

Introdução à Arquitetura e Organização de Computadores

Professor: Diego de Assis Monteiro Fernandes

Aula 2 - Introdução (Abstrações e Tecnologia)

1 Motivação

Os programadores têm uma constante preocupação com o desempenho de seus programas, procurando sempre desenvolvê-los de modo a retornar os resultados ao usuário rapidamente. Em 1960 e 70 a maior restrição com relação ao desempenho era a quantidade de memória. Isso gerava a seguinte meta de programação: “*minimizar o espaço em memória para tornar os programas rápidos*”. Na última década, os avanços no projeto de computadores e memórias superou tais problemas. Com isso, os programadores passaram a preocupar-se com outras questões, como: a hierarquia das memórias e a natureza paralela dos processadores. Tais questões possuem uma grande influência no projeto e criação de um programa, demandando assim um maior conhecimento da organização interna dos computadores. Esse conhecimento permite a compreensão de muitos dos segredos de programar um computador em sua linguagem nativa e a avaliação do desempenho do seus programas.

2 Analisando em baixo nível

Computadores trabalham manipulando e armazenando dados. Todos os dados, em um baixo nível, são modelados como um conjunto de *bits* (*binary digit*), onde um *bit* suporta somente dois tipos de valores: **0** e **1**. Tais dados são manipulados através de comandos, que busca-os e executam uma operação específica. Cada comando é denominado **instrução**. Em baixo nível, uma instrução é apenas uma coleção de *bits* que são compreendidas pelo computador. Por exemplo, os *bits* **1000110010100000** podem representar uma instrução de soma para dois números. Em resumo, dados e instruções são representadas como um conjunto de *bits* para o computador.

Os primeiros programadores não dispunham de linguagens de programação como C++, e a construção de programas era realizada através da composição de vários números binários. Entretanto, essa tarefa era extremamente tediosa, o que demandou uma notação mais próxima do modo de pensar do ser humano. A partir daí, foram criados programas para traduzir essa notação simbólica, que era uma linguagem de programação de baixo nível, para o binário. Esse programas são chamados **montadores** (assembler) e traduzem a versão simbólica de uma instrução em sua versão binária. Por exemplo:

```
add A, B
```

seria traduzido para:

```
1000110010100000
```

Essa linguagem simbólica é chamada **linguagem de montagem**. Embora tenha trazido um grande melhoramento, a linguagem de montagem ainda é muito complexa, pois exige que o programador racione de modo muito próximo ao funcionamento da máquina. Com isso, programadores desenvolveram programas, os **compiladores**, para traduzir uma linguagem mais natural (**linguagens de alto nível**) para a linguagem de montagem. Com tais programas, pode-se elaborar uma série de traduções onde o último código obtido é o código binário compreendido pelo computador. Por exemplo:

A+B

(compilador) seria compilado para

add A, B

(montador) que então seria traduzido para

1000110010100000

As linguagens de alto nível proporcionam diversos benefícios:

- Facilitam a programação, pois a linguagem é mais próxima do nossa forma de raciocínio.
- Aumentam a produtividade de programação, reduzindo o tempo de desenvolvimento de uma aplicação.
- Permitem que programas sejam independentes do computador em que é desenvolvido, uma vez que o compilador e o montador podem traduzir os programas na linguagem de alto nível para o binário de qualquer máquina.

2.1 Início dos sistemas operacionais

Muitos trechos de programas, que tratavam por exemplo da entrada e saída de dados, por serem sempre necessários, começaram a ser reutilizados. Com o passar do tempo, foi se tornando cada vez mais evidente que esses trechos de programa deveriam ser executados como um programa independente que tivesse como tarefa principal supervisionar os outros programas e fornecer serviços comumente requisitados. Tal programa é a base para o que é conhecido hoje como **sistema operacional**.

3 Detalhando um computador

Um computador é composto de uma série de dispositivos. Alguns deles tem como objetivo principal realizar a interação entre o homem e o máquina. Como exemplo, pode-se incluir teclado, mouse e monitor, dispositivos que funcionam como **entrada e saída** de dados. Esses dispositivos são chamados **dispositivos de entrada e saída**.

