

Sistemas Operacionais

Conceitos de Arquitetura

Edeyson Andrade Gomes

www.edeyson.com.br

Roteiro da Aula

- ▶ Máquinas de Níveis
- ▶ Revisão de Conceitos de Arquitetura



Máquina de Níveis

Máquina de níveis

- ▶ **Computador como máquina de níveis ou camadas:**
 - ▶ Nível 0 – Hardware;
 - ▶ Nível 1 – Sistema Operacional.
- ▶ **Hardware:**
 - ▶ Dispositivos físicos, microprogramação, linguagem de máquina.

Máquina de níveis

- ▶ **Software:**

- ▶ Sistema operacional, utilitários e aplicativos;
- ▶ Linguagem utilizada em cada um dos níveis varia da mais elementar (baixo nível) à mais sofisticada (alto nível).



Conceitos de Arquitetura

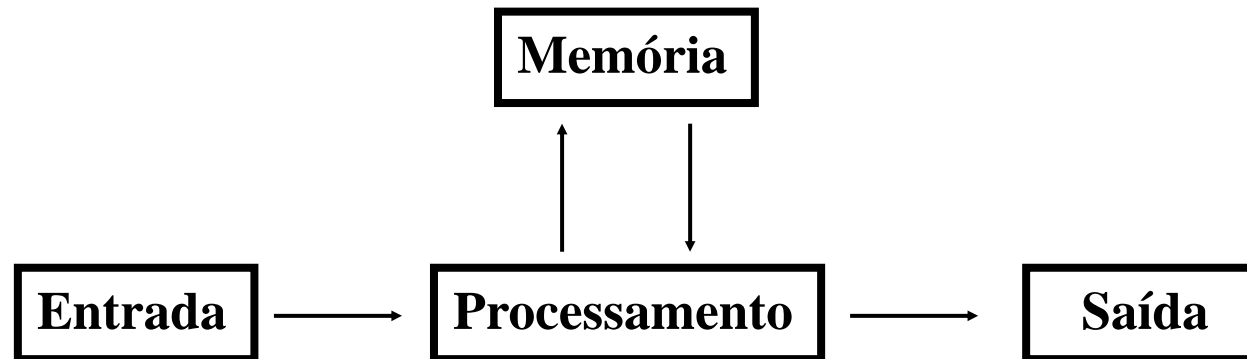
Revisão

Conceitos

▶ Hardware

- ▶ Três subsistemas básicos:
 - ▶ Unidade Central de Processamento;
 - ▶ Memória principal;
 - ▶ Dispositivos de entrada e saída.
- ▶ Subsistemas são também chamados de unidades funcionais;
- ▶ Implementações podem variar a depender da arquitetura.

