



ISO-011 – Sistemas Operacionais e Redes de Computadores

Prof. Me. Anderson Vanin

Quem sou eu

- Técnico Eletrônico
- Bacharel em Ciência da Computação
- Pós Graduação em Banco de Dados
- Mestre em Gestão do Conhecimento e Informática (Aplicado a VC)
- Atuação no CPS desde 2006

Material e recursos para as aulas

- github.com/profandersonvanin
- Email: anderson.vanin@fatec.sp.gov.br
- Todos tem um *Github*?

ISO-011 – Sistemas Operacionais e Redes de Computadores

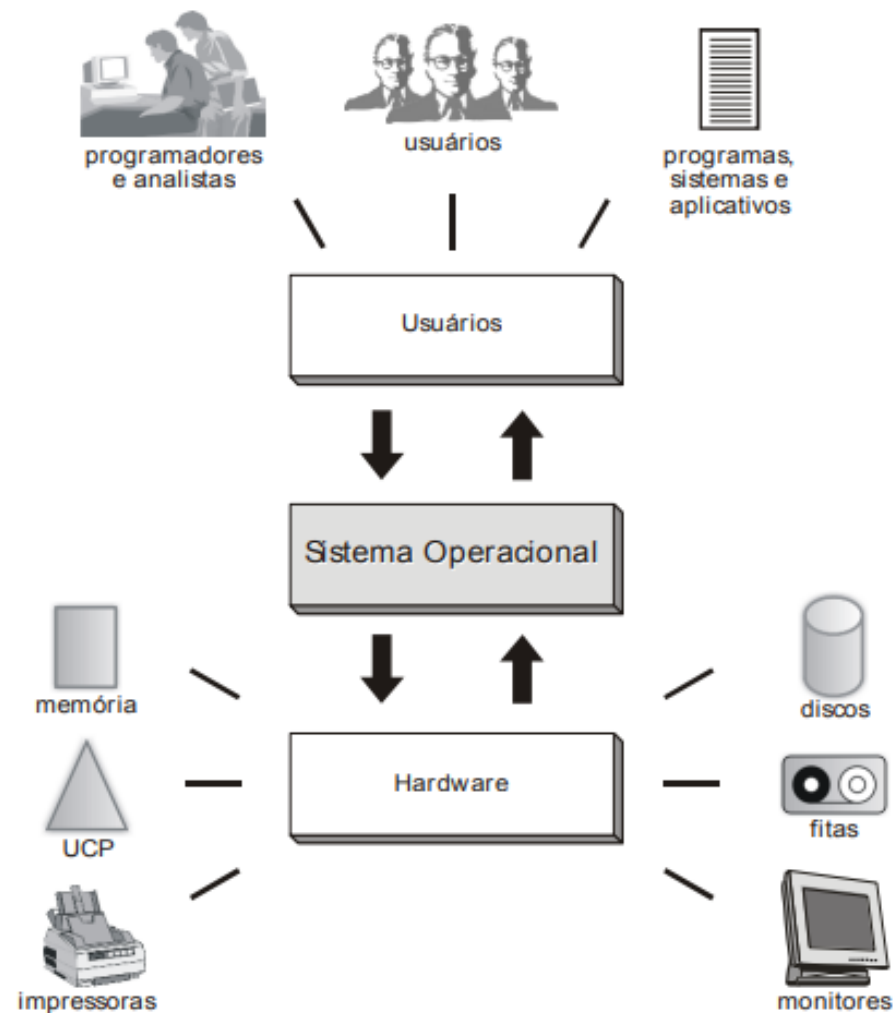
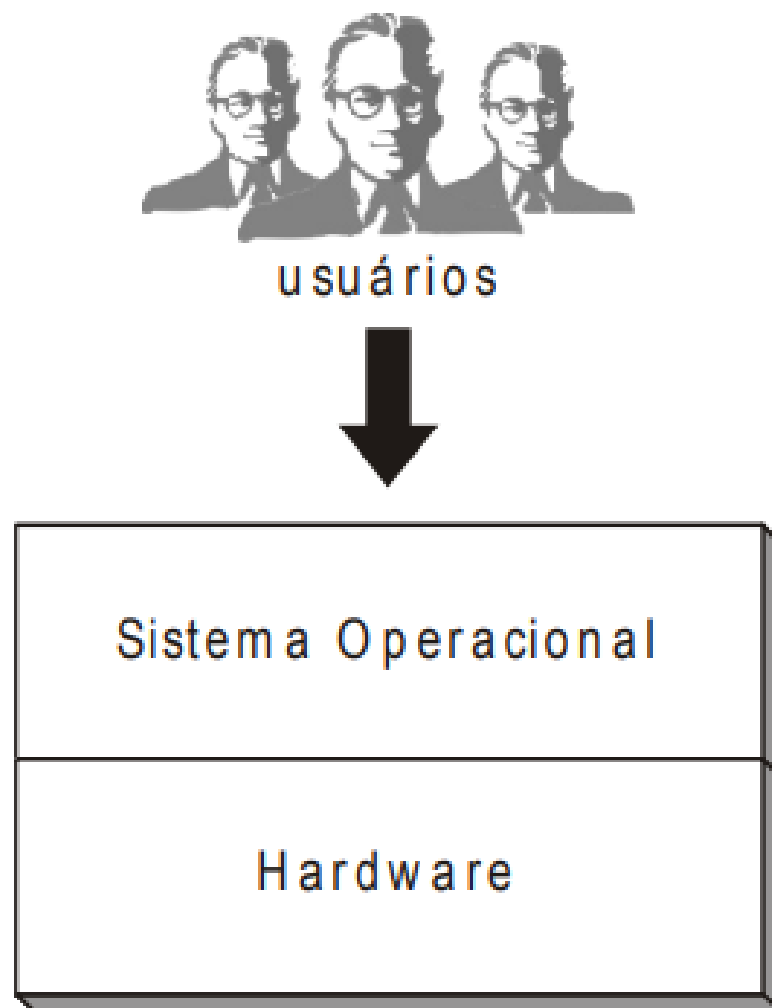
Ementa da Disciplina

