

# Sistemas Operacionais

Aula 1 – Apresentação da Disciplina

Prof. Paulo

[paulo.garcia@fmu.br](mailto:paulo.garcia@fmu.br)

# Sistemas Operacionais



**Ethos**



Hive OS



**PiMP**

**UNIX®**  
**IBM SYSTEM/360**  
**MVS/380**

**IBM OS/390**

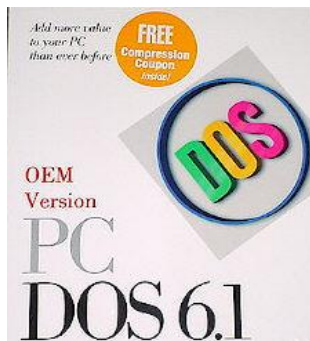
**XENIX**



debian



Mac OS



# Sistemas Operacionais

HD 7950 \$179, 280h/s on Bitcoin Gold  
Buy at [ethosdistro.com/7950](http://ethosdistro.com/7950)

Quick Start: [ethosdistro.com/Q](http://ethosdistro.com/Q)  
Support: [ethosdistro.com/S](http://ethosdistro.com/S)

run 'helpme' to get started, root/ethos password is 'live'  
toggle fullscreen terminal with ctrl+alt+left/right arrow

**ethOS**  
[ethosdistro.com](http://ethosdistro.com)



**MICROEJ**



**ARM**mbed

**Contiki**

**NUCLEUS**

**android**  
things



**snappy**

Blockgeeks



EOS



**OK ROKOS**  
core v8



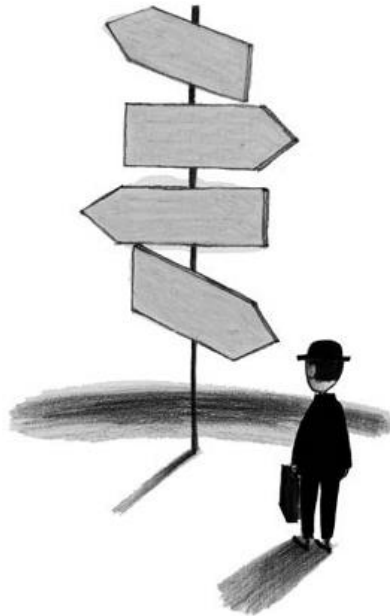
# Sistemas Operacionais

- Pauta
  - Planejamento Pedagógico (ementa, bibliografia, planejamento de ensino)
  - APS (Atividade Prática Supervisionada)
  - Parte 1
    - Introdução
    - História dos Sistemas Operacionais
    - Classificação de Sistemas operacionais
    - Tipos de Sistemas operacionais
  - Parte 2
    - Estrutura / arquitetura dos Sistemas Operacionais
    - Conceitos de Hardware
      - Dispositivos de E/S
      - Barramentos
    - Inicializando o Computador
    - Comunicação

# Sistemas Operacionais

## Ementa

- Aborda os **conceitos fundamentais de sistemas operacionais** contextualizados em cenários reais onde estes conceitos são utilizados, incluindo máquinas virtuais, *containers* e computação em nuvem. **Discute os princípios e os relacionamentos existentes entre os mecanismos de gerenciamento de processos, gerência de memória e entrada/saída e gerenciamento de arquivos.** Inclui também estudos de caso dos principais sistemas operacionais utilizados na atualidade.



**OBJETIVOS DE APRENDIZAGEM**

Analisar o funcionamento de Escalonamento em Sistemas Operacionais.  
Aplicar algoritmos de Alocação de Memória.  
Comparar o funcionamento do escalonamento e da alocação de memória com métodos diferentes.

**COMPETÊNCIAS RELACIONADAS**

III  
V  
VIII  
IX  
XVII

**ATIVIDADES A SEREM DESENVOLVIDAS**

As Atividades Práticas Supervisionadas - APS têm seu detalhamento publicado no ambiente virtual de aprendizagem (Blackboard) da disciplina. São publicadas na primeira quinzena de aulas e devem ser realizadas pelos estudantes até o limite do prazo da N1, em conformidade com o calendário acadêmico.

As APS devem ser realizadas pelos estudantes no próprio ambiente virtual de aprendizagem (Blackboard) ou ter seu upload realizado lá, onde também serão corrigidas pelo docente, ficando registradas em sua integralidade.

(i)

A fase **eficiente** consiste na utilização/escolha de uma das duas opções para cada equipe que são: um simulador para o escalonamento de processos e um simulador para alocação de memória. A equipe (máx. 3 pessoas) mostrará sua capacidade em entender e abstrair o problema dado, e com base em seus conhecimentos analisar o escalonamento e a alocação de memória e comparar os resultados. Após a Simulação, a equipe apresentará as soluções obtidas em um formato de artigo.

Obs: Nessa atividade o material sugerido aos alunos será artigos em Inglês sobre os respectivos assuntos.

(ii)

A fase **intensiva** consiste em realizar a elaboração de um artigo no formato da Sociedade Brasileira de Computação (SBC), com a intenção de relatar os resultados obtidos sobre Método de Escalonamento de Processos. Nessa fase, além das bibliografias recomendadas em classe, o aluno deve buscar referências científicas Nacionais sobre o assunto.

**AVALIAÇÃO**

A avaliação da APS será baseada nos princípios de autonomia pedagógica, feedback significativo e metacognição, culminando na autoavaliação do estudante. A nota da APS será atribuída no valor de 0,0 (zero) até 1,0 (um) ponto e vai compor a nota da A2, com base na rubrica de autoavaliação disponível no Ambiente Virtual de Aprendizagem. Só poderá realizar a autoavaliação o estudante que finalizar a atividade conforme instruções deste documento, postando-a até o dia solicitado pelo professor.



Mês	Dia	Conteúdo Previsto
Agosto	18	1 – CONCEITO E EVOLUÇÃO DE SISTEMAS OPERACIONAIS(SO)
	25	2 – Gerenciamento de Recursos: Conceitos Básicos e Aplicabilidade <b>** Postagem das Instruções da APS pelo Docente no AVA **</b>
Setembro	1	3 – Gerenciamento de Processos • Principais Conceitos sobre Processo • Diagrama de Execução/de Estados do Processo 4 – Gerenciamento de Processos • Usando linguagem de Programação • Simulação dos Processos
	8	5 – Gerenciamento de Processos • Threads • Modelos de Criação de Threads
	15	6 – Gerenciamento de Processos • Escalonamento (FIFO e Prioridade) • Troca de Contexto
	22	<b>Previsão de Lab</b>
	29	7 – Escalonamento de Processos • SJF-Preemptivo e Não-Preemptivo • Round Robin ou Alternância Circular
Outubro	6	8 – Comunicação e Sincronização de Processos • Programação Concorrente • Problemas de Compartilhamento de Recursos 9 – Comunicação e Sincronização de Processos • Deadlock • Algoritmo de Peterson/Semáforos e Monitores <b>Previsão de Lab</b>
	13	10 – Gerenciamento de Memória Real • Conceitos Básicos e Funcionalidade • Monoprogramação sem Trocas de Processos ou Paginação 11 – Gerenciamento de Memória • Com MAPA de BITS • Alocação de Segmentos Livres
	20	12 – Gerenciamento de Memória Virtual • Conceitos de Paginação • Implementação de Paginação <b>Previsão de Lab</b>
	27	13 – Gerenciamento de Memória Cache • Conceitos e Funcionalidade • Segmentação
Novembro	3	<b>Esclarecimento de dúvidas e Prova de Avaliação Continuada</b> <b>** Data Final para Postagem da APS pelo discente (Aluno) no AVA **</b>
	10	14 – Gerenciamento de Entrada e Saída e de Arquivos • Interrupção, Acesso Direto à Memória • Arquivos: Tipos, Sistemas e Acessos <b>** APS - Avaliação do docente e aplicação do Feedback coletivo em Sala de aula **</b>
	17	15 – Máquinas Virtuais, Computação em Nuvem - • Definição e Exemplos <b>** APS - Data Final para Autoavaliação do Aluno **</b>
	24	16 – Mecanismo de Segurança • Ambiente, controle de acesso aos recursos • Noções Básica de Criptografia
Dezembro	1	<b>Aplicação da Prova A2</b>
	8	<b>Vista de Provas A2</b>
	15	<b>Aplicação da Prova Substitutiva</b>

