

**CENTRO UNIVERSITÁRIO
INSTITUTO DE EDUCAÇÃO SUPERIOR DE BRASÍLIA – IESB**



Sistemas Operacionais

ELIAS FILHO

Aula 1

1. Visão Geral

1.1 – Introdução

1.2 – Funções Básicas

1.3 – Máquina de Níveis

1.4 – Histórico

1.5 – Tipos de Sistemas Operacionais

1.6 – Código de Máquina

2. Conceitos de Hardware e Software

2.1 – Introdução

2.2 – Hardware

2.3 – Software

1.1 Introdução

- **Sistema Operacional** ⇒ conjunto de rotinas executado pelo processador.
- **Principal função do SO** ⇒ controlar o funcionamento de um computador, gerenciando a utilização e o compartilhamento dos seus diversos recursos.
- É possível utilizar um computador sem um sistema operacional?
- **A grande diferença** entre um sistema operacional e aplicações convencionais é a maneira como suas rotinas são executadas em função do tempo.
- **Outros nomes** ⇒ monitor, executivo, supervisor, controlador.

1.2 Funções Básicas

- **Facilidade de acesso aos recursos do sistema**

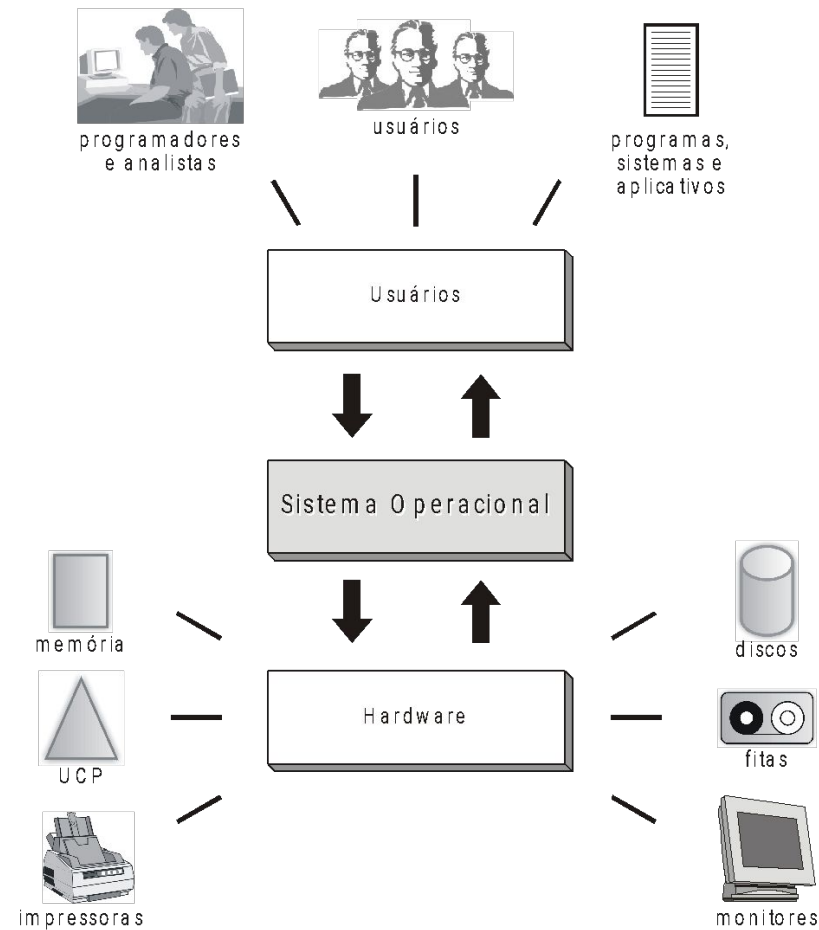
- O **usuário** não precisa saber como funciona o computador.
- **Máquina virtual** ⇨ o SO serve de interface entre os usuários e os recursos disponíveis no sistema computacional.
- **Utilitários do SO** ⇨ compiladores, linkers, bibliotecas, depuradores, etc.

- **Compartilhamento de recursos de forma organizada e protegida**

- Diminuição dos **custos**
- Execução de diversas tarefas ao **mesmo tempo**

1.2 Funções Básicas

Visão do sistema operacional

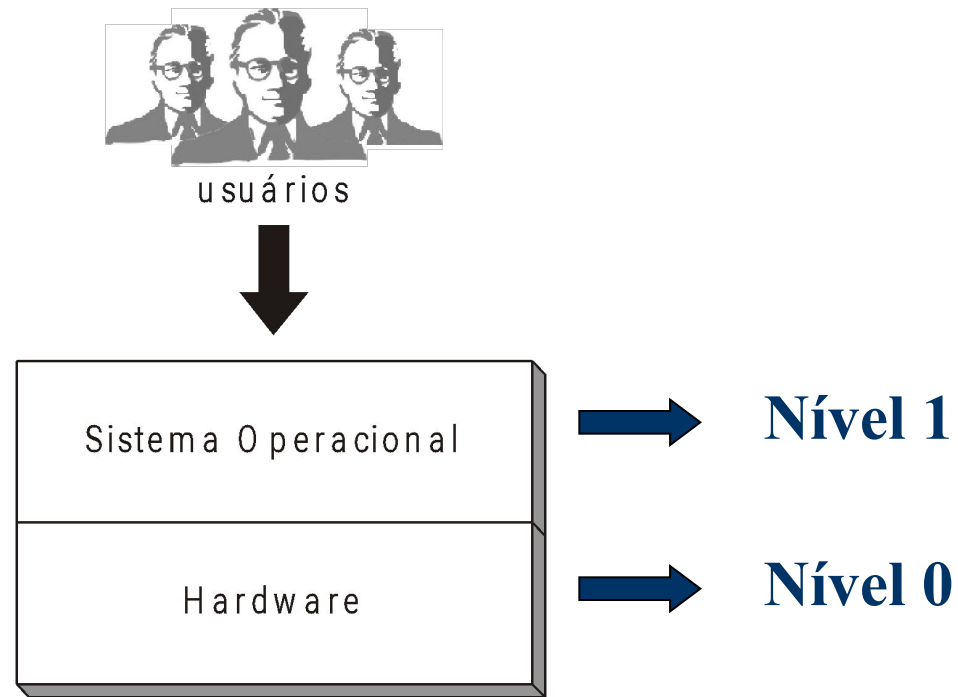


1.3 Máquina de Níveis

- **Hardware** ⇒ circuitos eletrônicos, placas, cabos e fontes de alimentação.
- **Software** ⇒ programas que oferecem serviços aos usuários.
- Uma operação efetuada pelo software pode ser **implementada** em hardware, enquanto uma instrução executada pelo hardware pode ser **simulada** via software.
- O SO faz a interface entre o hardware e software. Não há necessidade do programador se envolver com a **complexidade** do hardware.
- O computador pode ser visualizado como uma máquina de dois **níveis** ou **camadas**: hardware (nível 0) e sistema operacional (nível 1).

