: As memórias principal e secundária podem ser vistas como unidades de entrada e saída.

2 Alguns termos técnicos

1. **Dados:** Qualquer tipo de informação ou instrução que pode ser manipulada pelo computador. Ex: textos, imagens, etc.
2. **Bit:** Unidade básica para armazenamento, processamento e comunicação de dados.
3. **Byte:** Um conjunto de 8 bits.
4. **Palavra (*word*):** Um conjunto de n bytes (e.g. $n = 1, 2, 4, 8$). Os dados são organizados em palavras de n bytes. Os processadores mais modernos processam os dados em palavras de 16 bytes ou 128bits.
5. **Comandos:** São as instruções que fazem com que o computador execute tarefas.
6. **Programa:** É uma seqüência de instruções com alguma finalidade.
7. **Arquivo:** Conjunto de bytes que contém dados. Estes dados podem ser um programa, uma imagem, uma lista de nomes de pessoas, etc.
8. **Software:** É um conjunto de programas com um propósito global em comum.
9. **Hardware:** Consiste da parte física do computador.
10. **Sistema Operacional:** Conjunto de programas que gerenciam e alocam recursos de hardware e software. Ex: Unix, Windows98, OS2, MSDOS, etc.
11. **Linguagem de Programação:** Consiste da sintaxe (gramática) e semântica (significado) utilizada para escrever (ou codificar) um programa.
    (a) **Alto Nível:** Linguagem de codificação de programa independente do tipo de máquina e de fácil utilização pelo ser humano. Ex: Pascal, C, Algol, Cobol, Fortran (1° linguagem em meados de 1950), BASIC, Java, Python, Tcl/Tk, etc.
    (b) **Baixo Nível:** Linguagem de codificação baseada em mnemônicos. Dependente do tipo de máquina e de fácil tradução para a máquina. Conhecida como linguagem assembly.

12. **Linguagem de Máquina:** Conjunto de códigos binários que são compreendidos pela CPU de um dado computador. Dependente do tipo de máquina.

13. **Compilador:** Traduz programas codificados em linguagem de alto ou baixo nível (i.e. código fonte) para linguagem de máquina (i.e. código executável). Ex: O assembler transforma um programa em assembly para linguagem de máquina. Uma vez compilado, o programa pode ser executado em qualquer máquina com o mesmo sistema operacional para o qual o programa foi compilado.

14. **Interpretador:** Traduz o código fonte para código de máquina diretamente em tempo de execução. Exemplos de linguagens interpretadas são BASIC, Python, Tcl/Tk e LISP.

15. **Algoritmos:** São procedimentos ou instruções escritos em linguagem humana antes de serem codificados usando uma linguagem de programação. Uma receita de bolo é um bom exemplo da organização de um algoritmo.

3 Objetivos do curso

A Figura 2 mostra um diagrama das tarefas básicas para a solução de problemas usando um computador. O objetivo principal deste curso é exercitar estas tarefas definindo vários conceitos de computação e usando a linguagem C como ferramenta de programação.

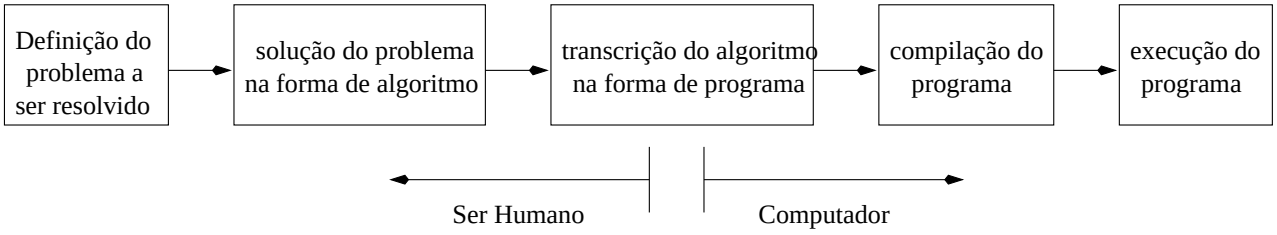

Figura 2: Etapas da resolução de problemas usando um computador.

4 Jogo das gavetas

Essencialmente, programar um computador para executar uma dada tarefa é estabelecer regras de manipulação de informações na sua memória principal através de uma seqüência de comandos. A memória principal funciona como um armário de gavetas, cuja configuração varia de programa para programa. Cada programa estabelece o número de gavetas e as gavetas possuem nome, endereço e capacidade de armazenamento diferentes. As gavetas podem armazenar números inteiros, números reais, e caracteres, os quais requerem número de bytes diferentes. O conteúdo das gavetas pode ser lido ou modificado utilizando seu nome ou endereço.

Suponha, por exemplo, que gostaríamos que o computador calculasse a soma de dois números inteiros. Assim como o ser humano, os números são armazenados na memória, são somados, e depois o resultado é armazenado na memória. Para armazenar os números na memória, precisamos estabelecer nome, endereço e capacidade de armazenamento de cada gaveta compatível com um número inteiro. Depois atribuímos os valores para cada gaveta. Como os números não serão mais necessários após a soma, nós podemos usar uma das gavetas para armazenar o resultado. Pedimos

então que o computador execute a soma e armazene o resultado em uma das gavetas. Esses comandos podem ser traduzidos na forma de um algoritmo.

1. Considere as gavetas a e b.
2. Atribua 20 para a e 30 para b.
3. Some a com b e coloque o resultado em a.

Mais especificamente, as gavetas são chamadas **variáveis** e os algoritmos são escritos de uma forma mais elegante.

1. Sejam a e b variáveis inteiras.
2. Faça $a \leftarrow 20$ e $b \leftarrow 30$.
3. Faça $a \leftarrow a + b$.

Este algoritmo pode então ser transcrito na liguagem C da seguinte forma.

```
/* função principal. As funções e alguns comandos têm o escopo
definido entre parênteses. */
int main() {
int a,b;
  a = 20; b = 30;
  a = a + b;
  printf("%d\n",a); /* imprime na tela o resultado. */
}
```

O programa pode ser editado e gravado em um arquivo exemplo.c. O arquivo pode ser compilado usando o comando "gcc exemplo.c -o exemplo" e o arquivo executável "exemplo" será gerado.

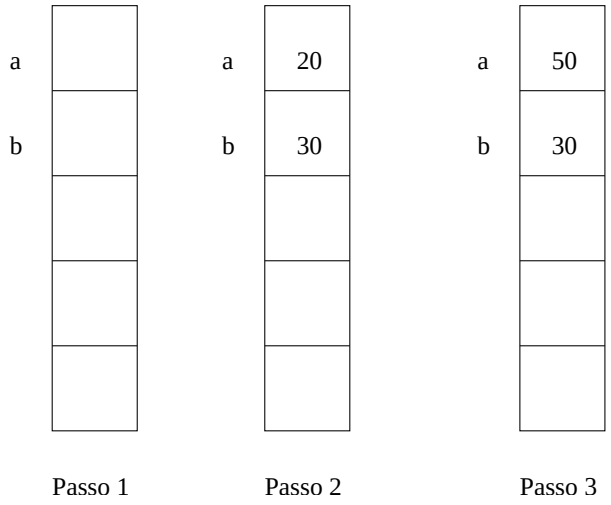

Figura 3: Exemplo do jogo das gavetas.