*(\*) Datas sujeitas a alteração em razão de mudanças do Calendário ou por replanejamento pelo Professor.*

# Sistemas Operacionais





# Sistemas Operacionais



- Reafirmando o que já é de conhecimento de todos:
  - Justificativas de faltas e solicitações de abonos de faltas devem ser encaminhadas para a Secretária seguindo as orientações no Manual do Aluno.
  - Observem os prazos e os canais para a entrega de justificativas de faltas.
  - O Professor não pode tirar e nem abonar faltas.
  - Ausência coletiva de alunos implica em falta para todos na lista de presenças.
  - Chegar mais tarde ou sair mais cedo da sala implica em falta na lista de presenças proporcional as aulas não assistidas no dia.
  - Atenção com a tolerância de 15 minutos!
  - Atenção com os Feriados!



## BÁSICA

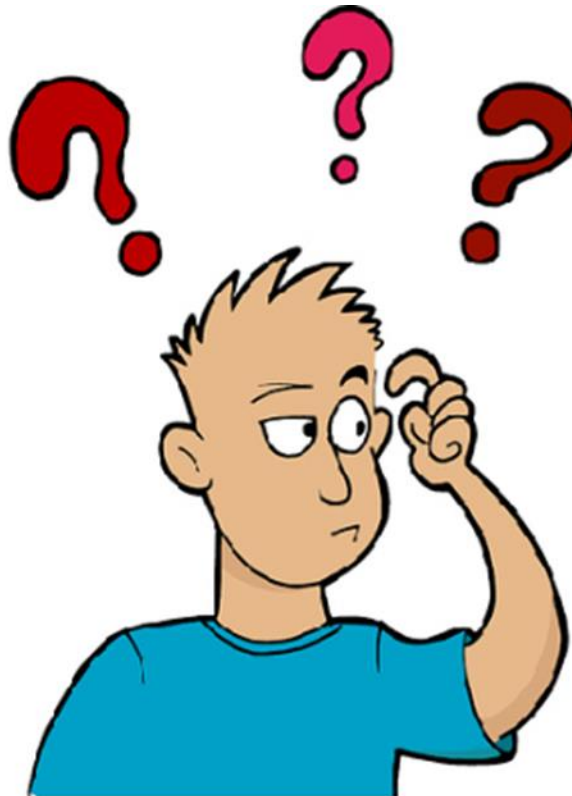
- |   |  |
|---|--|
| 1 | TANENBAUM, Andrew S. e BOS, Herbert. <b>Sistemas Operacionais Modernos</b> . 4. ed. São Paulo: Person Education do Brasil, 2016 (Disponível na Biblioteca Virtual -Person) |
| 2 | MACHADO, Francis B. e Maia, Luiz P. , <b>Arquitetura de Sistemas Operacionais</b> .5.ed.Rio de Janeiro:LTC,2103(Disponível na Biblioteca Virtual-Minha Biblioteca)         |
| 3 | NEMETH, Evi. <b>Manual Completo do Linux: Guia do Administrador</b> .2.ed.São Paulo,2007 (Disponível na Biblioteca Virtual -Person)  |

## COMPLEMENTAR

- |   |   |
|---|---|
| 1 | DEITEL, Paul J. e Deitel Harvey. <b>C: como programar</b> . 6.ed.São Paulo. Person Prentice Hall, 2011. (Disponível na Biblioteca Virtual- Person)  |
| 2 | DEITEL, Harvey e Deitel, Paul J. <b>JAVA: como programar</b> . 6.ed.São Paulo. Person Prentice Hall, 2011. (Disponível na Biblioteca Virtual- Person)   |
| 3 | DEITEL, H.M. <b>Sistemas Operacionais</b> ,3.ed.São Paulo.Person, 2005 (Disponível na Biblioteca Virtual - Person)  |
| 4 | OLIVEIRA, Rômulo Silva de . <b>Sistemas Operacionais</b> - Vol. 11 - 4ª ed.Porto Alegre, 2010 Bookman(Disponível na Biblioteca Virtual-Minha Biblioteca:<br><a href="https://integrada.minhabiblioteca.com.br/#/books/9788577806874">https://integrada.minhabiblioteca.com.br/#/books/9788577806874</a>                   |
| 5 | SILBERSCHATZ, Abraham e GALVIN, Peter B. . <b>Fundamentos de Sistemas Operacionais</b> . 9.ed.São Paulo, LTC (Disponível na Biblioteca Virtual-Minha Biblioteca:<br><a href="https://integrada.minhabiblioteca.com.br/#/books/978-85-216-3001-2">https://integrada.minhabiblioteca.com.br/#/books/978-85-216-3001-2</a> ) |

# Sistemas Operacionais

- DÚVIDAS



# Sistemas Operacionais

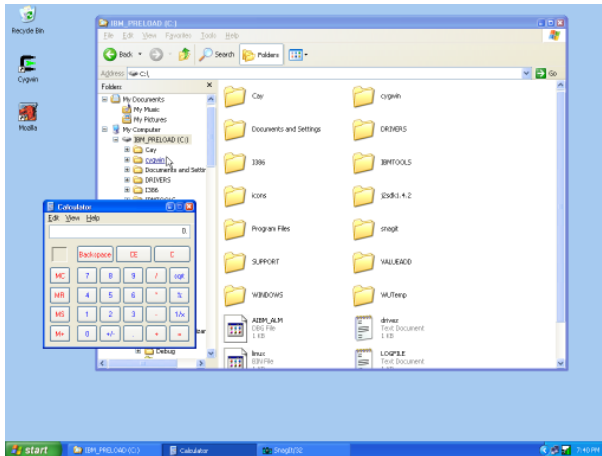
- Parte 1
  - Introdução
  - História dos Sistemas Operacionais
  - Classificação de Sistemas operacionais
  - Tipos de Sistemas operacionais
- Parte 2
  - Estrutura / arquitetura dos Sistemas Operacionais
  - Conceitos de Hardware
    - Dispositivos de E/S
    - Barramentos
    - Inicializando o Computador
    - Comunicação