Visão Geral. Introdução à Sistemas Operacionais. Conceitos Básicos de Hardware e Software. Concorrência em Sistemas Operacionais. Estrutura do Sistema Operacional. Processos e Threads. Sincronização e Comunicação entre Processos. Gerencia do Processador. Gerencia de Memória e Memória Virtual. Sistemas Operacionais distribuídos. Sistemas Operacionais de Rede. Protocolos de comunicação.

Quem são meus alunos?



Introdução



Introdução

- **Programas de Aplicação:** desenvolvidos pelo usuário.
- **Programas Utilitários:** ferramentas de auxílio às aplicações.
- **Programa:** conjunto de comandos com finalidade específica.
- **Processo:** programa em estado de execução (consumindo recursos da máquina).
- **Sistema Operacional:** programa que estende os recursos de hardware da máquina tornando o uso do equipamento mais fácil (transparência), econômico (compartilhamento de recursos), eficiente (maior aproveitamento do processador) e confiável (isolamento entre usuários).

Funções do Sistema Operacional

- Definição da interface com o usuário;
- Compartilhamento de hardware entre usuários;
- Compartilhamento de dados entre usuários;
- Gerenciamento dos dispositivos de entrada e saída;
- Tratamento e recuperação de erros.

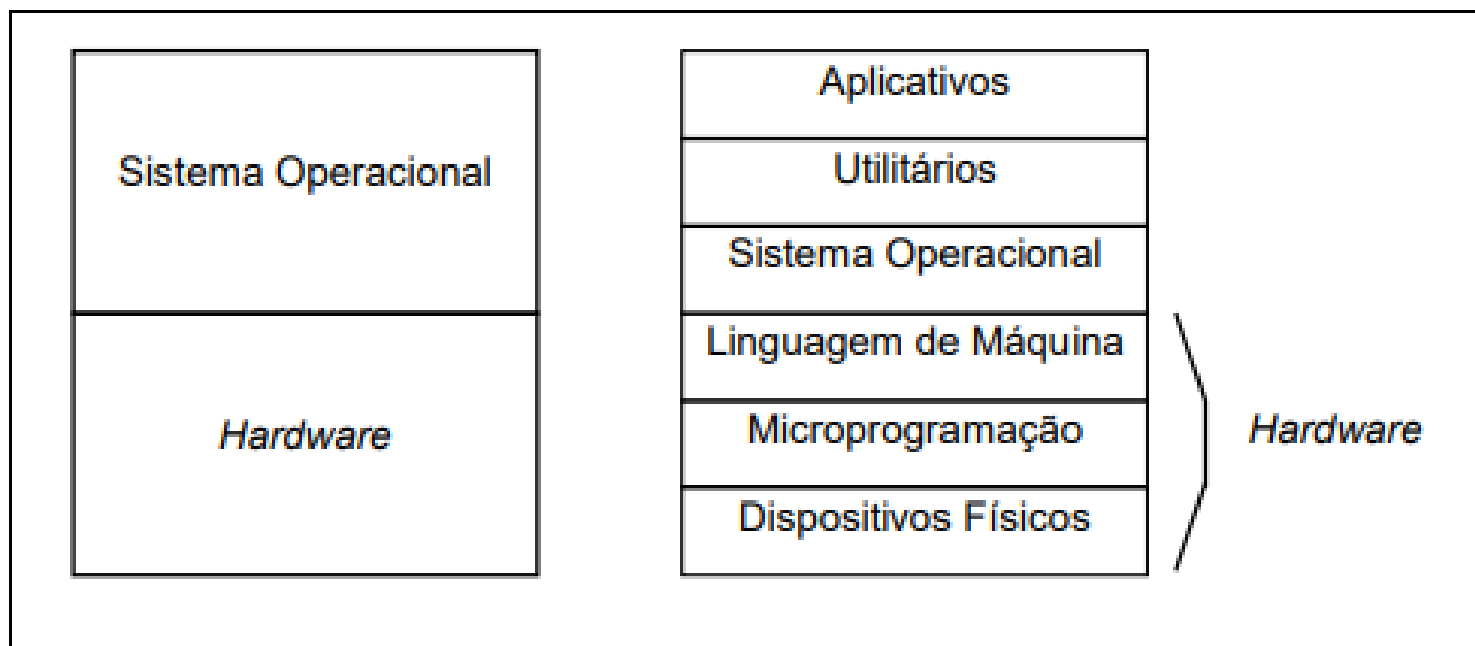
Execução do SO

Um SO não é executado de forma linear como a maioria das aplicações, com início, meio e fim.

Suas rotinas são executadas concorrentemente em função de eventos assíncronos, ou seja, eventos que podem ocorrer a qualquer momento