1.3 Máquina de Níveis

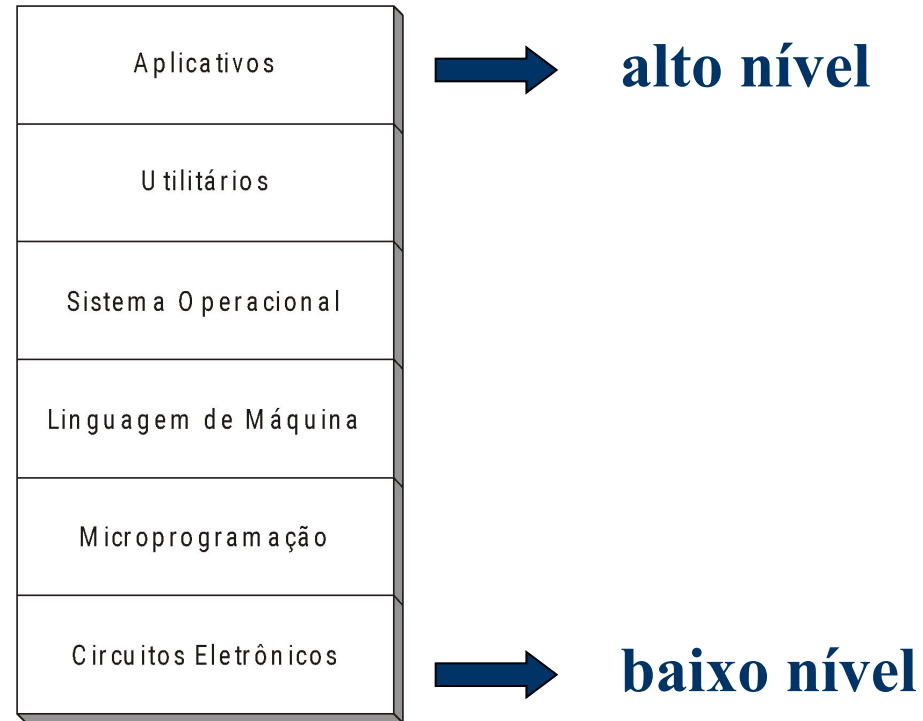
Visão do usuário



Esta visão modular e abstrata é chamada máquina virtual

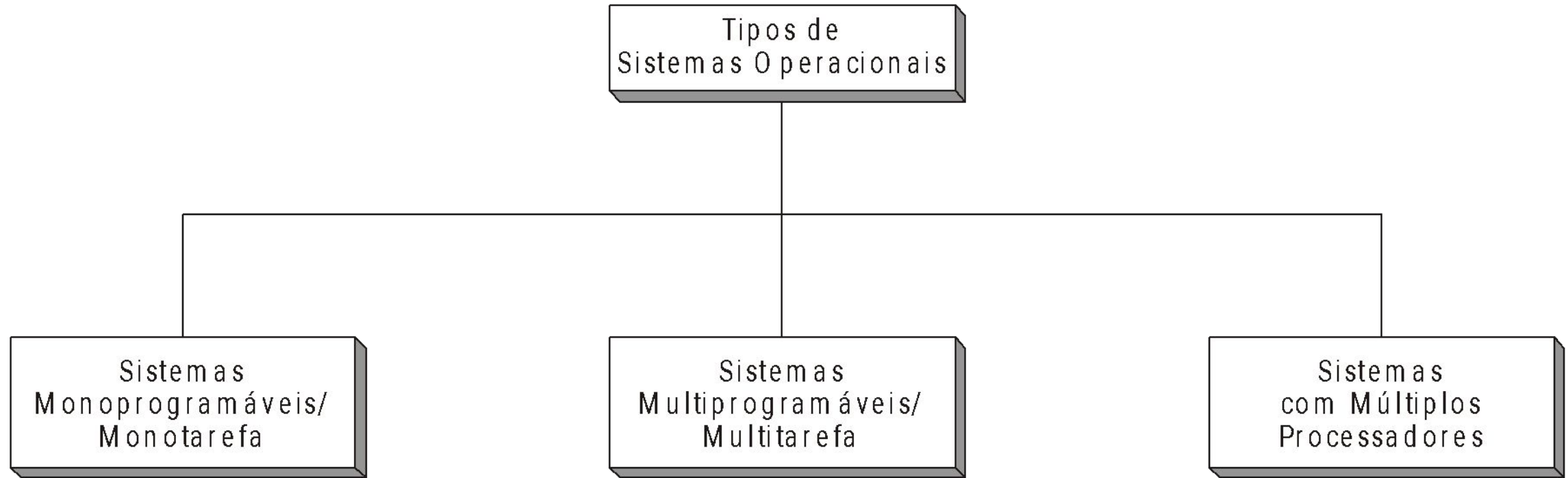
1.3 Máquina de Níveis

Na realidade, o computador possui vários níveis. A linguagem utilizada em cada um desses níveis é diferente.



Exemplo de uma estrutura de níveis.

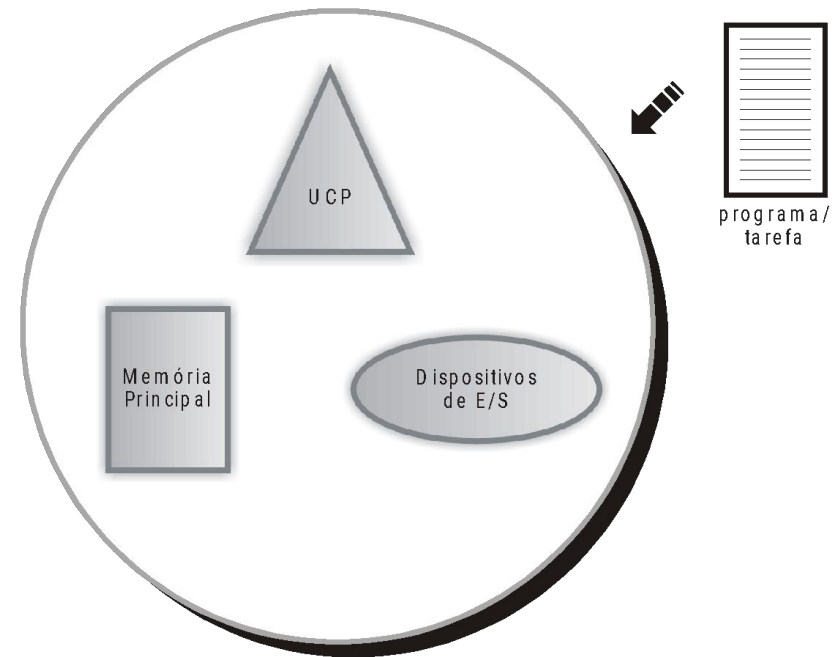
1.5 Tipos de Sistemas Operacionais



1.5 Tipos de Sistemas Operacionais

SOs monoprogramáveis/monotarefa

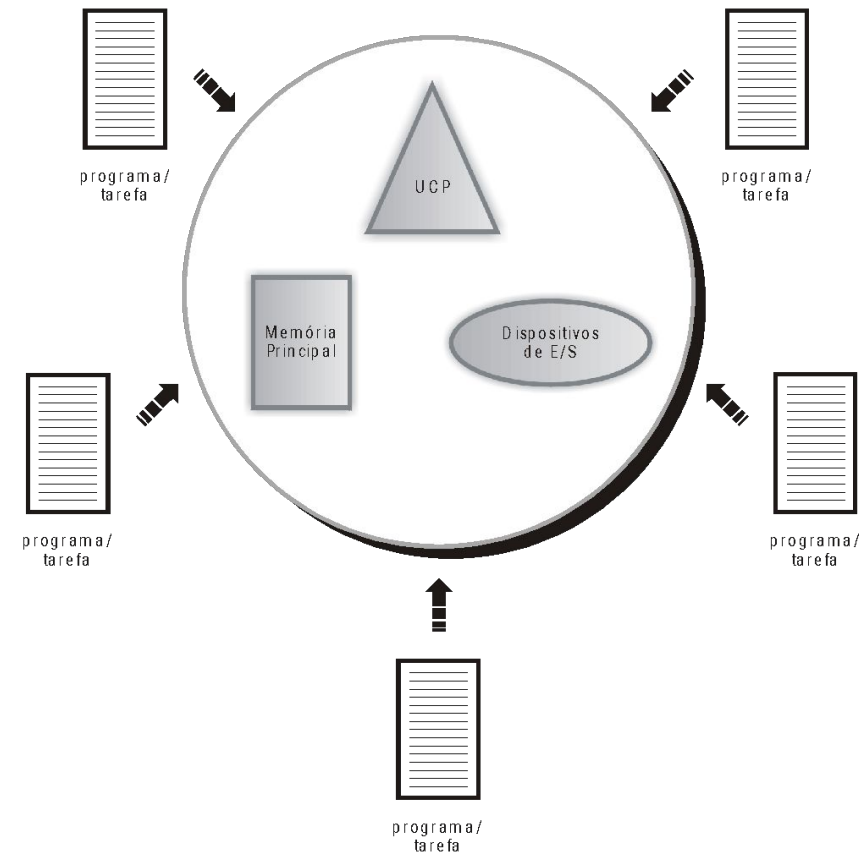
- Todos os recursos do sistema ficam dedicados a uma única tarefa.
- Desperdício de recursos.
- Implementação simples.