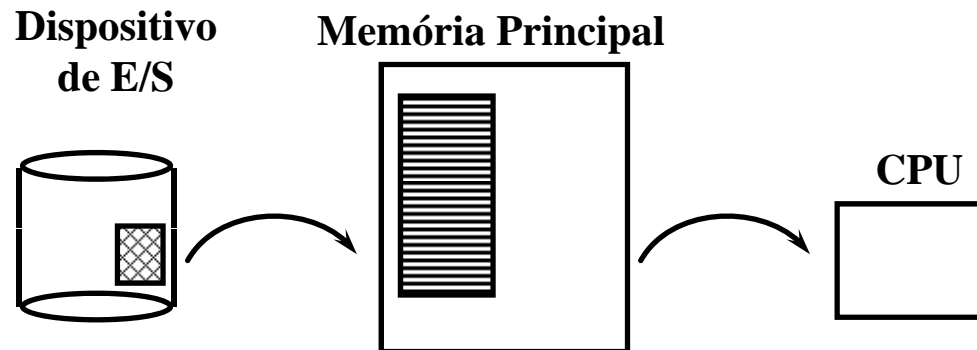
Modelo de Computador



Modelo de Computador

◆ Execução de programas

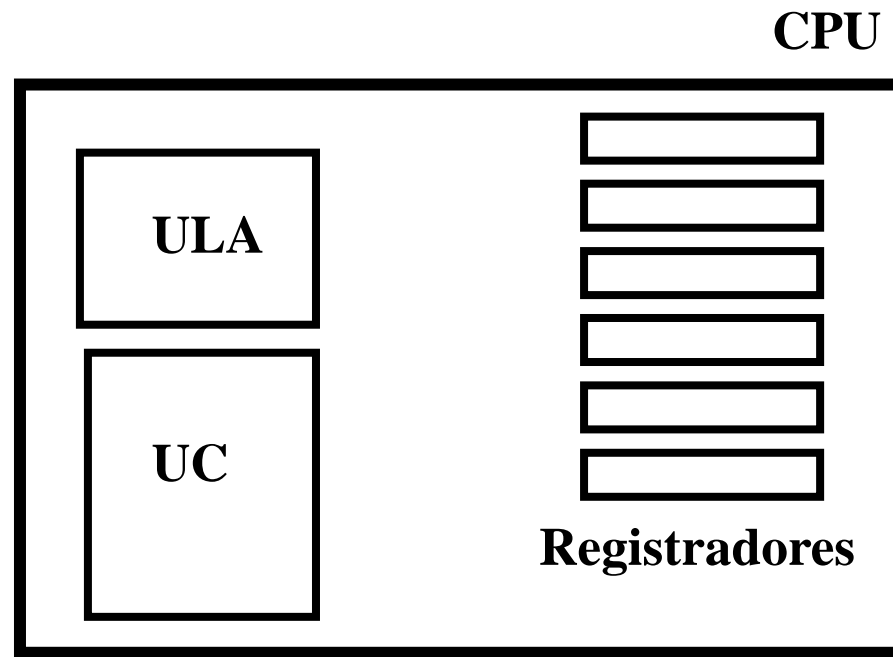
- Programa armazenado em dispositivo de E/S
- Carga do programa na memória (SO)
- Execução das instruções (uma a uma) na CPU



CPU

- ▶ Controla cada unidade funcional do sistema
- ▶ Execução das instruções dos programas
 - ▶ Aritméticas
 - ▶ Comparação
 - ▶ Movimentação
- ▶ Componentes Básicos

CPU



CPU

- ▶ **Unidade de Controle (UC)**
 - ▶ Controla as atividades de todos componentes
 - ▶ Decodifica cada instrução a ser executada
 - ▶ Usa sinal de sincronismo
 - ▶ **clock**

CPU

- ▶ **Unidade Aritmética e Lógica (ULA)**
 - ▶ Adição, Subtração, Multiplicação, etc.
 - ▶ Operações booleanas:
 - ▶ And, Or, Xor, etc.

CPU

- ▶ **Registradores**
 - ▶ Locais de armazenamento de informações (bits)
 - ▶ Pequenas posições de memória
 - ▶ Localizadas dentro da CPU
 - ▶ Acesso mais rápido que à memória principal
 - ▶ Número e tamanho varia de acordo com a CPU

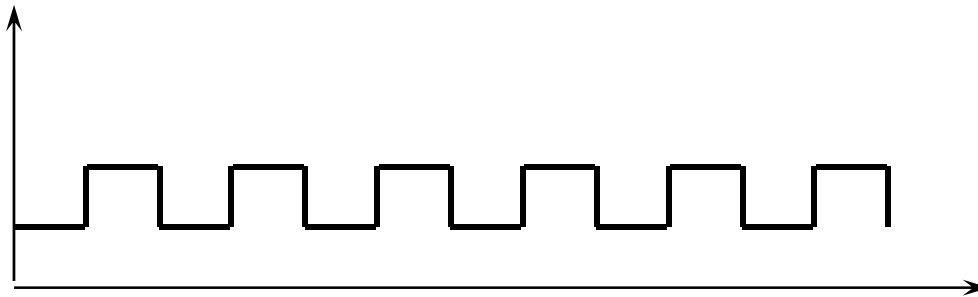
CPU

- ▶ **Registradores de Propósito Geral**
 - ▶ Disponível para os programadores / compiladores
 - ▶ Armazenamento temporário (Acumuladores)
- ▶ **Registradores de Propósito Específico**
 - ▶ Program Counter (PC) ou Instruction Pointer (IP)
 - ▶ Stack Pointer (SP)
 - ▶ Flag ou Status Register ou Program Status Word (PSW)
 - ▶ Instruction Register