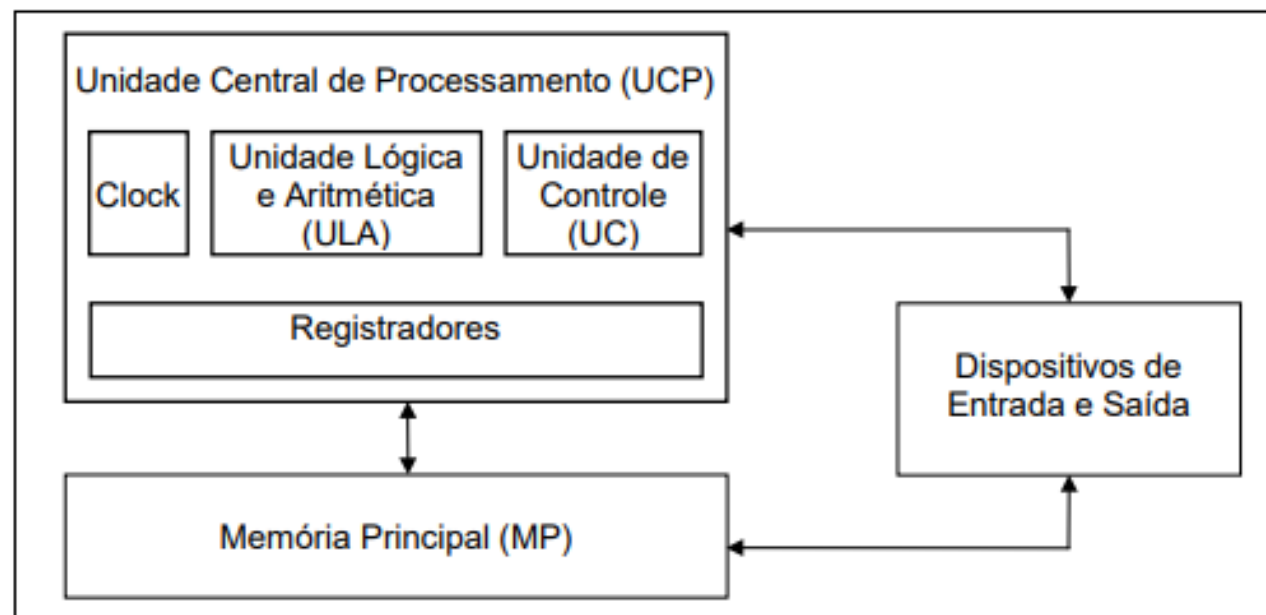
Máquina de níveis

Hardware e Software são logicamente equivalentes. Operações efetuadas pelo software podem ser implementadas em hardware e instruções executadas pelo hardware podem ser simuladas via software (Tanenbaum).



Hardware

O computador é constituído por um conjunto de componentes interligados agrupados em três subsistemas básicos (unidades funcionais): **unidade central de processamento, memória principal e dispositivos de entrada e saída**. As implementações variam de acordo com o fabricante, porém todos estão presentes.



Unidade central de processamento

Ou simplesmente processador.

- Unifica todo o sistema controlando as funções realizadas pelas unidades funcionais.
- Executa todos os programas, sendo que estes estão na **memória principal (MP)**.
- A **UCP** busca a instrução na **MP** e a interpreta para executá-la.
- A **UC** é responsável pela emissão de sinais de controle (gravação em disco ou busca de instrução na memória).