1.5 Tipos de Sistemas Operacionais

SOs multiprogramáveis/multitarefa

- Recursos são compartilhados entre os diversos usuários e aplicações.
- Redução do tempo de resposta das aplicações processadas.
- Pode ser monousuário ou multiusuário.



1.5 Tipos de Sistemas Operacionais

SOs multiprogramáveis /multitarefa



Classificados de acordo como as aplicações são gerenciadas

1.5 Tipos de Sistemas Operacionais

Sistemas batch

- Não exige interação do usuário; tempo de resposta pode ser longo; atualmente, não existe sistemas exclusivamente dedicados a este tipo de processamento.

Sistemas de tempo compartilhado

- Cada programa recebe uma fatia de tempo para executar; permitem a interação dos usuários; conhecidos como sistemas on-line; a maioria das aplicações comerciais são processadas em sistemas de tempo compartilhado.

Sistemas de tempo real

- O tempo de processamento deve estar dentro de limites rígidos; um programa utiliza o processador o tempo que for necessário ou até que apareça outro mais prioritário; usados em aplicações de controle de processos (tráfego aéreo, usinas nucleares, etc).

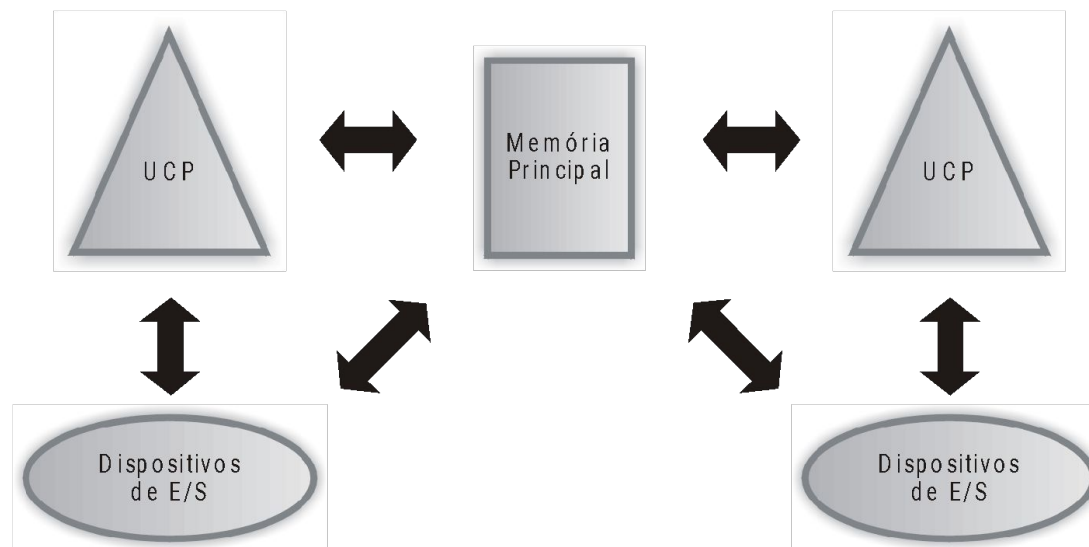
1.5 Tipos de Sistemas Operacionais

Sistemas com múltiplos processadores

- Permitem que vários programas sejam executados ao mesmo tempo ou que um mesmo programa seja subdividido em partes para serem executadas simultaneamente em mais de um processador.
- **Escalabilidade** ⇨ capacidade de ampliar o poder computacional do sistema apenas adicionando novos processadores.
- **Disponibilidade** ⇨ capacidade de manter o sistema em operação mesmo em casos de falhas.
- **Balanceamento de carga** ⇨ possibilidade de distribuir o processamento entre os diversos processadores.
- **Formas de comunicação entre as CPUs** ⇨ sistemas fortemente acoplados e sistemas fracamente acoplados.

Sistemas com Múltiplos Processadores

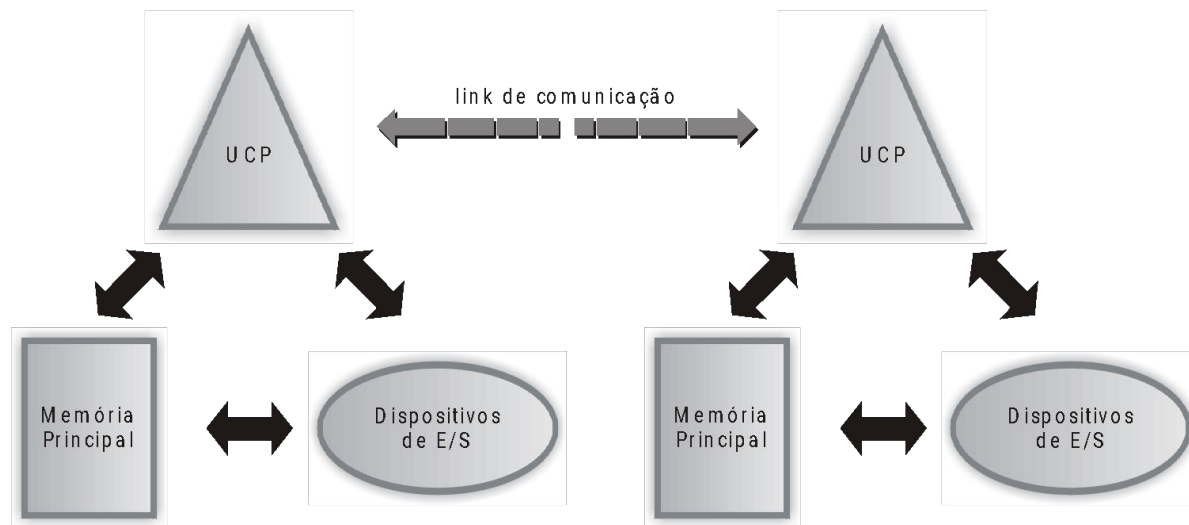
Sistemas fortemente acoplados ou multiprocessadores



- São gerenciados por um único SO.
- Os **sistemas SMP** (Symmetric Multiprocessors) \Rightarrow os processadores possuem tempo uniforme de acesso à memória principal.
- Os **sistemas NUMA** (Non-Uniform Memory Access) \Rightarrow acesso a memória não apresenta tempo uniforme.