CPU

▶ Clock

- ▶ Sinal elétrico de frequência constante
- ▶ Medido em Hertz (Hz) - pulsos por segundo
- ▶ Usado como referência pelos componentes do sistema
- ▶ Uma instrução pode durar vários pulsos de clock



CPU

◆ Conjunto de Instruções

- Instruções de máquinas disponíveis na CPU
- Tipos e quantidade variam de acordo com a CPU
- Compatibilidade de programas
- RISC x CISC

CPU

- ◆ Suponha que um robôX atende comandos em binário.
 - ◆ 00 – avança um passo
 - ◆ 01 – retorna um passo
 - ◆ 10 – desloca-se um passo à esquerda
 - ◆ 11 – desloca-se um passo à direita
- ◆ O que ele fará se em sua memória forem armazenadas as instruções:
 - ◆ 00 00 10 00 11 00 11

- ◆ Suponha que um robôY atende comandos em binário.
 - ◆ 00 – retorna um passo
 - ◆ 01 – avança um passo
 - ◆ 10 – desloca-se um passo à direita
 - ◆ 11 – desloca-se um passo à esquerda
- ◆ O que ele fará se em sua memória forem armazenadas as instruções:
 - ◆ 00 00 10 00 11 00 11

CPU

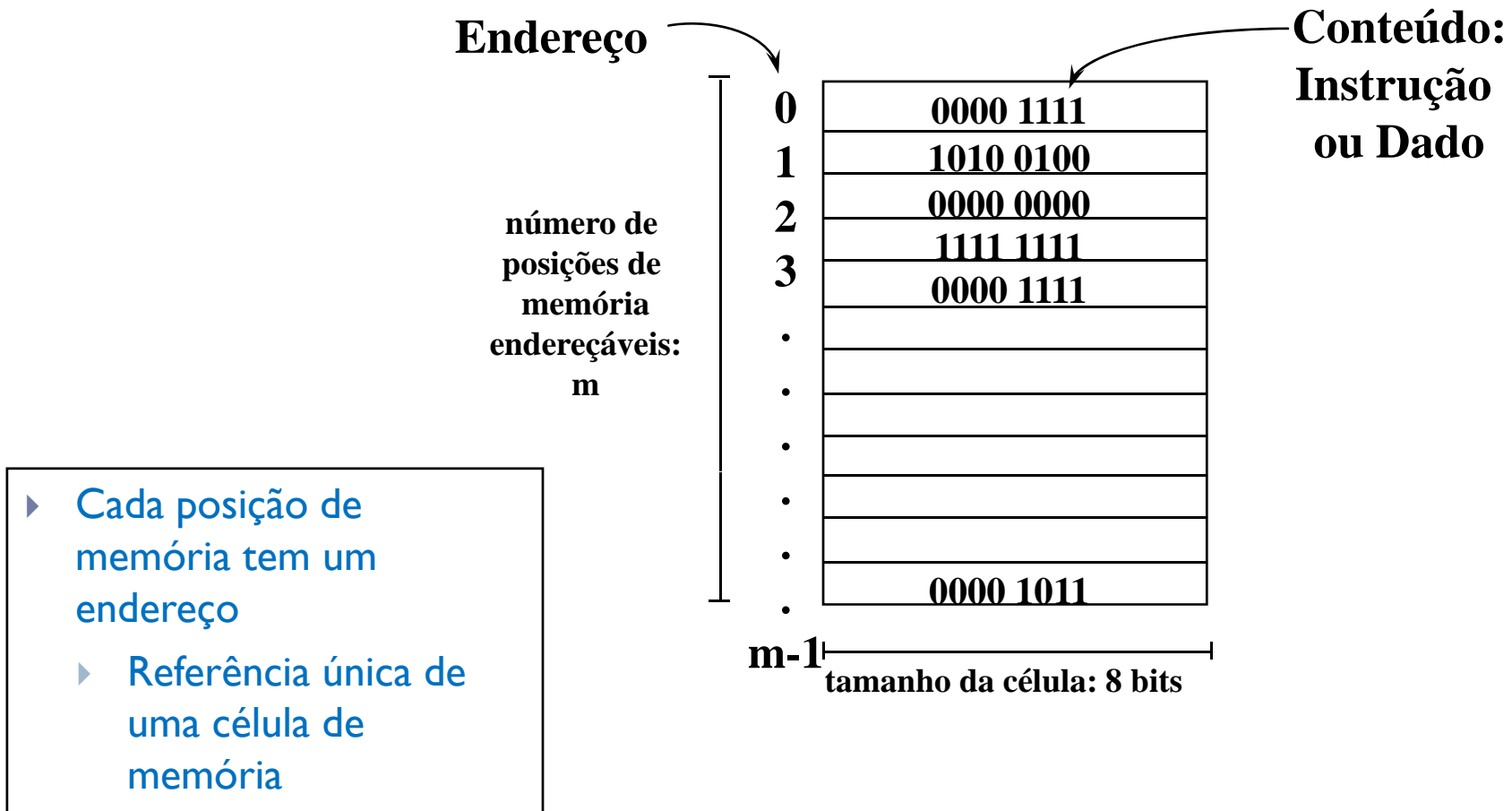
◆ Desempenho de processadores

- Depende de:
 - » Conjunto de instruções disponível
 - » Clock
 - » Tipo da aplicação: comercial x científica
- Benchmark
 - » Usado para comparar processadores diferentes

Memória

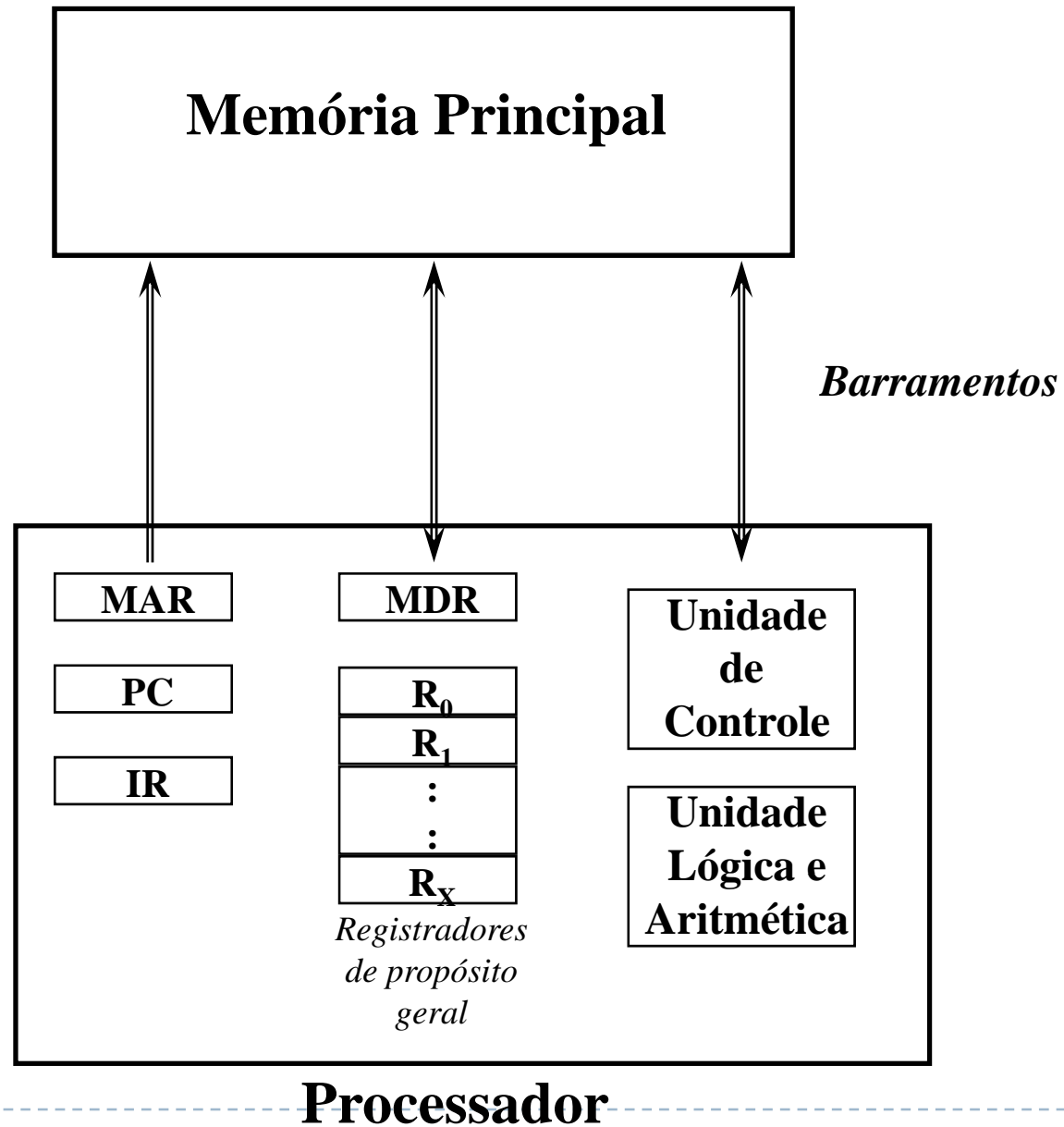
- ▶ **Memória Principal**
 - ▶ Armazena informações (bits)
 - ▶ Programas (instruções)
 - ▶ Dados
 - ▶ Composta por várias unidades de acesso
 - ▶ Células
 - ▶ Posições de memória
 - ▶ Número determinado de bits (normalmente 8)