- A **ULA** é responsável pela realização de operações lógicas (comparações) e aritméticas (somas e subtrações).
- A velocidade da **UCP** é determinada pelo número de instruções que ela executa em uma unidade de tempo (não existe padronização). **MIPS** (milhões de instruções por segundo - $PS2/50 = 2 \text{ MIPS}$)

Calculo de MIPS

O MIPS refere-se às milhões de instruções por segundo executadas pelo processador de um computador. Este número dá uma ideia da velocidade de uma CPU, pois processadores mais rápidos têm um MIPS maior do que os mais lentos.

Passo 1: Divida o número de instruções por parte do tempo de execução. Por exemplo, se um computador realiza pelo menos 1 milhão de instruções em 0,05 segundo, o cálculo seria $1 \text{ milhão} / 0.05 = 20 \text{ milhões}$.

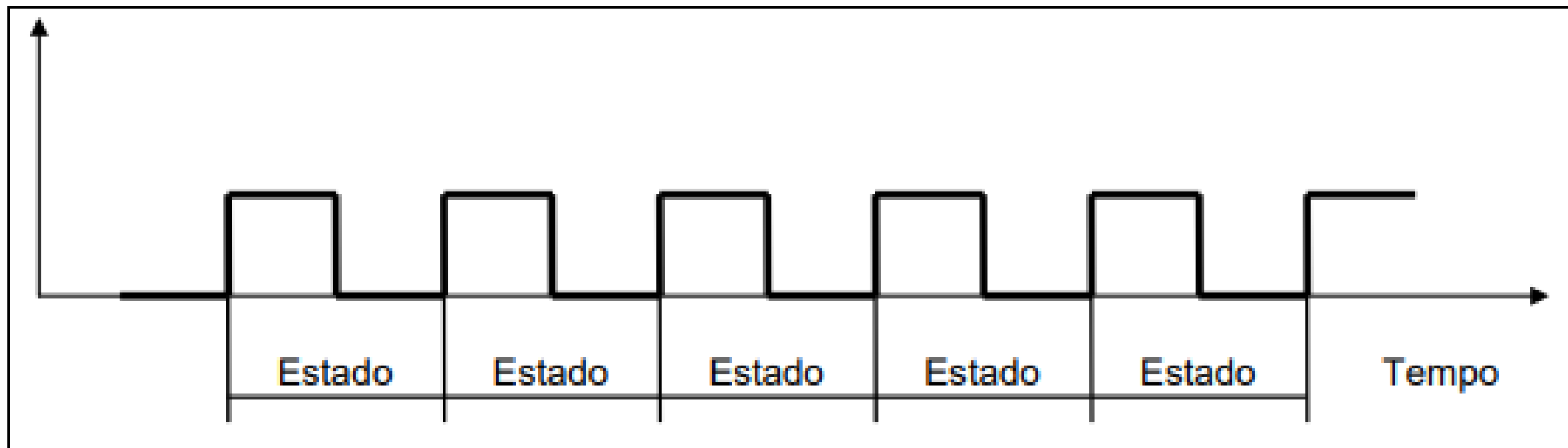
Passo 2: Divida esse número por 1 milhão para encontrar as milhões de instruções por segundo. Para este exemplo: $20000000 / 1000000 = 20 \text{ MIPS}$.