Sistemas com Múltiplos Processadores

Sistemas fracamente acoplados



- Dois ou mais sistemas computacionais conectados através de linhas de comunicação. Cada sistema funciona de forma **independente**.
- Classificação: sistemas operacionais de rede e sistemas distribuídos.
- **SOR** ⇒ os usuários têm conhecimento dos hosts e seus serviços.
- **SD** ⇒ o SO esconde os detalhes dos hosts individuais e passa a tratá-lo como um conjunto único. Ex: cluster.

1.6 Código de Máquina

Sequência da instruções executadas pelo computador

Linguagem de Alto Nível

```
let a = 7;  
let b = 2;  
let c = a+b  
if (c != 0){  
    c++  
}
```

Linguagem de Montagem - assembly

```
guarda 7 em a  
guarda 2 em b  
pega a  
soma b  
guarda em c  
compara c e 0  
se_igual_pula 1 linha  
incrementa c
```

1.6 Código de Máquina

Bits e bytes

- O que é o bit é byte?
- **Bit e Byte** são as menores unidades usadas para medir armazenamento e transmissão de dados em computação, embora a quantidade de **bits e bytes** em um GB possa variar, dependendo do sistema de unidades.
- Em um computador, cada um desses 0s e 1s é chamado de **bit**, que vem do inglês **B**inary Digit**** ou dígito binário. Essa é a menor unidade possível para o armazenamento de informação.
- Como bit é uma unidade muito pequena, costuma-se trabalhar com grupos de **8 bits**, essa quantidade agrupada de bits é chamada de **byte**. Por convenção, quando medimos o armazenamento em um computador usamos o byte.
- Para representar essas unidades, utilizamos **b** (“b” minúsculo) para o bit e **B** (“B” maiúsculo) para byte. Ou seja, $1B = 8b$.
- Você pode encontrar essas unidades em conjunto com prefixos SI como k(kilo, ou mil unidades), M(mega, ou 1 milhão de unidades), G (giga, ou 1 bilhão de unidades) e assim por diante. Por exemplo:
- Um arquivo de **200 MB** é um arquivo com **200 milhões de bytes**.
- Uma internet com velocidade de **20 Mbps** transfere **20 milhões de bits por segundo** ou **2,5 milhões de bytes por segundo**.

2. Conceitos de Hardware e Software

2.1 – Introdução

2.2 – Hardware

Processador; memória principal; memória cache; memória secundária; dispositivos de E/S; barramento; pipelining; arquiteturas RISC e CISC; análise de desempenho

2.3 – Software

Tradutor; interpretador; linker; loader; depurador; interpretador de comandos e linguagem de controle; ativação/desativação do sistema

2.1 Introdução

- **Hardware** ⇨ é a parte física do sistema computacional.
- **Software** ⇨ é o programa ou instruções que diz ao computador o que fazer.

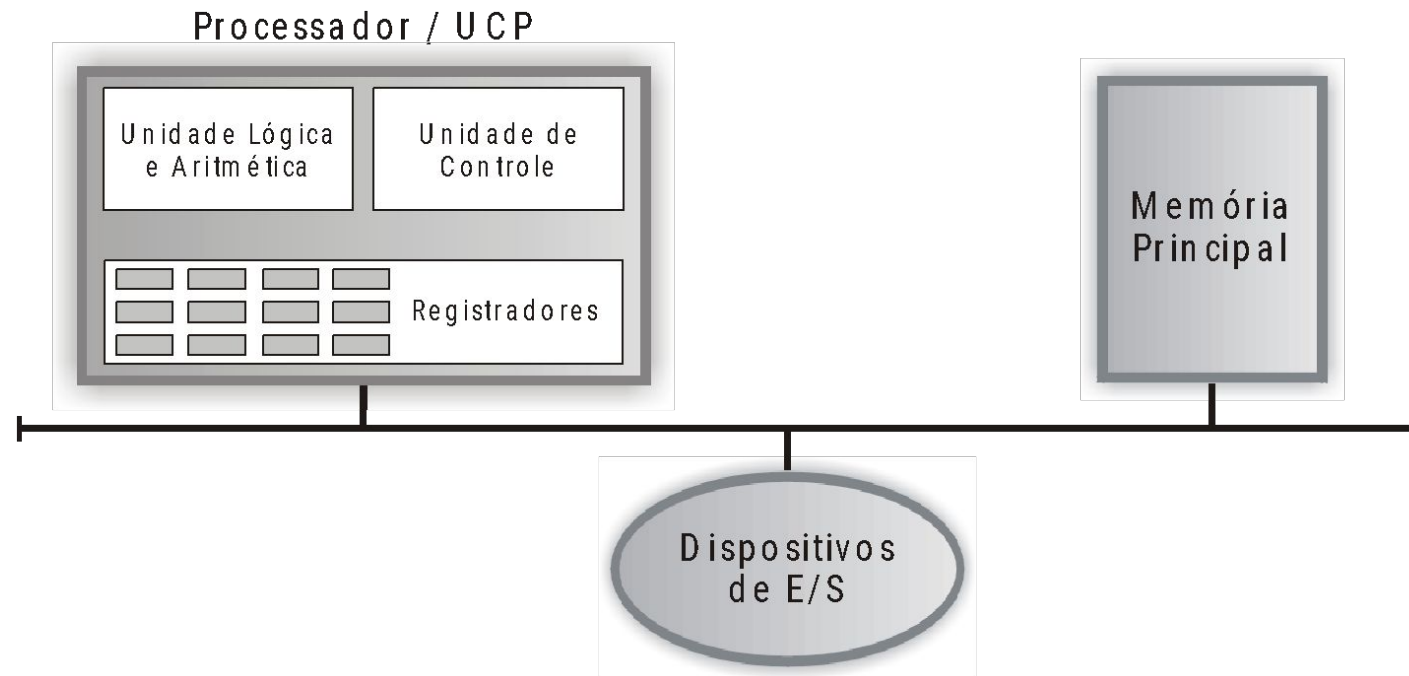
“Hardware é aquilo que você chuta, software é aquilo que você xinga”.

João Ubaldo Ribeiro, escritor

2.2 Hardware

Os componentes de um sistema computacional são agrupados em três unidades funcionais:

1. processador ou unidade central de processamento;
2. memória principal;
3. Dispositivos de entrada e saída.



2.2 Hardware

Processador ou Unidade Central de Processamento (UCP)

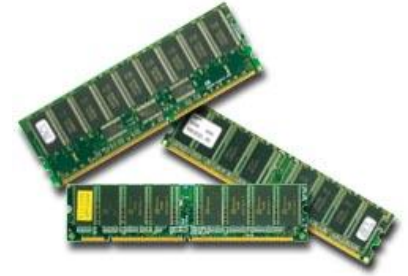
- Principal função ⇨ controlar e executar instruções presentes na memória principal.
- Componentes:
 - **Unidade de Controle (UC)** ⇨ gerenciar as atividades
 - **Unidade Lógica e Aritmética (ULA)** ⇨ realização de operações lógicas
 - **Clock** ⇨ sincronização das funções do processador
 - **Registradores**
 - **CI** ou **PC** ⇨ contém o endereço da próxima instrução
 - **AP** ou **SP** ⇨ contém o endereço de memória do topo da pilha
 - **PSW** (registrador de status) ⇨ informações sobre a execução de instruções