# Sistemas Operacionais

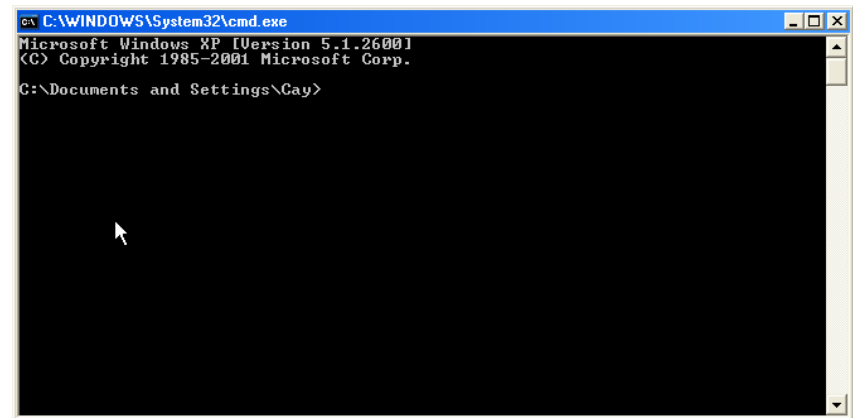
## Parte 1

# Introdução

- Introdução: Um sistema computacional é formado por componentes que o tornam complexo: (um ou mais processadores, memória, discos, impressoras, teclado, mouse, monitor, interfaces de rede e outros dispositivos de e/s, etc.)
- Como gerencia-los de forma otimizada?
- Como os programadores de software interagem com esses dispositivos?
- Como os usuários interagem com um sistema computacional?
- Interface Shell ou GUI (*Graphical User Interface*)? O que isso significa?



**GUI** – Interface amigável e baseada em ícones e menus. Comum em muitos SO's



**Shell** – Interface baseada em caracteres e textos. Dispensa ícones e menus. Usa mouse em situações limitadas. Não é uma interface para usuários inexperientes. Necessária em situações específicas.



# Introdução

- **Modo Núcleo ou Modo Supervisor**
  - Acesso completo ao hardware e executa qualquer instrução possível naquela máquina;
- **Modo Usuário**
  - Acesso parcial. Executa um subconjunto de instruções, em geral relacionadas ao controle da máquina ou a operações de e/s.
- **O que é um Sistema Operacional?**
  - É o software que executa em modo núcleo duas funções específicas:
    - Fornecer aos programadores de aplicativos (e a seus programas), um conjunto de recursos abstratos claros em vez de recursos confusos de hardware;
    - Gerenciar os recursos de Hardware



# História dos Sistemas Operacionais

- **Evolução dos Sistemas Operacionais**

- 1º Computador – Charles Babbage (1792 / 1871), Matemático Inglês – Por questões financeiras e técnicas, não pode materializar o seu modelo matemático. Não possuía um sistema operacional.

- **A Primeira Geração (1945-1955):**

- Válvulas e Painéis com *Plugs*



- **A Segunda Geração (1955 - 1965):**

- Transistores e Sistemas Batch ou em Lote (Off-line)
- Mainframes – Computadores de grande porte
- SO's: FMS – Fortran Monitor System e IBSYS SO da IBM (7094)

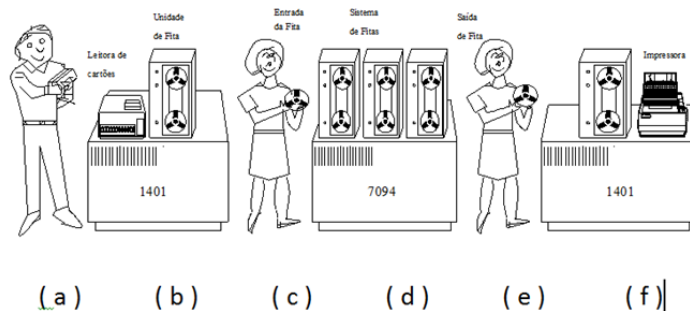
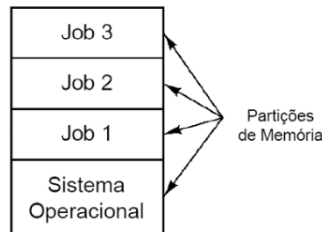


Figura 1 - Um sistema "batch" antigo. (a) Programadores levam cartões ao 1401. (b) 1401 lê batch de jobs em fita. (c) A operadora acopla fita de entrada no 7094. (d) O 7094 faz o processamento. (e) A operadora acopla fita de saída no 1401. (f) O 1401 imprime a saída.

# História dos Sistemas Operacionais

- **A Terceira Geração (1965 - 1980):**

- CI – Circuito Integrado
- Multiprogramação
  - Processamento Científico – Baixo I/O
  - Processamento Comercial – Alto I/O – 80 a 90% do tempo total de processamento
- Ociosidade do processador
  - Solução:



- Spooling
  - (*Spool – Simultaneous Peripheral Operation On Line*)
- Timesharing
  - Compartilhar através de terminais *on line*

- **Sistemas Operacionais:**

- CTSS – *Compatible Time Sharing System* – MIT
- MULTICS – *Multiplexed Information and Computing Service* – MIT, Bell Labs e GE
  - Ken Thompson – Versão + Simples e monouser do Multics que evoluiu para o UNIX

# História dos Sistemas Operacionais

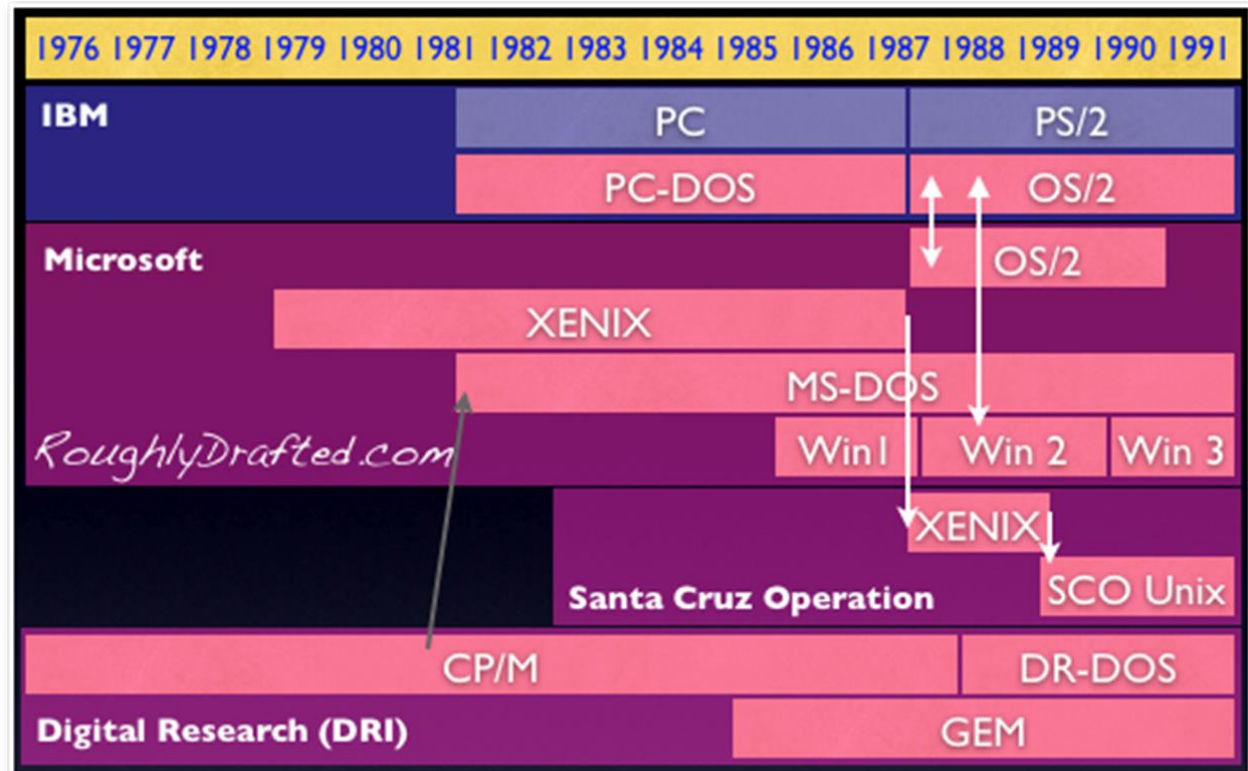
- **A Quarta Geração (1980):**
  - **Computadores Pessoais**
  - Desenvolvimento de circuitos LSI (*Large Scale Integration*). Chips contendo milhares de transistores em um centímetro quadrado de silício, a era do computador pessoal começava.
  - SO dominante: CP/M – *Control Program for Microcomputers* – Digital Reserch
    - Processadores Z80 (Zilog), 8080 (Intel)
    - Computadores Altair 8800, Osborne, TRS, Commodore, etc.
  - TRS-80, Apple II, IBM-PC