Passo 3: Alternativamente, divida o número de ciclos por segundo (CPU) com o número de ciclos por instrução (CPI) e, em seguida, divida por 1 milhão para encontrar o MIPS. Por exemplo, se um computador com CPU de 600 megahertz tinha um IPC de 3: $600 / 3 = 200$; $200 / 1 \text{ milhão} = 0.0002 \text{ MIPS}$.

Clock

- Dispositivo da UCP que gera pulsos elétricos constantes (síncronos) em um mesmo intervalo de tempo (sinal de clock).
- Este intervalo determina a frequência de geração dos pulsos e consequentemente seu período.
- A cada período do sinal de clock dá-se o nome de **estado**.
- O sinal é usado pela UC para executar as instruções.
- Como uma instrução demora vários estados, em um estado apenas parte dela é executada.
- O tempo de duração da instrução é determinado pelo seu número de estados e o tempo de duração do estado.

Clock



Registradores

- Dispositivos de alta velocidade localizados na UCP.
- Armazenam dados temporariamente.
- Alguns são de uso específico, outros de uso geral.
- O número varia de acordo com a arquitetura do processador.
- Merecem destaque:
 - **contador de instruções (program counter - PC):** armazena a instrução que a UCP deve executar.
 - **apontador da pilha (stack pointer - SP):** contém o endereço de memória do topo da pilha (estrutura do sistema que guarda informações sobre tarefas que tiveram de ser interrompidas por algum motivo).
 - **registrador de estado (program status word - PSW):** armazena informações sobre a execução do programa, como overflow

Memória Principal

- Composta por unidades chamadas células.
- **Volátil** (necessita estar energizada).
- **O tamanho da célula (em bits)** varia de acordo com a arquitetura do equipamento.
- O acesso é feito através de um número chamado **endereço de memória**.
- O endereço é especificado no registrador de endereço de memória (**memory register address - MAR**).
- Com o conteúdo desse registrador a UCP acessa o endereço.
- O conteúdo do endereço acessado é armazenado no registrador de dados de memória (**memory buffer register - MBR**).
- A capacidade de memória é limitada pelo tamanho do MAR (2^n bits, ou seja, de 0 a $2^n - 1$) (IBM PS2/50 = 16 Mb).

Memória Principal

Operação de Leitura	Operação de Gravação
1. O MAR recebe da UCP 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, indicando que uma operação de leitura deve ser realizada.	2. A UCP armazena no MBR a informação que deverá ser gravada.
3. O conteúdo da célula, identificada pelo endereço contido no MAR, é transferido para o MBR	3. A UCP gera um sinal de controle para a memória principal, indicando que uma operação de gravação deve ser realizada.
	4. A informação contida no MBR é transferida para a célula de memória endereçada pelo MAR.

Memória Cache

- **Alta velocidade.**
- Antes de acessar um dado na MP o processador busca ele na memória cache.
- Se encontra não acessa a MP, caso contrário transfere um bloco de dados a partir do dado referenciado para a cache.
- Maior performance.
- Alto custo.