Memória



Memória

- ▶ **Interligação Memória - Processador**
 - ▶ Registradores de uso específico:
 - ▶ Memory Address Register - MAR - REM
 - ▶ Memory Data Register - MDR - RDM



Memória

- ▶ **Operação de Leitura da Memória**
 - ▶ CPU armazena endereço da célula a ser lida no MAR
 - ▶ CPU gera sinal de controle indicando que a operação é de leitura da memória
 - ▶ Memória recupera informação armazenada na posição endereçada e coloca no barramento de dados, chegando ao MDR

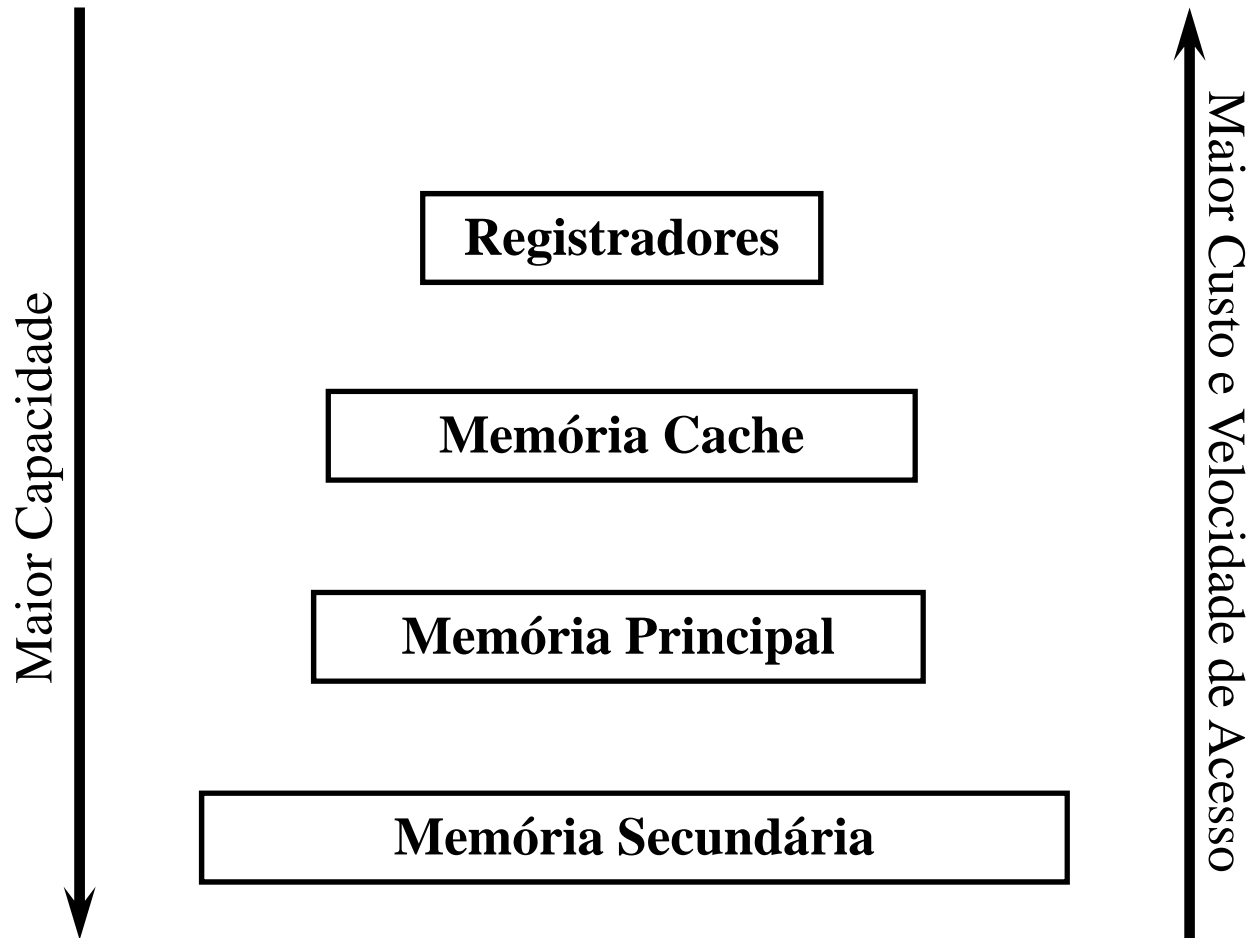
Memória

- ◆ **Operação de Gravação na Memória**
 - CPU armazena endereço da célula a ser gravada no MAR
 - CPU armazena a informação a ser gravada no MDR
 - CPU gera sinal de controle indicando que a operação é de gravação na memória
 - Memória armazena informação do barramento de dados na posição endereçada

Memória

- ▶ **Memória Cache**
 - ▶ Memória de alta velocidade
 - ▶ Localizada entre CPU e Memória Principal
 - ▶ Aumento de desempenho com custo razoável
 - ▶ Algoritmo
 - ▶ Hit rate - taxa de acertos
 - ▶ Interna x Externa

Memória



Dispositivos de E/S