# História dos Sistemas Operacionais

- **Evolução de Sistemas Operacionais de Plataforma baixa**

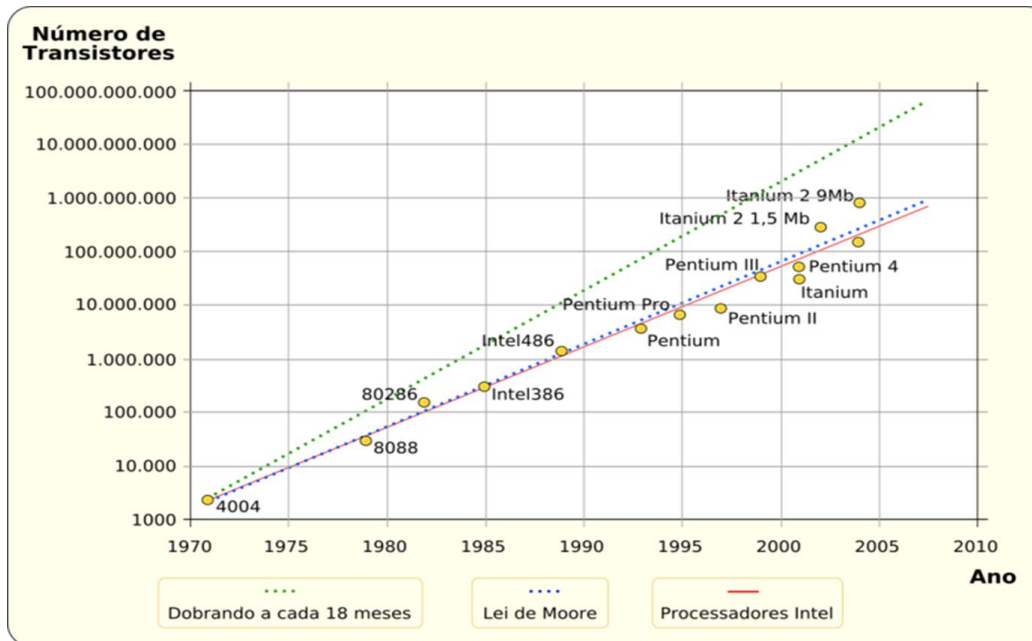
- **CP/M**
- **OS/2**
- **Xenix**
- **A era Windows**
  - MS-DOS
  - W98
  - WNT
- **O Unix**
- **O Linux**
- **SO Redes**
  - Novell Netware
  - Windows Server
  - Linux Server



# História dos Sistemas Operacionais

- **Lei de Moore**

- Dobrar a capacidade e manter preços a intervalos de 18 a 24 meses
- Se confirma nos últimos 50 anos
- Com a abundância de transistores na CPU, pode-se ir além das arquiteturas superescalares - *Multithreading* ou *Hyperthreading* ou *HT* (intel)
- CPU's *multi Core*





# Classificação de SO's

- **Sistemas Monotarefa x Sistemas Multitarefa**

- Quanto a capacidade de gerenciar mais de uma tarefa ao mesmo tempo
- **Monotarefa:**
  - Execução de um processo / tarefa por vez, com todos os recursos a disposição (CPU, Memória, periféricos, etc.), ocorrendo ociosidade / desperdício!
- **Multitarefa:**
  - Aceita e gerencia várias tarefas em processamento concorrente

- **Sistemas Monousuário x Sistemas Multiusuário**

- Quanto a capacidade de gerenciar mais de um usuário ao mesmo tempo, compartilhando os recursos de hardware e software.
- **Monousuário:**
  - Aceita e gerencia apenas um usuário. Não reconhece mais que um usuário conectado.
- **Multiusuário:**
  - Aceita e gerencia vários usuário. Permite mais que um usuário.
  - Sistema Multiusuário é Multitarefa, pois cada usuário tem ao menos uma tarefa

# Classificação de S.O's

- **Sistemas Monoprocessados x Sistemas Multiprocessados**
  - Quanto a capacidade de reconhecer e gerenciar computadores com mais de um processador.
  - **Monoprocessado:**
    - Reconhece e usa apenas um processador.
  - **Multiprocessado:**
    - Reconhece e usa mais de um processador.
    - Devem ser considerados sistemas operacionais para sistemas Distribuídos e de Multicomputadores.
- Os sistemas de computação distribuídos são adotados para tarefas de alto desempenho e podem ser divididos em duas classes distintas:
  - **Sistemas de computação de cluster (Homogêneos)**
  - **Sistemas de computação em grade (Heterogêneos)**

# Classificação de SO's

- **Cluster**

- A partir da redução dos preços dos computadores pessoais.
- Programa único é executado em paralelo.
- É um conjunto de máquinas conectadas por uma rede, no qual cada máquina executa um ou mais servidores. Estas máquinas estão conectadas por uma rede local, com alta largura de banda e baixa latência.
- **Exemplo:** Sistemas *Beowulf* baseado em Linux.