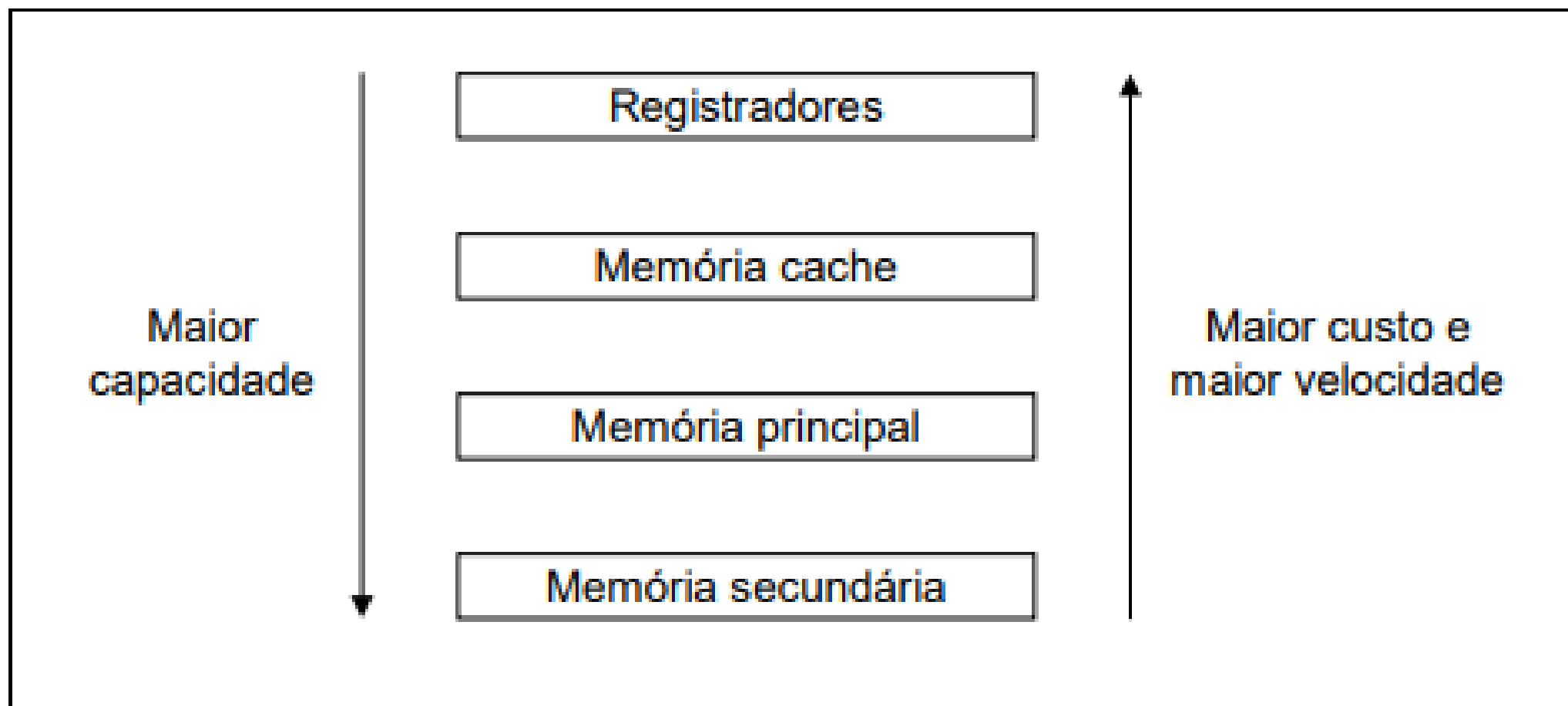
Memória Secundária

- Meio permanente de armazenamento (**não-volátil**).
- **Acesso lento** se comparado às outras memórias.
- Custo baixo.
- Capacidade de armazenamento muito superior à MP.
- Fitas magnéticas, discos magnéticos e óticos.

Dispositivos de entrada e saída

- Permitem comunicação entre o computador e o mundo externo.
- Dispositivos de memória secundária (discos e fitas) e de interface homem-máquina (teclados, monitores, impressoras, plotters).
- Interfaces mais amigáveis como scanners, canetas óticas, mouses, microfones e touch-screens.