2.2 Hardware

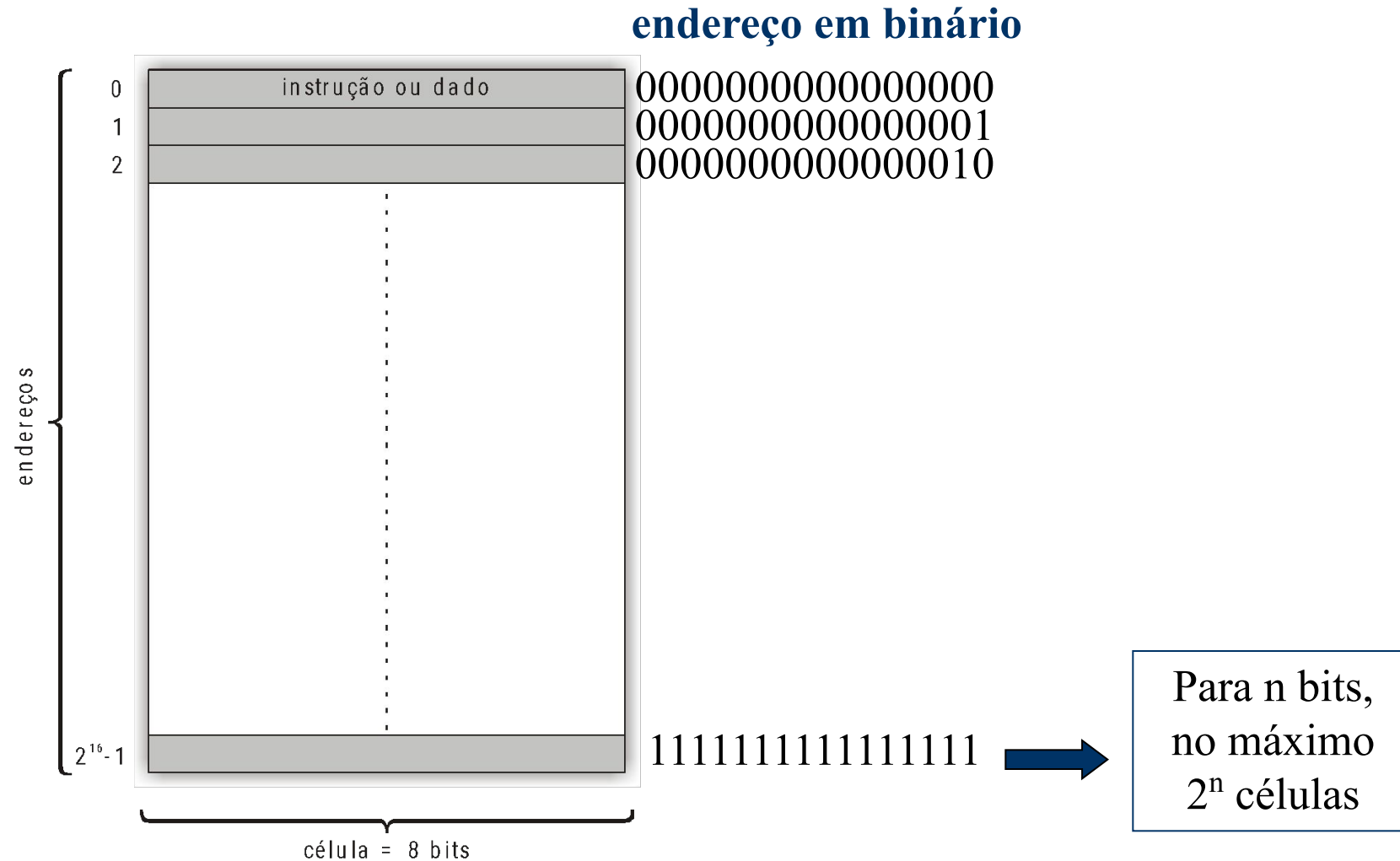
Memória Principal, Primária ou Real

- É o local onde são armazenados instruções e dados
- É composta por **células**
- Cada célula é composta por um determinado número de bits.
- **Bit**: zero ou 1
- A maioria dos computadores utiliza o byte (8 bits) como tamanho da célula
- Endereço ⇨ referência única a uma célula de memória.
- **Volátil** ⇨ RAM (Random Access Memory)
- **Não-Volátil** ⇨ ROM (Read-Only Memory)
 - Bios (<https://youtu.be/3xysT32ktlQ>)
 - Boot(<https://youtu.be/yVXzi3xhGp8>)



2.2 Hardware

Memória principal com 64 Kbytes ($= 2^6 \times 2^{10}$ bytes $= 2^{16}$ bytes)



2.2 Hardware

Ciclo de leitura e gravação

Operação de Leitura	Operação de Gravação
1. A UCP armazena no MAR o endereço da célula a ser lida	1. A UCP armazena no MAR o endereço da célula que será gravada
2. A UCP gera um sinal de controle para a memória principal	2. A UCP armazena no MBR a informação a ser gravada
3. O conteúdo de MAR é transferido para o MBR	3. A UCP gera um sinal de controle para a memória principal
4. O conteúdo do MBR é transferido para um registrador da UCP	4. O conteúdo do MBR é transferido para a célula endereçada pelo MAR

- **MAR** (memory address register) ⇨ contém o endereço da memória a ser acessado.
- **MBR** (memory buffer register) ⇨ guarda o conteúdo de uma ou mais células da memória.

2.2 Hardware

Memória Cache

- **Objetivo** ⇒ minimizar a diferença entre a velocidade do processador e a velocidade da memória principal
- **Características**
 - Volátil
 - Alta velocidade
 - Pequena capacidade
 - Alto custo
- **Cache hit** ⇒ o processador encontra o dado na cache
- **Cache miss** ⇒ o processador precisa acessar a MP para pegar o dado
- **Princípio da localidade** ⇒ dados localizados próximos entre si ⇒ a cache pode se antecipar ao pedido.



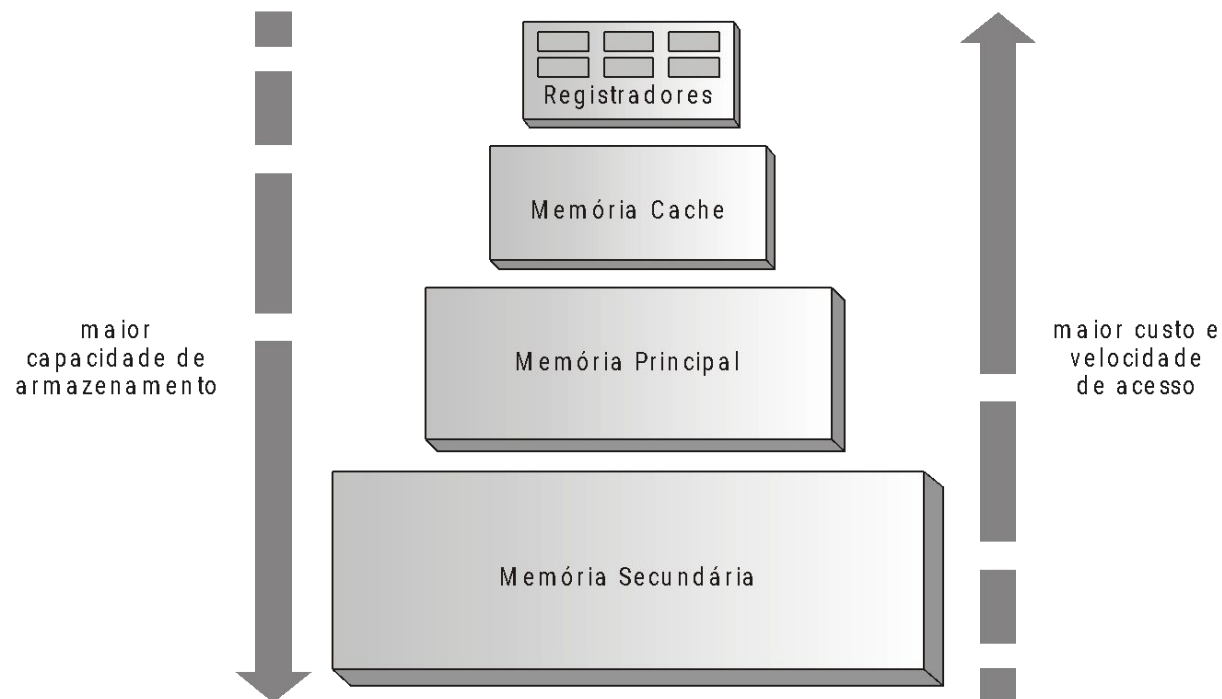
2.2 Hardware

Memória Secundária

- Não-volátil, acesso lento, maior capacidade de armazenamento
- Exemplos: fita magnética e disco magnético