- **Grade**

- Sistemas Distribuídos organizados como uma federação de computadores, na qual cada sistema pode estar sob um sistema administrativo diferente, e podem ser muito diferente no que tange a hardware, software e tecnologia empregada.
- Alto grau de heterogeneidade.
- Software para permitir acesso a recursos (muitas vezes clusters) de diferentes organizações, reunidos para permitir a colaboração de um grupo.
- Foco na arquitetura, e divide-se em 04 camadas, em acordo com figura ao lado:



# Tipos de SO's

- Para *Mainframe*



- Para servidores



- Para multiprocessadores / Multicomputadores



- Para computadores pessoais



- Para computadores portáteis



- Embarcados



- Para nós Sensores



# Tipos de SO's

- De tempo real



- Para *Smart Cards*



- Para IoT



- Para moedas digitais



# Atividade extra classe

Elaborar pesquisa de um dos tipos de Sistema Operacional abaixo. Entrega na próxima aula.

- Para multiprocessadores / Multicomputadores



- Embarcados



- Para nós Sensores



- De tempo real



- Para IoT (*Internet of Things*)



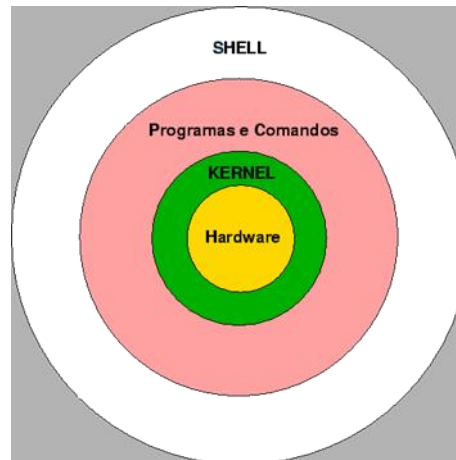


# Sistemas Operacionais

## Parte 2

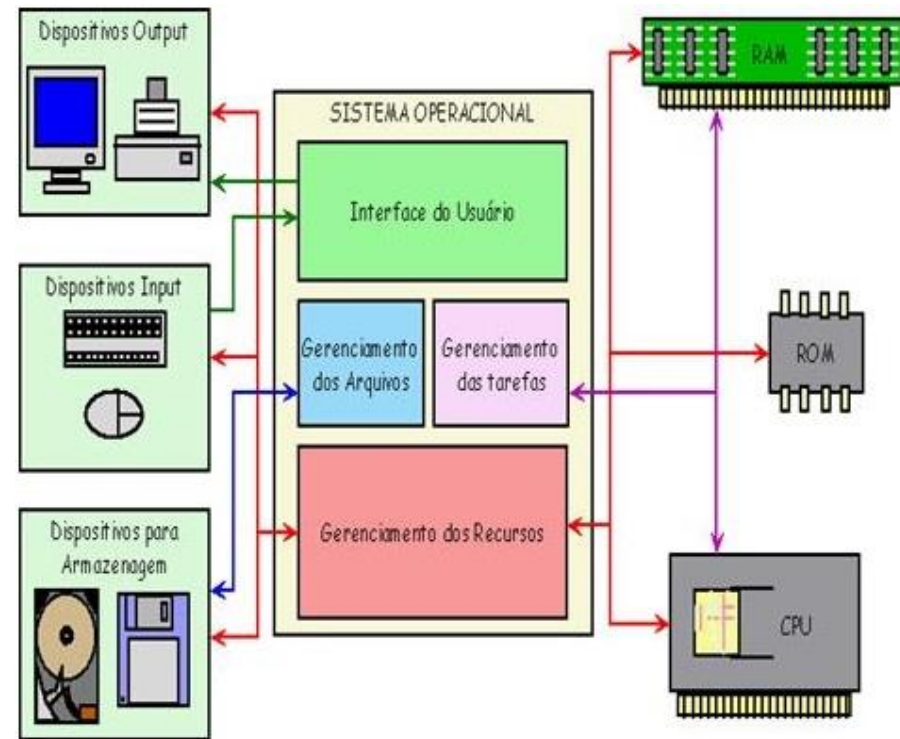
# Estrutura dos Sistemas Operacionais

- O *Kernel* ou Núcleo é uma parte do Sistema Operacional, oculto para os usuários. Face a sua importância para a operação de um computador, é mais seguro mantê-lo distantes de usuários inexperientes.
- Ele inicia quando o computador é ligado, com a detecção de todo o *hardware* indispensável ao funcionamento da máquina (monitor, placa de vídeo etc.). O SO é carregado e, assim que o usuário se loga, o *Kernel* passa a administrar as principais funções do S.O.: O gerenciamento da memória, dos processos, dos arquivos, de todos os periféricos, etc.
- O *Kernel* como um Maestro, é o responsável por garantir que todos os programas terão acesso aos recursos (p.e.: RAM, etc.) simultaneamente, permitindo um compartilhamento concorrente, mas sem oferecer riscos à integridade da máquina e de sua operação.



# Estrutura dos Sistemas Operacionais

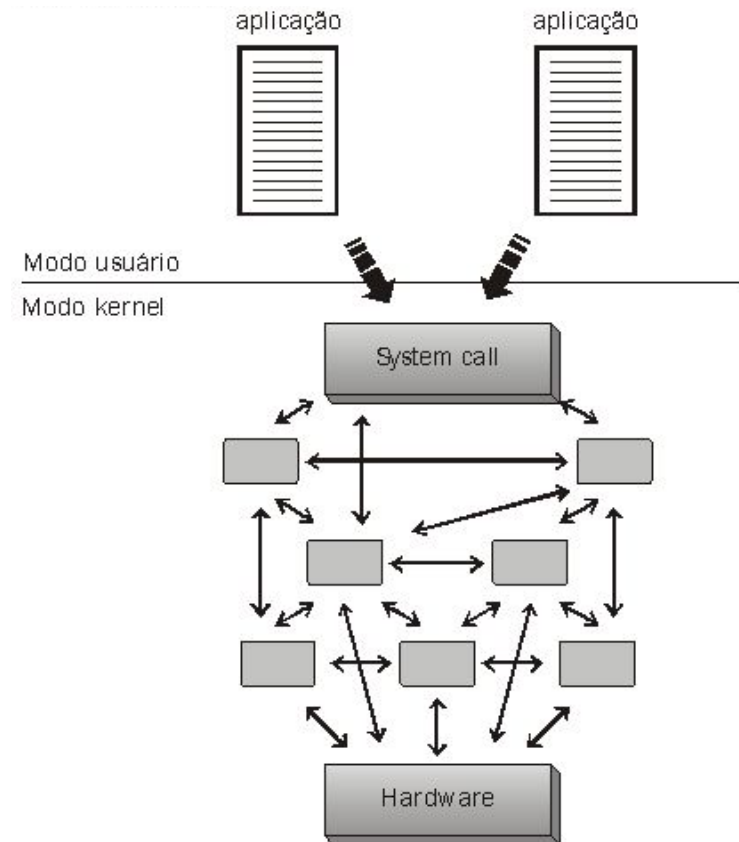
- Quais são os serviços encontrados no núcleo?
  - Tratamento das interrupções e exceções;
  - Criação e eliminação de processos e threads;
  - Sincronização e comunicação entre processos e threads;
  - Escalonamento e controle dos processos e threads;
  - Gerência de memória;
  - Gerência do sistema de arquivos;
  - Gerência dos dispositivos de entrada/saída;
  - Suporte a redes locais e distribuídas;
  - Contabilização do uso do sistema;
  - Auditoria e segurança do sistema.
- Muitos desses serviços são disponibilizados aos programadores através das *system calls*
- No Windows as *system calls* são chamadas de API – *Application Program Interface*
- System Calls* são implementadas através de interrupções de *software (traps)*