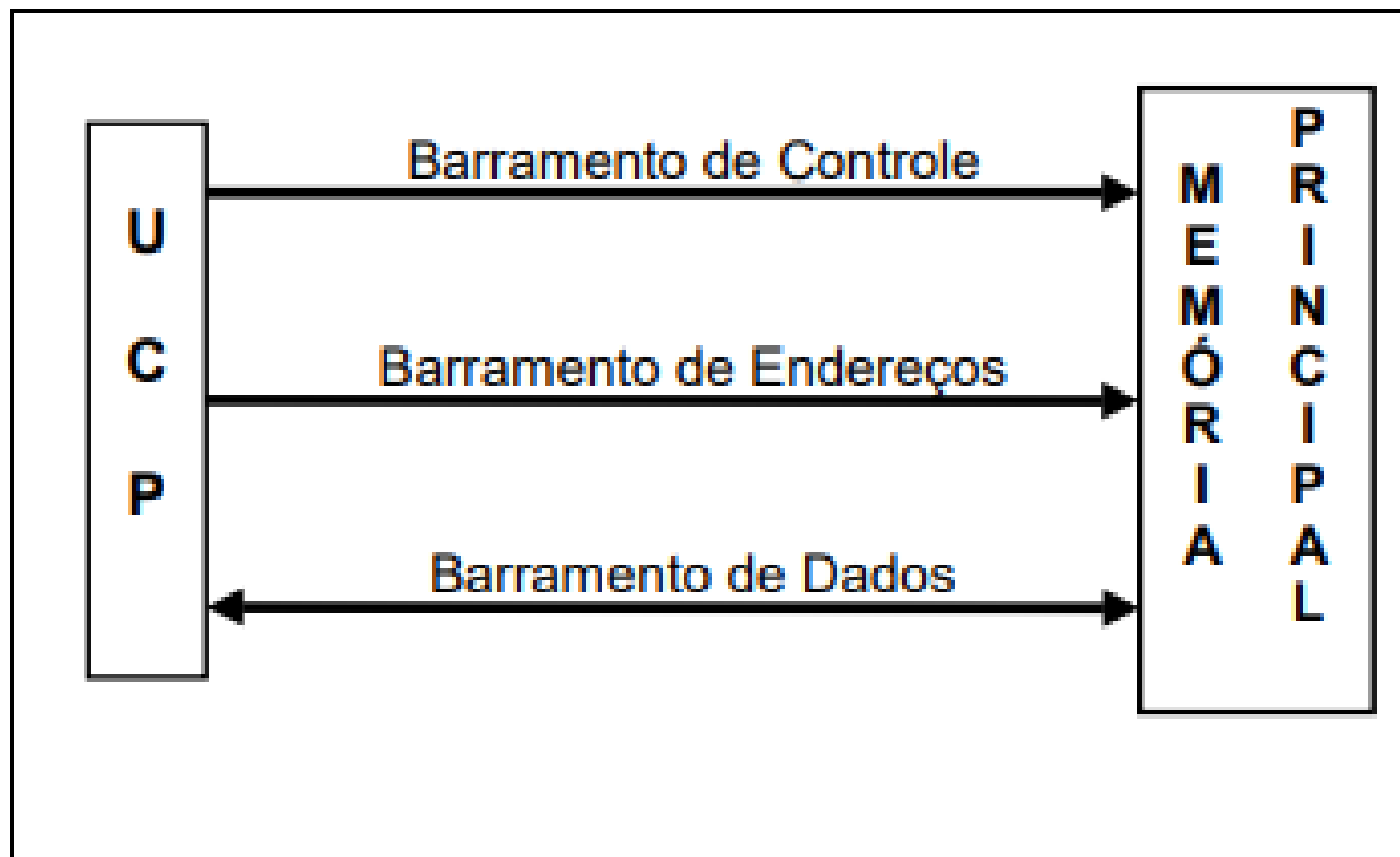
Dispositivos de entrada e saída



Barramento

- Linhas de comunicação entre a UCP, a MP e os dispositivos de E/S.
- De acordo com a transmissão, podem ser unidirecionais (um só sentido) ou bidirecionais (em dois sentidos).
- Exemplo: comunicação UCP / MP.
 - **Barramento de Dados:** transmite informações entre as unidades funcionais (**MBR**).
 - **Barramento de Endereços:** especifica o endereço da célula a ser acessada (**MAR**).
 - **Barramento de Controle:** sinais relativos a leitura/gravação.