- ▶ **Dispositivo de E/S**
 - ▶ Comunicação com meio externo
 - ▶ Memória Secundária e Interface Homem-Máquina
- ▶ **Barramentos**
 - ▶ Conjunto de fios paralelos
 - ▶ Interliga CPU, Periféricos e Memória Principal

Barramentos

- ▶ **Barramento de Dados**
 - ▶ Número de bits por operação de E/S ou acesso à Memória
 - ▶ Processadores de 8, 16, 32 ou 64 bits
 - ▶ Fluxo Bi-direcional (leitura / gravação)
- ▶ **Barramento de Endereços**
 - ▶ Capacidade de Armazenamento = 2^n
 - ▶ Fluxo Uni-direcional (CPU → MEM ou E/S)

Barramentos

- ▶ **Barramento de Controle**
 - ▶ Diversos sinais de controle
 - ▶ Read / Write
 - ▶ Controle de Interrupções
 - ▶ Fluxo Uni ou Bi-direcional (depende do sinal de controle)

Pipelining

- ▶ **Análogo à Linha de Montagem**
 - ▶ Tarefas divididas em subtarefas
 - ▶ Execução em diferentes estágios

- ▶ **Divisão da execução de uma instrução**
 - ▶ Busca da Instrução e Operandos
 - ▶ Execução da Operação
 - ▶ Armazenamento dos Resultados

Pipelining

- ▶ Paralelismo \Rightarrow Aumento de Desempenho
- ▶ Técnica empregada em sistemas com um ou mais processadores