# Arquitetura de Sistemas Operacionais

## • Arquitetura Monolítica

- O sistema operacional é escrito como um conjunto de procedimentos, sendo que cada um pode chamar um dos demais sempre que necessário
- **Construção**
  - Compila-se todos os procedimentos com o objetivo de criar os módulos-objeto.
  - Juntam-se todos os módulos-objeto usando o *linker*, criando-se um único programa executável chamado *kernel*
- **Prós**
  - Desempenho
- **Contras**
  - Manutenção difícil. Tanenbaum considera como uma “grande bagunça”
- **Sistemas que a utilizavam**
  - MS-DOS e os baseados nele (windows até o 98/Me)
  - Primeiros sistemas UNIX



# Arquitetura de Sistemas Operacionais

## • Arquitetura em Camadas

- Com o aumento da complexidade e do tamanho do código dos sistemas operacionais, foram incorporadas técnicas de programação estruturada e modularizada.
- O sistema é dividido em níveis sobrepostos: cada camada oferece um conjunto de instruções que podem ser utilizadas pelas camadas superiores.

### • Prós

- O isolamento dos serviços do sistema operacional facilita a manutenção.
- A hierarquia protege as camadas mais internas.

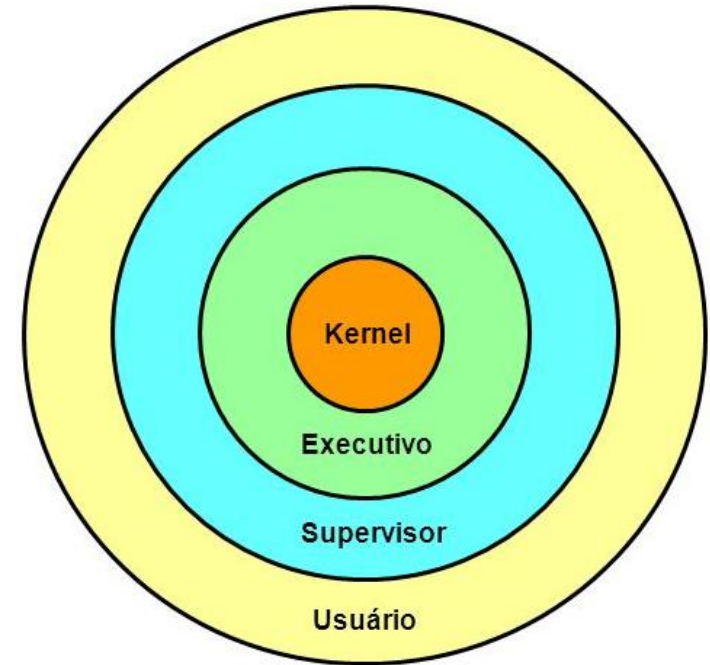
### • Contras

- Diminuição do desempenho por causa da quantidade de mudança de modos de acesso.

### • Sistemas que a utilizam:

- Maioria das versões do Windows 2000 em diante

- Maioria das versões do UNIX e LINUX



# Arquitetura de Sistemas Operacionais

## • Arquitetura Microkernel

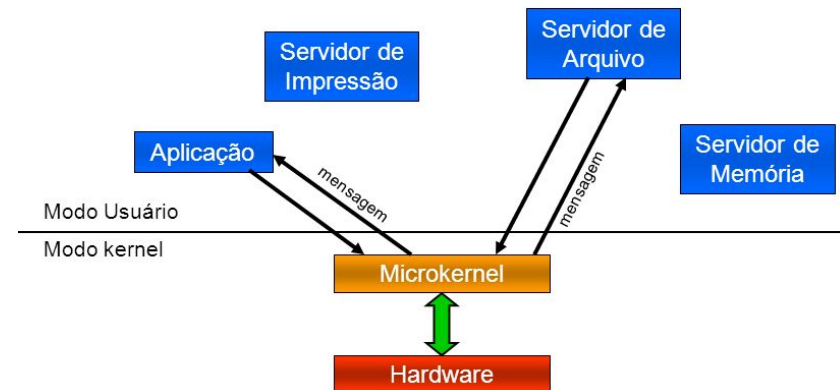
- É fruto do resultado da tendência de um núcleo de S.O. se tornar o menor possível;
- Os serviços são disponibilizados em processos;
- Cada processo é responsável em gerenciar um conjunto específico de funções como gerência de memória, gerência de arquivos, gerência de processos etc.;
- Dois tipos: processo cliente e processo servidor
- A principal função do núcleo é gerenciar a comunicação entre esses processos.

### • Prós

- Maior proteção do núcleo: todos os processos são executados em modo usuário;
- Alta disponibilidade: se um servidor falhar, o sistema não ficará altamente comprometido;
- Maior eficiência: a comunicação entre serviços poderá ser realizada entre vários processadores ou até mesmo várias máquinas distribuídas;
- Melhor confiabilidade e escalabilidade

### • Contras

- Grande complexidade para sua implementação
- Menor desempenho devido à necessidade de mudança de modo de acesso





# Arquitetura de Sistemas Operacionais

- **Arquitetura Máquina Virtual**

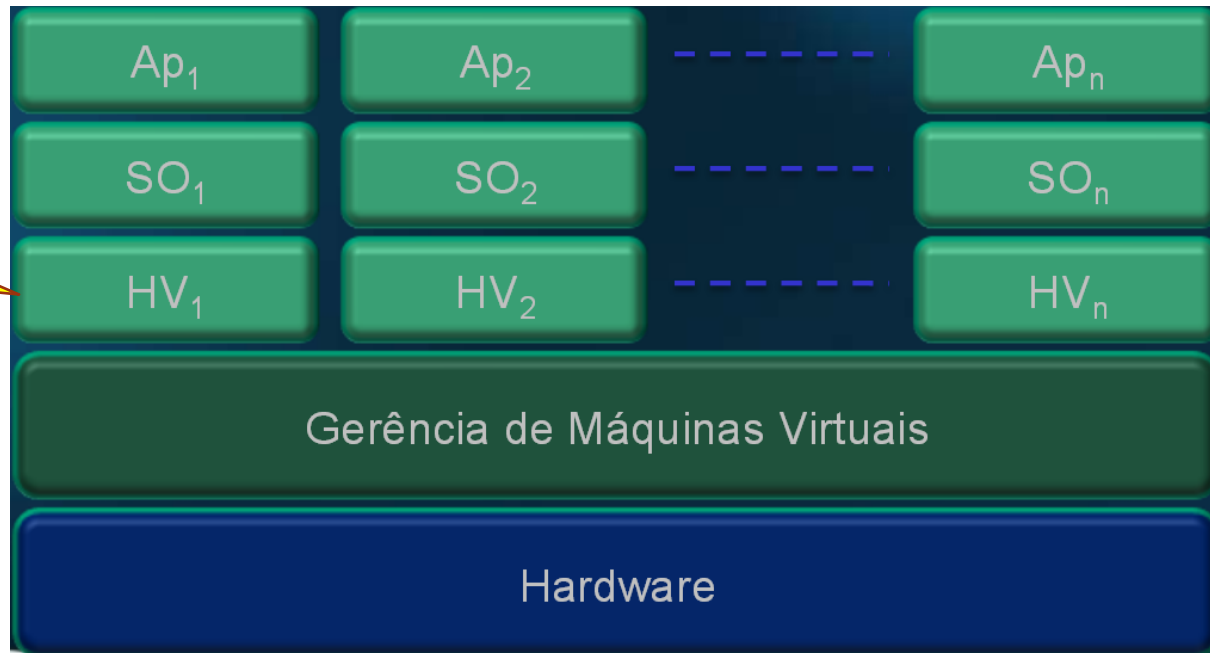
- O modelo de Máquina Virtual ou *Virtual Machine* (VM) cria um nível intermediário entre o *hardware* e o sistema operacional, denominado gerência de máquinas virtuais

- **Prós**

- Cria isolamento total de cada VM

- **Contras**

- Grande complexidade, devido à necessidade de compartilhar e gerenciar os recursos de hardware entre as diversas VMs.