Barramento



Inicialização do Sistema (Boot)

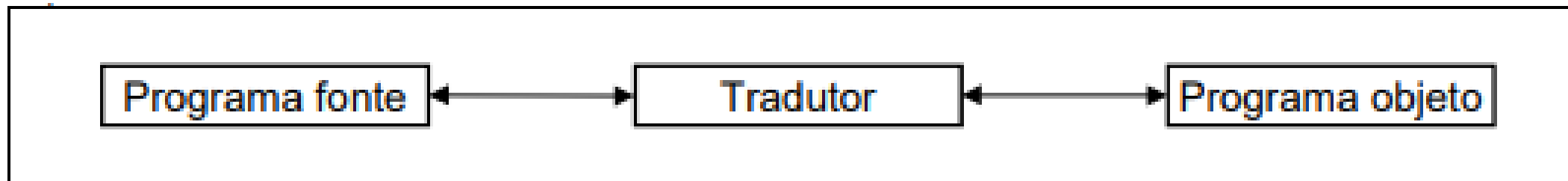
- SO é essencial pois disponibiliza a totalidade dos recursos disponíveis no equipamento.
- Ao ligar o computador é necessário que o SO seja transferido da memória secundária para a principal.
- Este processo é denominado boot e é realizado por um programa localizado em uma posição específica do disco (boot block, Master Boot Record).

Software

- Torna o hardware útil.
- Facilita o trabalho do usuário (conceito de camadas).

Tradutor

- Programas escritos em linguagem de alto nível
- Menor preocupação com aspectos de hardware (endereços de memória para carga e variáveis).
- Não executado diretamente, precisam ser “traduzidos” para um módulo chamado objeto (linguagem de máquina, mas ainda não executável na maioria dos casos, pois podem necessitar de rotinas externas).



Tradutor

Dividem-se em:

- **Montador:** linguagem de montagem (**mnemônicos**) associada à linguagem de máquina do processador. Ligada diretamente ao processador, diferente, portanto, para cada fabricante. (**linguagem assembly**).
- **Compilador:** linguagem de alto nível, independente de implementação (**portável**).

Interpretador

Traduz mas não gera código objeto.

- Traduz e executa a instrução logo a seguir.
- Tempo de execução elevado.
- Exemplos: **Basic e dBase.**

Linker

- Gera, a partir de um ou mais códigos objetos, um único código executável.
- Resolve referências externas utilizando-se de bibliotecas (módulos objetos ou definições de símbolos).
- Determina a região da memória onde o programa será carregado para ser executado (relocação).
- **Código absoluto:** os endereços são resolvidos em tempo de linkedição (inviável em ambientes multiprogramáveis).
- **Código relocável:** os endereços serão resolvidos no momento da carga.

Loader

- Ou carregador, coloca fisicamente o programa na memória.
- O procedimento varia de acordo com o código gerado pelo linker.
- Se absoluto: deve ser conhecido o endereço de memória inicial e o tamanho do módulo.
- Se relocável: pode ser carregado em qualquer lugar da memória.
- A execução é iniciada logo após a carga.

Depurador

- Utilizado para encontrar erros de lógica.
- Permite ao usuário:
 - Acompanhar a execução do programa instrução à instrução.
 - Verificar e alterar valores de variáveis.
 - Criar pontos de parada (**breakpoints**).
 - Determinar o envio de mensagens quando da alteração de uma variável (**watchpoints**).

Linguagem de Controle

- Forma mais direta de se comunicar com o SO.
- Comandos interpretados pelo **shell (interpretador de comandos)**.
- Sequência de comandos em um arquivo (**tipo batch**).

Linguagem de Máquina

- Linguagem que o computador realmente consegue entender.
- Codificada em **formato binário**.
- Programas longos com maiores chances de erros.
- **Conjunto de instruções é específico de cada processador**

Micro programação

- Para cada instrução em linguagem de máquina, existe um micro programa associado.
- Normalmente existem 25 microinstruções básicas interpretadas pelos circuitos eletrônicos.
- Computadores micro programáveis permitem a criação de novas instruções de máquina com novos micro programas.
- Em PC os micro programas estão gravados em ROM.

Exercício

1. Quais seriam as principais dificuldades que um programador teria no desenvolvimento de uma aplicação em um ambiente sem um sistema operacional?
2. Explique o conceito de máquina virtual. Qual a grande vantagem em utilizar esta metodologia?
3. Defina o conceito de uma máquina de camadas.
4. Explique a seguinte frase: “O Sistema Operacional protege o usuário da máquina e a máquina do usuário”.