Relação entre dispositivos de Armazenamento



2.2 Hardware

Dispositivos de Entrada e Saída

- Objetivo ⇨ permitir a comunicação entre o SO e mundo externo
- Os dispositivos utilizados como memória secundária
 - Discos
 - Fitas magnéticas
- Os dispositivos que têm por finalidade a comunicação usuário-máquina
 - Teclados
 - Monitores
 - Impressoras
 - Scanner
 - Mouse

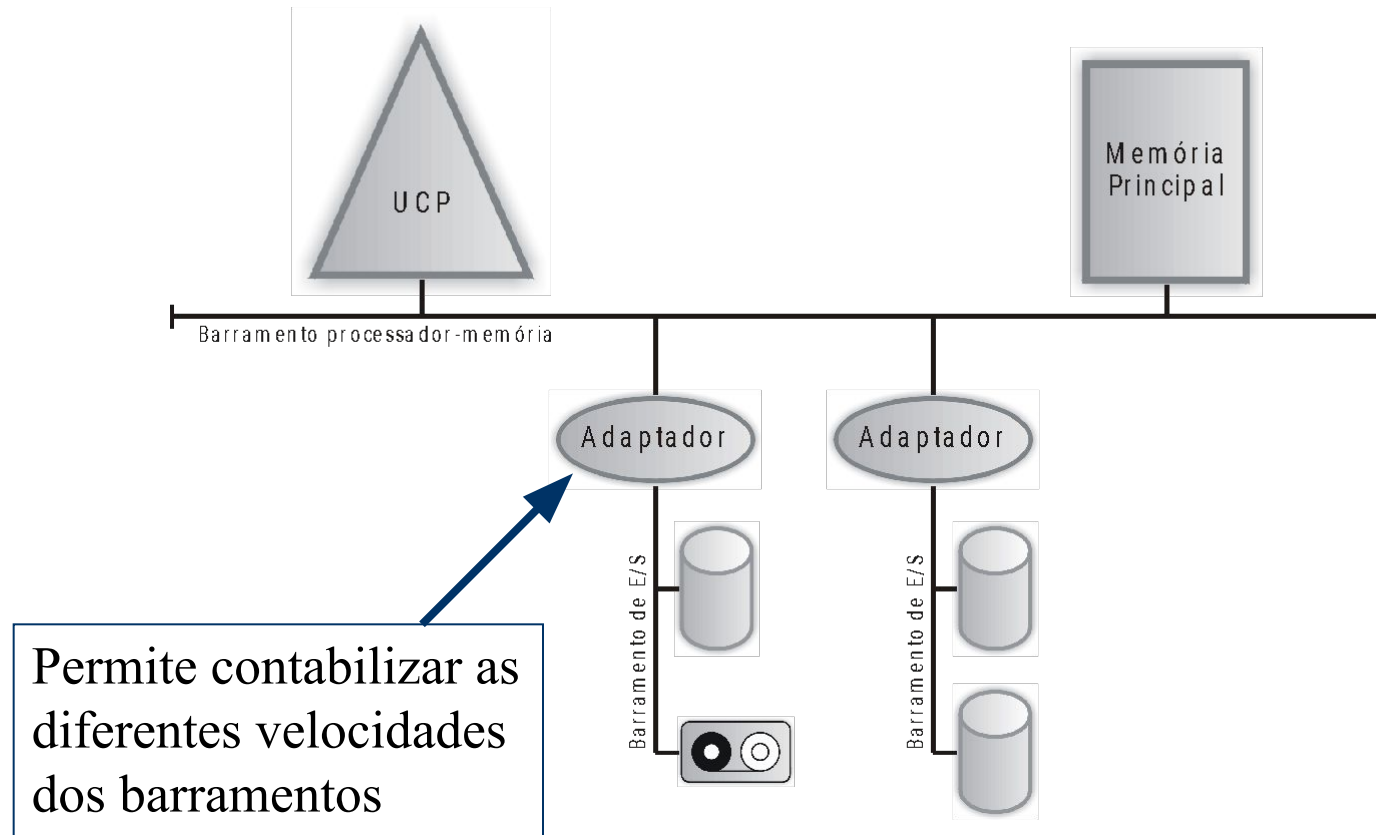
2.2 Hardware

Barramento ou Bus

- É um meio físico de comunicação entre as unidades funcionais de um sistema computacional
- Geralmente possui
 - **Linhas de controle** ⇨ trafegam informações de sinalização
 - **Linhas de dados** ⇨ trafegam informações como instruções, operandos e endereços.
- **Classificação:**
 - Barramentos processador-memória ⇨ curta-extensão e alta velocidade
 - Barramentos de E/S ⇨ maior extensão, mais lentos e permitem conexões de diferentes dispositivos
 - Barramentos de backplane ⇨ integra barramentos