Conceitos de Software

Conceitos de Software

▶ Tradução

- ▶ Necessidade de codificar instruções para máquina

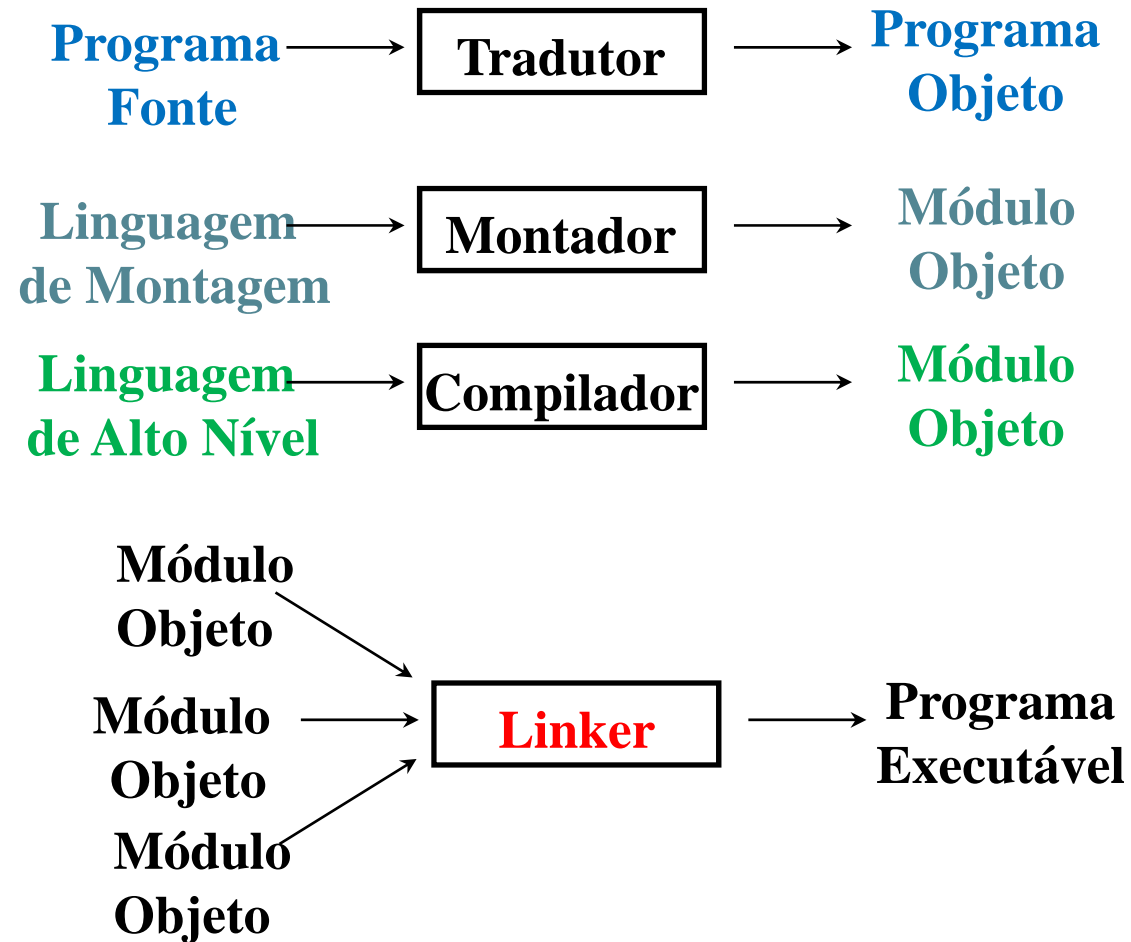
▶ Evolução

- ▶ Programação via painéis
- ▶ Linguagens de programação

▶ Linguagens de Programação

- ▶ Maior independência do HW
- ▶ Maior facilidade de desenvolvimento / manutenção

Tradução



Tradução

- ▶ **Linguagens de Montagem (Assembly)**
 - ▶ Mnemônicos associados às instruções do processador
 - ▶ Específica para cada processador
 - ▶ Montadores (Assembler)

Tradução

- ▶ **Linguagem de Alto Nível**
 - ▶ Sem relação direta com o processador
 - ▶ Transportabilidade
 - ▶ Pascal, Fortran, Cobol, VB, Java
 - ▶ Compiladores
 - ▶ Interpretadores
 - ▶ Caso especial do Java: JVM

Tradução

- ▶ **Módulos Objeto**
 - ▶ Código de máquina não executável
 - ▶ Referências a módulos externos não resolvidas

- ▶ **Módulos Executáveis**
 - ▶ Códigos de máquina executáveis
 - ▶ Linkers
 - ▶ Bibliotecas
 - ▶ Relocação