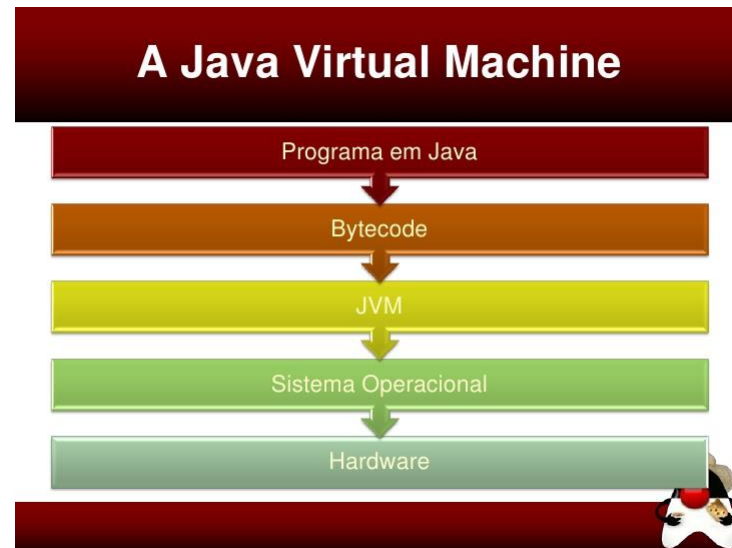


Cada uma oferece uma cópia do hardware

# Arquitetura de Sistemas Operacionais

- **Arquitetura Máquina Virtual**

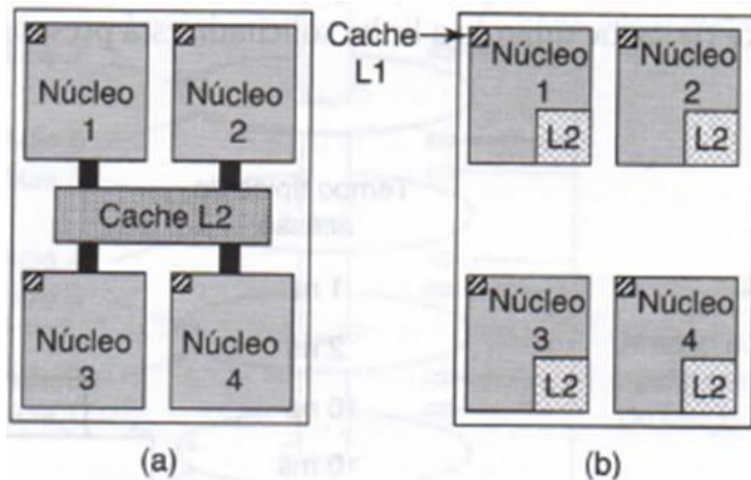
- Outro exemplo: Máquina Virtual Java (JVM).
- É utilizada de uma maneira um pouco diferente.
- Quando a *Sun Microsystems* inventou a linguagem Java, inventou também uma máquina virtual denominada de JVM.
- O código Java é executado pelo interpretador da JVM, que possui seu próprio conjunto de instruções.
- **Prós**
  - Portabilidade;
  - Código pode ser enviado pela Internet;
  - É independente do hardware e do sistema operacional.
- **Contras**
  - Menor desempenho se comparado com uma aplicação compilada.



# CONCEITOS DE HARDWARE

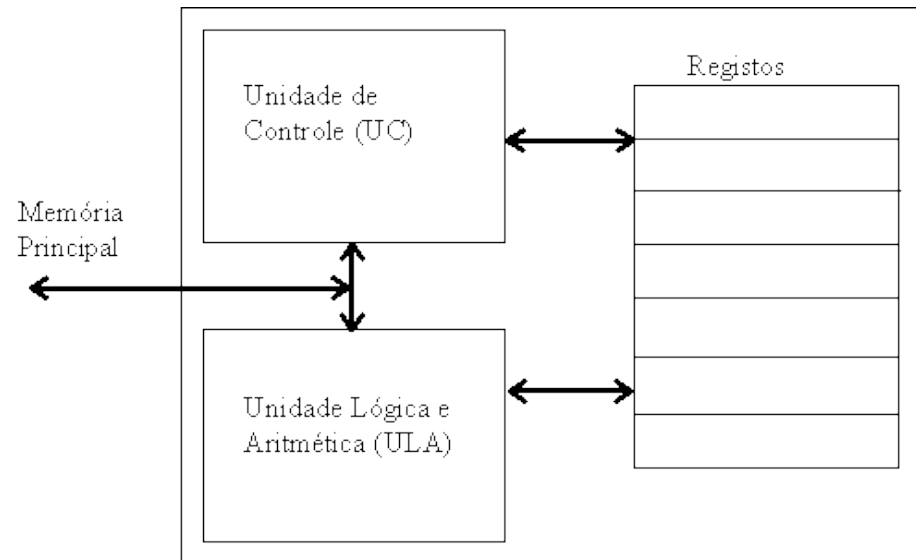
- **CPU**

- Também temos chips de CPU com dois, quatro ou mais processadores completos ou núcleos (core).