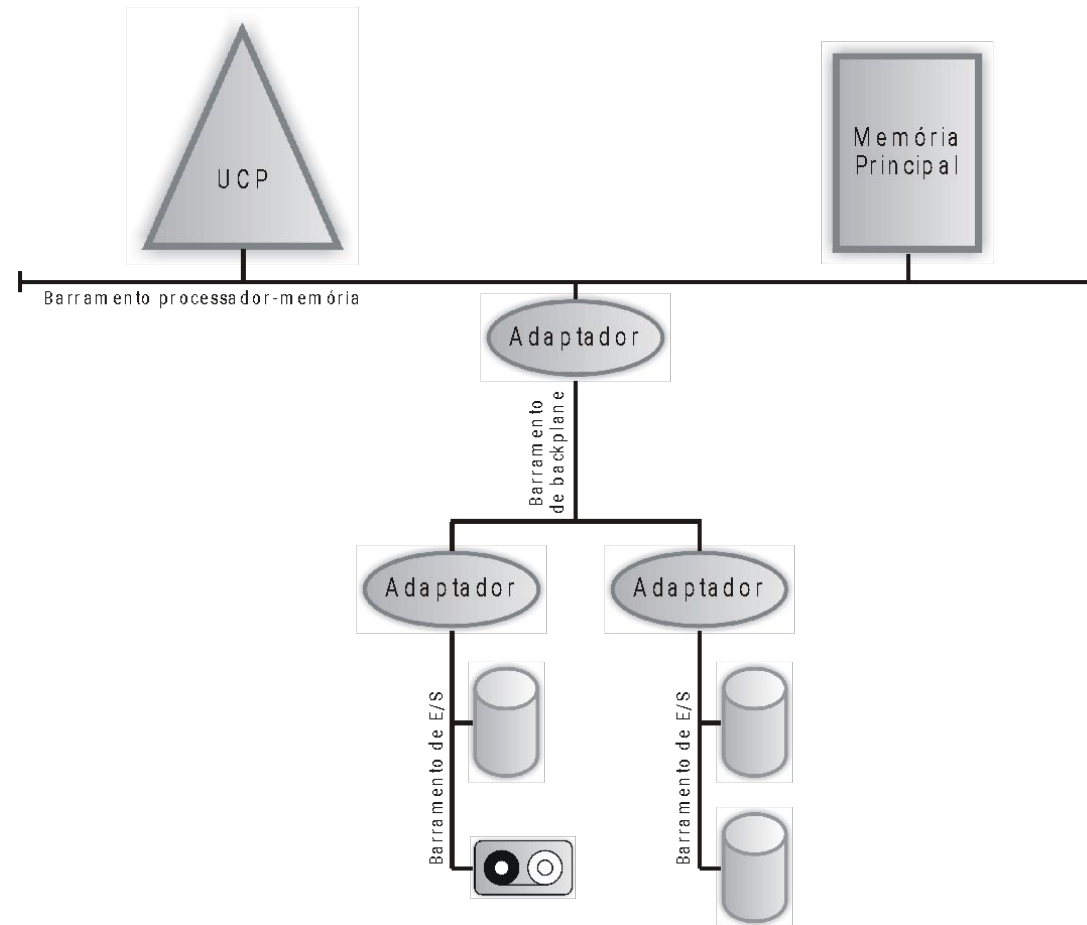
2.2 Hardware

Barramentos processador-memória e de E/S



2.2 Hardware

Barramento de backplane



2.2 Hardware

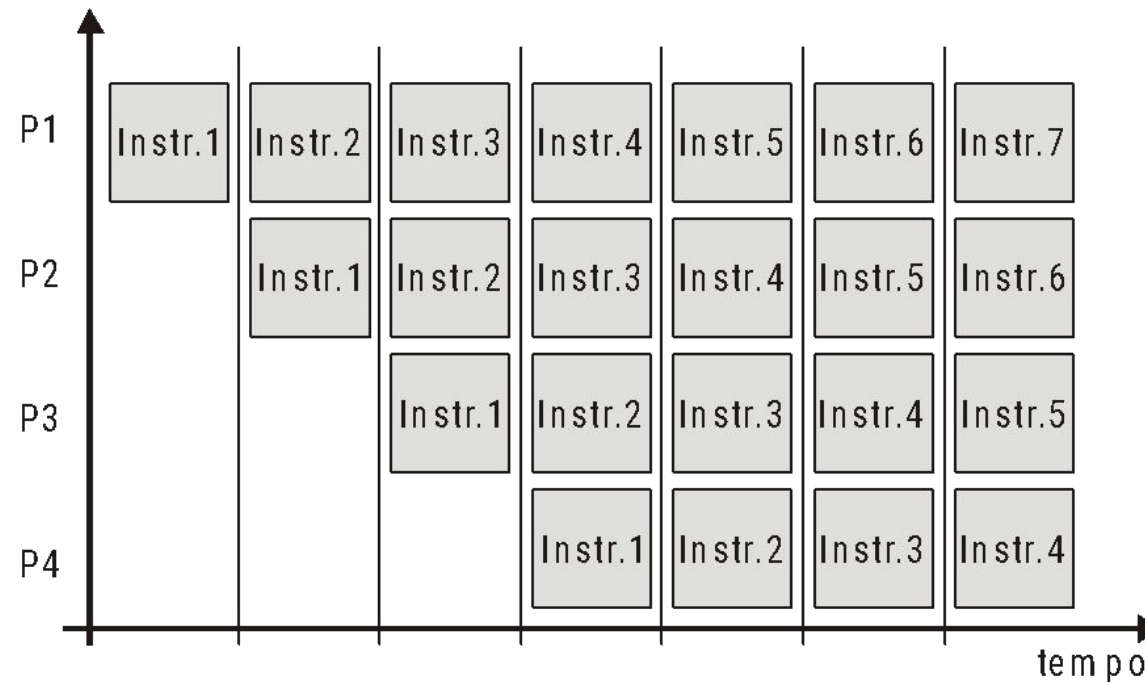
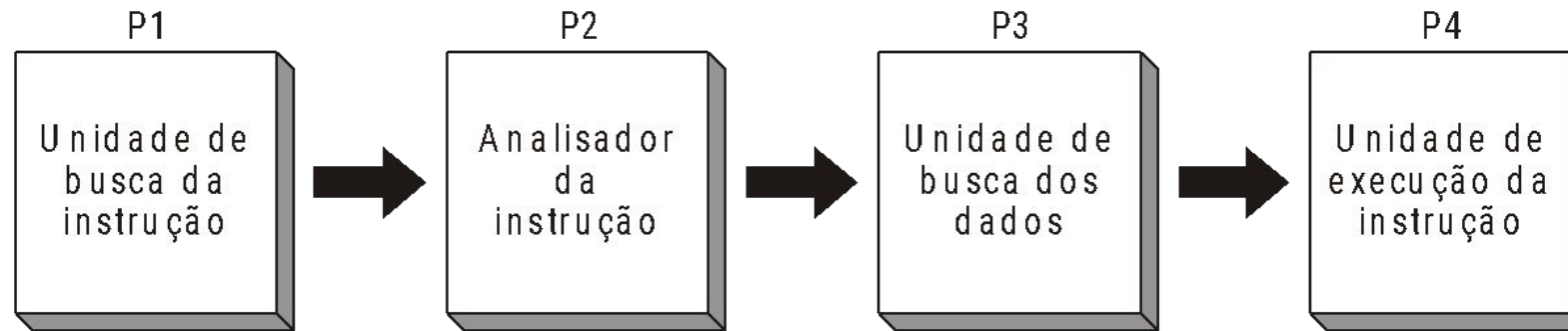
Pipelining

- É uma técnica que permite ao processador executar múltiplas instruções paralelamente em estágios diferentes
- Semelhante a uma linha de montagem
- Pode ser empregado em sistemas com um ou mais processadores
- Aumenta o desempenho dos sistemas operacionais.



2.2 Hardware

Arquitetura pipeline com quatro estágios



2.2 Hardware

Arquiteturas RISC e CISC

- **Linguagem de máquina** ⇒ linguagem que o processador é capaz de entender
- Cada processador possui um conjunto definido de **instruções** de máquina definido pelo fabricante
- Diferentes processadores podem utilizar diferentes linguagens de máquina
- **RISC** (Reduced Instruction Set Computer) ⇒ possui poucas instruções de máquina. Instruções são executadas mais rapidamente. Facilita a implementação do pipelining. Ex: SPARC da Sun.
- **CISC** (Complex Instruction Set Computers) ⇒ possui instruções complexas que são interpretadas por microprogramas. Ex: Pentium da Intel.

2.2 Hardware

Máquina de níveis



Nos processadores **RISC**, um programa de linguagem de máquina é executado diretamente pelo hardware.