Conceitos de Software

- ▶ **Loader - Carregador**

- ▶ Carga do programa na memória para execução
- ▶ Absoluto x Relocável

- ▶ **Debugger - Depurador**

- ▶ Pesquisa de erros de lógica
- ▶ Rastreamento

Conceitos de Software

- ▶ Linguagem de Controle
 - ▶ Comunicação Usuário com o SO
 - ▶ Caracter x Gráfica
 - ▶ Interpretador de Comandos - Shell
 - ▶ Arquivos de Comandos

Conceitos de Software

- ▶ Linguagem de Máquina
 - ▶ Instruções do Processador
 - ▶ Acesso aos Registradores / Memória
 - ▶ Modos de endereçamento
 - ▶ Tipos de dados

Conceitos de Software

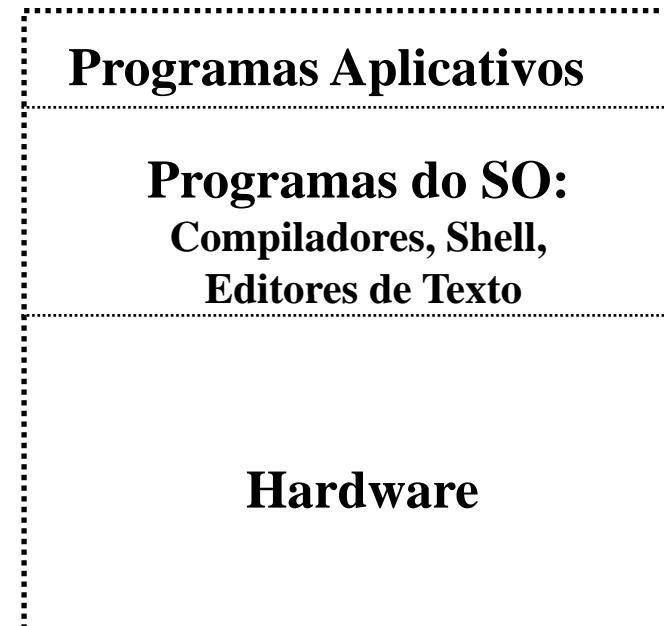
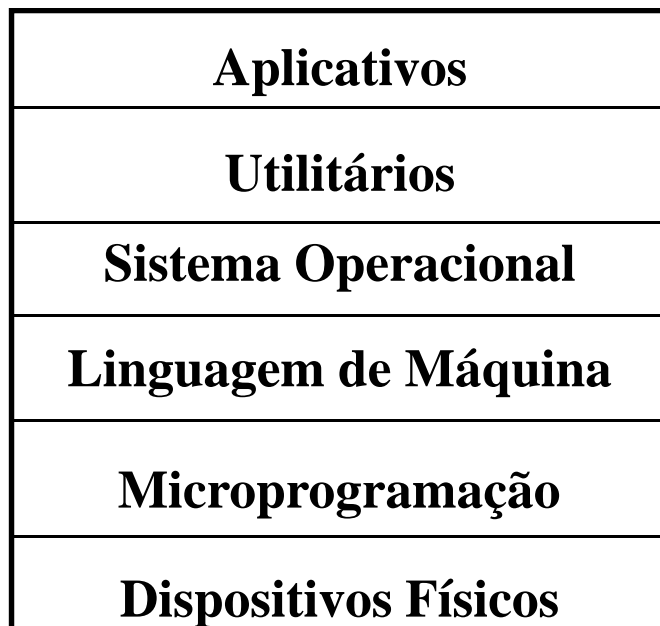
- ▶ Linguagem de Máquina
 - ▶ Programa em linguagem de máquina
 - ▶ Totalmente codificado em binário
 - ▶ Diretamente processado pela CPU
 - ▶ Não requer tradução ou relocação
 - ▶ Não pode ser executado em outros processadores

Conceitos de Software

▶ Microprogramação

- ▶ Cada instrução em linguagem de máquina tem um microprograma associado
- ▶ O microprograma é formado por várias microinstruções
- ▶ Microinstruções são executadas pelo hardware
- ▶ Arquitetura CISC
- ▶ Processadores microprogramáveis
 - ▶ Aceitam novas instruções - novos microprogramas

Máquina de Níveis





Conclusões