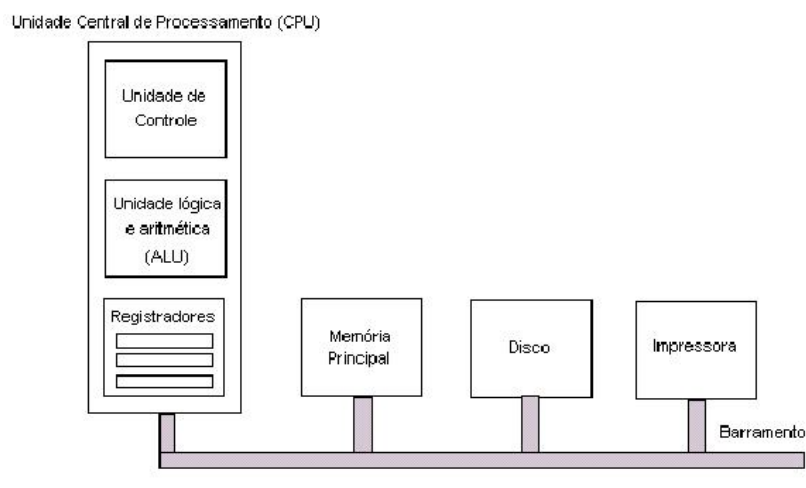


Figura 1: Estrutura básica de um computador.

Além dos dispositivos de entradas e saída, outros componentes também são evidentes. Um deles é a placa mãe que é composta basicamente de três partes, onde são conectados os dispositivos de E/S, a memória e o processador. A **memória** armazena os programas que estão sendo executados e, além disso, também contém os dados necessários para a execução de tais programas. O **processador** é a parte ativa do computador, executando diretamente as instruções de um programa. Ele executa operações simples como a adição e teste de números, a ativação de dispositivos de E/S, etc...

Se um nível com mais detalhes for analisado, é possível observar que um processador é composto de dois componentes principais: **a unidade de controle e a unidade lógica e aritmética**. A unidade lógica e aritmética (ULA) realiza as operações propriamente ditas e a unidade de controle (UC) informa à unidade lógica e aritmética, memória e dispositivos de E/S o que desempenhar de acordo com as instruções do programa.

Analisando a memória em mais detalhes, como no caso do processador, pode-se notar dois tipos de memória: RAM e cache. Na memória RAM são armazenados os dados e o programa em execução. A memória cache consiste de uma memória pequena e rápida que atua como um *buffer* da RAM e tem como objetivo diminuir o tempo de acesso aos dados pelo processador. Para aprimorar esse conceito, os computadores são dotados de dois níveis de cache L1 e L2, onde L1 é o nível mais próximo ao processador. A memória RAM do computador é volátil, ou seja, os dados armazenados são perdidos quando o computador é desligado. Por isso, além do computador possuir esse tipo de memória, denominada **memória primária ou principal**, outro tipo de memória, a **memória secundária**, também é utilizada. Atualmente, existem vários dispositivos que funcionam como memória secundária: Disco rígido, Fita magnética, Floppy, Zip, CD, DVD, etc...

Três principais fatores podem ser adotados para apontar as diferenças entre as memórias de um computador:

- volatilidade;
- custo;
- tempo de acesso.

Para conectar o processador, a memória e os dispositivos de entrada e saída, são utilizados os meios de interconexão que estabelecem a comunicação entre os componentes através de **barramentos** sob seu controle.

3.1 Definição de arquitetura de um computador

É possível notar que os detalhes de um computador podem ser escondidos de modo a oferecer um modelo mais simples ao usuário ou ao programador. Essa técnica, a **abstração**, é fundamental no projeto de computadores. Uma das abstrações mais importantes é a interface entre o hardware e o software de mais baixo nível. Essa abstração é composta pelo **conjunto de instruções da arquitetura**, ou simplesmente **arquitetura de uma máquina**. *O conjunto de instruções da arquitetura inclui tudo o que os programadores necessitam saber para fazer com que uma linguagem de máquina binária funcione corretamente, incluindo instruções, dispositivos de E/S, etc.* Essa interface padronizada permite o desenvolvimento de programas sem o conhecimento específico do hardware.

Outra definição de arquitetura é dada a seguir:

A arquitetura de um computador é sua estrutura e comportamento como vistos por um programador de linguagem de máquina, e envolve o seguinte: formato de instruções, modos de endereçamento, conjunto de instruções e a organização geral do processador, memória e E/S.

Os aspectos básicos fundamentais em uma arquitetura são:

1. Os tipos de instruções disponíveis;
2. Os tipos de dados utilizados;
3. Os mecanismos disponíveis para alterar o fluxo de controle;
4. A organização e endereçamento da memória;

5. A relação entre o conjunto de instruções e a organização da memória;
6. O método pelo qual é implementado.

3.2 Classificação básica

Como já mencionado em um alto nível de abstração, um computador é geralmente organizado em processador, memória, dispositivos de E/S e barramento. Entretanto, alguns casos constituem uma variação ou exceção ao que foi mencionado. Com base no relacionamento e organização desses componentes de um sistema computacional é possível elaborar uma classificação. Essa classificação divide os sistemas computacionais em duas classes: máquinas von Neumann e máquinas não-von Neumann.