2.2 Hardware

Comparando Arquiteturas

Instruções - Arquitetura RISC	Instruções - Arquitetura CISC
Poucas	Muitas
Executadas pelo hardware	Executadas por microcódigo
Formato fixo	Diversos formatos
Utilizam poucos ciclos de máquina	Utilizam múltiplos ciclos
Poucos modos de endereçamento	Diversos modos de endereçamento
Muitos registradores	Poucos registradores
Pipelining	Pouco uso de pipelining

2.2 Hardware

Análise de Desempenho

- Verificar o desempenho de processadores
- Ciclo de clock \Rightarrow intervalo de tempo entre os pulsos de um sinal de clock
- Frequência do clock \Rightarrow é o inverso do ciclo do clock e é dado em Kiloherztz (KHz) ou Megahertz (MHz)
- Tempo de UCP \Rightarrow tempo para executar um mesmo programa

$$\text{Tempo de UCP} = n^{\circ} \text{ de ciclos} \times \text{ciclo de clock}$$

ou

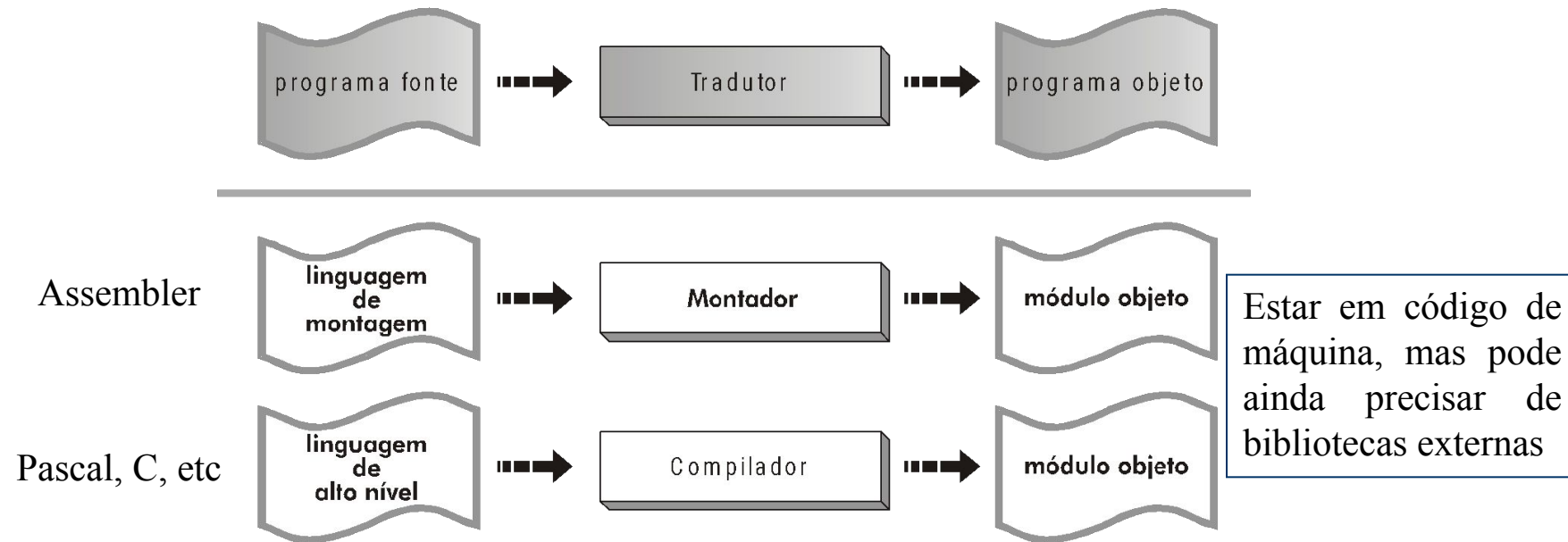
$$\text{Tempo de UCP} = \text{ciclo de clock} / \text{frequência do clock}$$

2.3 Software

Nomenclatura

- **Utilitários** ⇨ fornecem serviços complementares ao SO
- **Aplicativos ou aplicações** ⇨ desenvolvidos pelos usuários

Tradutor: montador e compilador



2.3 Software

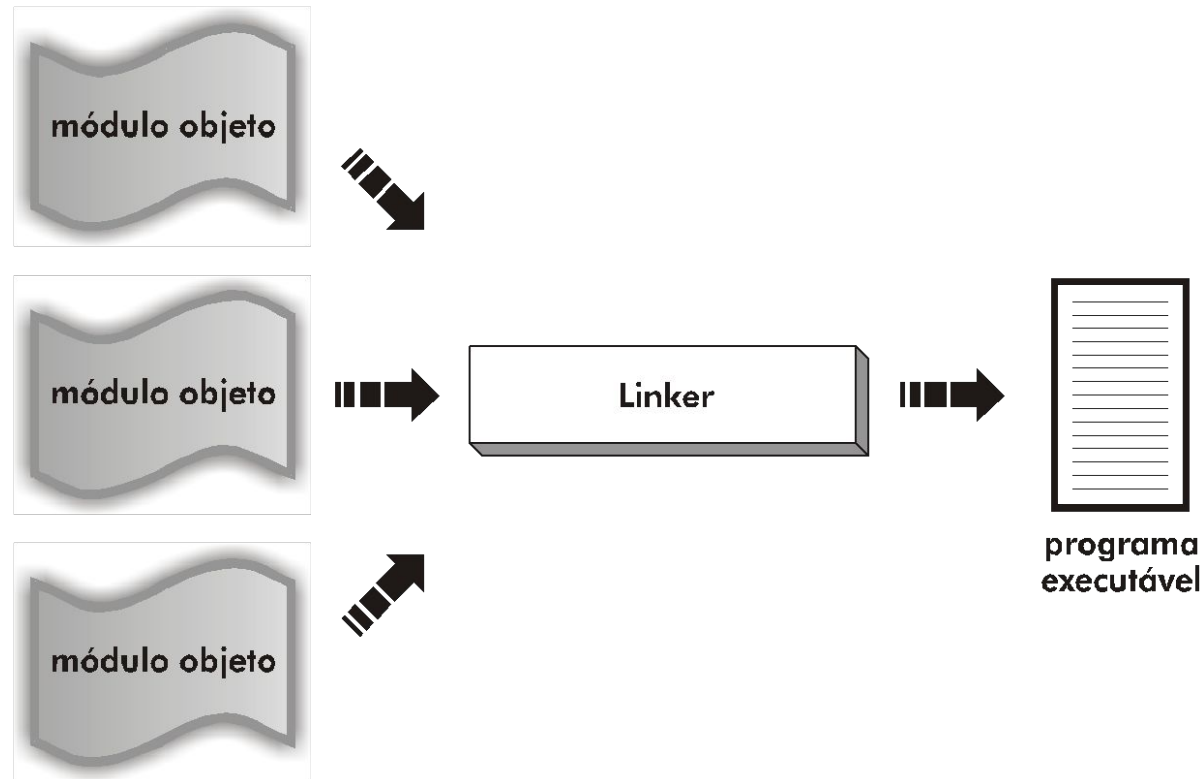
Interpretador

- Tradutor que não gera módulo-objeto
- Traduz cada comando e executa
- Desvantagem: tempo na tradução
- Vantagem: dados dinâmicos (mudam de tipo durante a execução)
- Exemplos: Basic e Perl

2.3 Software

Linker ou Editor de Ligação

- Resolve todas as referências simbólicas existentes entre os módulos e reserva memória para a execução do programa.



2.3 Software

Loader ou Carregador

- Carregar na memória principal um programa para ser executado
- Loader absoluto ⇒ precisa conhecer apenas o endereço de memória inicial e o tamanho do módulo para realizar o carregamento
- Loader relocável ⇒ o programa pode ser carregado em qualquer posição de memória.

Depurador

- Permite acompanhar a execução de um programa e detectar erros
- Visualização e alteração de variáveis
- Breakpoints ⇒ pontos de parada
- Watchpoints ⇒ monitora conteúdo de variáveis

2.3 Software

Interpretador de Comandos e Linguagem de Controle

- Interpretador de comandos ou shell permite ao usuário interagir com o sistema
- Linguagem de controle são os comandos disponíveis pelo interpretador
- Em geral, o interpretador de comandos não faz parte do sistema operacional

Ativação/Desativação do Sistema

- Ativação ou boot ⇒ carga do SO e execução dos arquivos de inicialização
- Desativação ou shutdown