3.2.1 Máquinas von Neumann

A característica de máquinas von Neumann é a composição do sistema a partir de três subsistemas básicos: CPU, memória principal e sistema de entrada e saída. A CPU (unidade central de processamento), por sua vez, tem três blocos principais: **unidade de controle (UC)**, **unidade lógico-aritmética (ULA)** e **registradores**, incluindo-se aí um registrador contador de programa (PC) que indica a posição na memória da instrução a executar. Outro registrador importante é o IR que é o registrador que contém a instrução corrente em execução. São características das máquinas von Neumann a utilização do conceito de programa armazenado, a execução seqüencial de instruções e a existência de um caminho único entre memória e unidade de controle.

Durante sua operação, a execução de um programa é uma seqüência de ciclos de máquina von Neumann, compostos por:

1. Busca da instrução: transfere instrução da posição de memória apontada por PC para a CPU
2. Execução da instrução: a unidade de controle decodifica a instrução, armazenada em IR, e gerencia os passos para sua execução pela ULA

Uma variante do modelo básico de máquinas von Neumann é denominado de máquinas Harvard, onde há vias separadas para dados e instruções entre memória principal e CPU. A origem do termo vem dos computadores Mark I a Mark IV, desenvolvidos em Harvard, com memórias de dados e instruções separadas.

3.2.2 Máquinas não-von Neumann

As máquinas que não se enquadram na definição de máquinas von Neumann são denominadas máquinas não-von Neumann. Essa categoria é ampla, incluindo sistemas computacionais tais como as máquinas paralelas, onde várias unidades de processamento executam programas de forma cooperativa.

4 Linguagens, níveis e máquinas virtuais

Esse trecho da aula tem como objetivo detalhar um pouco do que foi descrito anteriormente sobre a interação entre as várias linguagens utilizadas em um computador.

Um programa é uma seqüência de instruções que descreve como executar uma determinada tarefa. Cada computador é capaz de executar um conjunto limitado de instruções simples (por exemplo, mover um dado na memória) e esse conjunto de instruções primitivas é denominada linguagem de máquina (L1). Como já foi dito, é extremamente tedioso construir uma aplicação utilizando essa linguagem.

A solução imediata é construir então uma nova linguagem (L2) mais conveniente. Entretanto, o computador ainda só é capaz de executar as instruções de L1. Existem duas formas distintas para realizar a interação entre tal linguagem e o computador, ou seja fazer com o computador execute L2:

- **Tradução:** substitui cada instrução de um programa escrito por L2 por uma seqüência equivalente de instruções em L1, gerando todo o programa na linguagem L1.

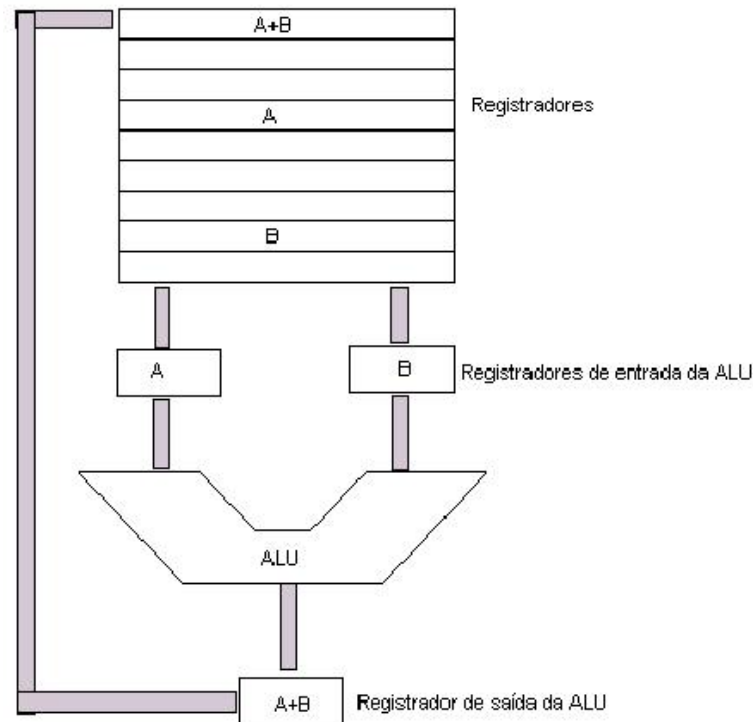


Figura 2: Fluxo de dados de uma máquina de von Neumann.

- **Interpretação:** examina cada instrução durante a execução de um programa em L2 e executa para cada instrução o conjunto de instruções equivalentes em L1.

Se L2 ainda não for adequada para se construir programas, pode-se criar varias outras linguagens mais apropriadas: L3, L4, etc. Cada linguagem, nesse caso, interagem com a linguagem imediatamente inferior a ela. Esse conjunto de linguagens resultam então na formação de uma hierarquia, constituindo um computador de conjunto de camadas. Uma hierarquia possível é descrita a seguir.

- **Nível de lógica digital:** constituído pelos circuitos formados pelas portas lógicas (AND, OR, NOT,...)
- **Nível de microprogramação:** cada computador possui um nível de microprogramação diferente, onde sua função é a de executar o microprograma..
- **Nível de máquina convencional:** descreve de modo mais abstrato as instruções. Esse nível é bem semelhante entre máquinas distintas. Além disso, alguns computadores não possuem o nível de microprogramação e, nesse caso, as instruções do nível de máquina convencional são executadas diretamente pelos circuitos digitais.
- **Nível de sistema operacional:** é um nível híbrido, é composto pelas instruções do nível 2 mais algumas instruções que são disponibilizadas com o intuito de simplificar ao programador algumas tarefas (ver a seção no início da aula que descreve sucintamente o início dos sistemas operacionais). Esse nível pode alterar a organização do computador para os níveis superiores, incluindo mecanismos mais sofisticados. Destacam-se a organização diferente de memória (memória virtual) e a execução de vários processos de modo concorrente (escalonamento de processos).

A partir do nível 4 as linguagens são utilizadas por programadores para a implementação de aplicações em geral e, por isso, as instruções são descritas através de palavras ou abreviações. Ao contrário disso as linguagens dos níveis 1,2 e 3 são numéricas e são sempre interpretadas.

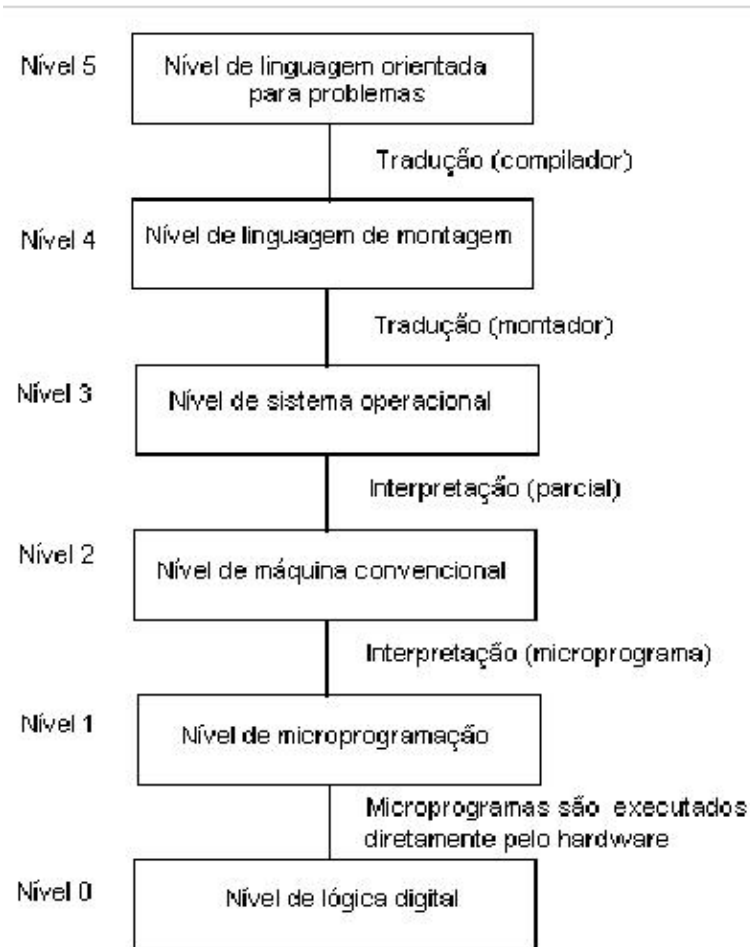


Figura 3: Níveis de um computador.

- Nível de linguagem de montagem: provê um método de escrita de programas para os níveis 1,2 e 3. Lembrando que o programa fundamental desse nível é o montador (assembler).
- Nível de linguagem orientada para problemas: são linguagens de alto nível como BASIC, C, COBOL, FORTRAN, LISP, Pascal. Em geral programas escritos por essas linguagens são traduzidas para o nível 3 ou 4 por compiladores. Em alguns casos também podem ser interpretadas.

Existem, além dos níveis descritos, níveis superiores que são projetados para a criação de instruções específicas para certas aplicações. Analisando essa hierarquia de níveis, pode-se perceber que é possível escrever programas sem conhecer detalhadamente os níveis de base, ou seja, os detalhes do hardware.