(a) Chip quad-core com uma cache L2 compartilhada

(b) Chip quad-core com caches L2 separadas



# CONCEITOS DE HARDWARE

- **HIERARQUIA TÍPICA DE MEMÓRIAS**

+ VELOCIDADE

- TAMANHO

+ CUSTO R\$



- VELOCIDADE

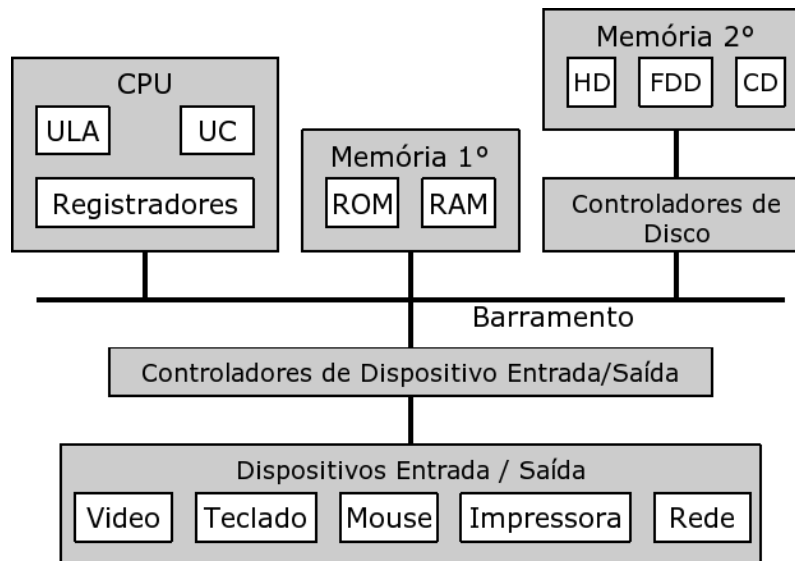
+ TAMANHO

- CUSTO R\$

# CONCEITOS DE HARDWARE

## ● DISPOSITIVOS DE ENTRADA E SAÍDA

- Os dispositivos de E/S envolvem um controlador e o próprio dispositivo.
- A interface entre o dispositivo e o SO será suprida pela Controladora.
- Esta interface é padronizada de forma a permitir, por exemplo, a conexão de um HD IDE/SATA de qualquer fabricante em qualquer sistema operacional.
- A controladora tem um software próprio, o **driver do dispositivo**, que pode variar para cada SO.



# CONCEITOS DE HARDWARE

- **DISPOSITIVOS DE ENTRADA E SAÍDA**

- Temos 3 formas de carregar os drives de dispositivos E/S no SO:
- A 1ª é instala-lo no núcleo e então reiniciar o SO (Unix);
- A 2ª é adicionar uma entrada a um arquivo do sistema operacional informando que ele precisa do driver e, então, reiniciar o sistema. No momento da inicialização, o sistema operacional busca e encontra os drivers de que ele precisa e os carrega. (Windows)
- A 3ª é capacitar o SO a aceitar novos drivers enquanto estiver em execução e instalá-los sem a necessidade de reinicializar. Esse modo está se tornando cada vez mais comum. Dispositivos acoplados a quente, como dispositivos USB e IEEE 1394 precisam sempre de drivers carregados dinamicamente.

# CONCEITOS DE HARDWARE

- **DISPOSITIVOS DE ENTRADA E SAÍDA**
- Os controladores possuem registradores:
  - Por exemplo, um controlador de discos deve ter, no mínimo, registradores para especificar endereços do disco e de memória, contador de setores e indicador de direção (leitura ou escrita), etc.
- Para atingir o controlador, o driver recebe um comando do SO (Interrupção) e o traduz em valores apropriados a serem escritos nos registradores de dispositivos.
- O grupo de todos esses registradores de dispositivos forma o espaço de porta de E/S.
- Interrupções são importantes para o SO

# CONCEITOS DE HARDWARE

## ● DISPOSITIVOS DE ENTRADA E SAÍDA

- A entrada e a saída podem ser realizadas das seguintes formas:
- **No método mais simples**, um programa de usuário emite uma chamada de sistema, a qual o núcleo traduz em uma chamada ao driver apropriado. O driver então inicia a E/S e fica em um loop perguntando continuamente se o dispositivo terminou a operação de E/S (em geral há um bit que indica se o dispositivo ainda está ocupado). Quando a operação de E/S termina, o driver põe os dados onde eles são necessários (se houver) e retorna.
- O sistema operacional então remete o controle para quem chamou. Esse método é chamado de **espera ocupada** e tem a desvantagem de manter a CPU ocupada interrogando o dispositivo até que a operação de E/S tenha terminado.
- **No segundo método**, o driver inicia o dispositivo e pede a ele que o interrompa quando terminar. Dessa maneira, ele retorna o controle da CPU ao sistema operacional. O sistema operacional então bloqueia, se necessário, o programa que o chamou pedindo o serviço e procura outra tarefa para executar. Quando o controlador detecta o final da transferência, ele gera uma **interrupção** para sinalizar o término.



# CONCEITOS DE HARDWARE

- **DISPOSITIVOS DE ENTRADA E SAÍDA**

- Os *dispositivos E/S* são os denominados *periféricos*.
- Permitem a interação do processador com o homem, possibilitando a entrada e/ou a saída de dados.
- Os *de entrada* codificam a informação que entra em dados que possam ser processados pelo computador.
- Os *de saída* decodificam os dados processados pelo computador em informação compreensível para o usuário.
- Há dispositivos que funcionam para entrada e saída de dados, como o modem e o drive de disquete. Outro dispositivo híbrido de dados é a rede de computadores.
- Exemplos: Monitor de vídeo, mouse, teclado, impressora, HD's, etc.

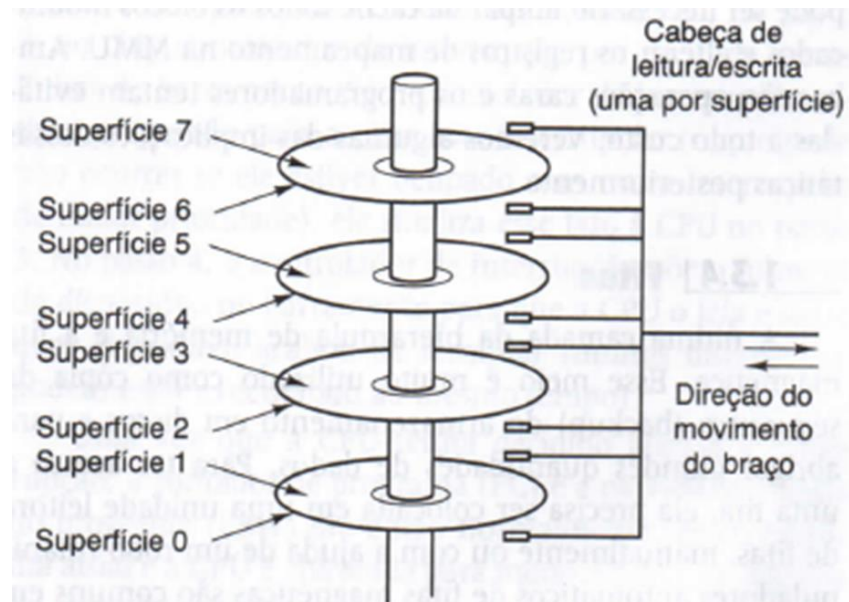
# CONCEITOS DE HARDWARE

## • DISPOSITIVOS DE ENTRADA E SAÍDA

### • HD (Hard Discs)

- Em relação a RAM: Muito maiores, Muito mais baratos, Muito mais lentos
- Mas, porque são mais lentos? Porque possuem partes mecânicas
- Existem discos com rotação de: 5.400, 7.200, 10.800 ou 15.000 RPM.
- Quanto maior a rotação ➡ melhor o desempenho do disco.
- Tecnologias disponíveis:

	IDE/ATA	SATA	SCSI	SAS
TRANSMISSÃO	Paralela	Serial	Serial	Serial
FLUXO	half-duplex	full-duplex	full-duplex	full-duplex
TAXA DE TRANSMISSÃO	133 MB/s	600 MB/s	640 MB/s	375 MB/s
FREQUENCIA	66 MHz	6,0 GHz	160 MHz	3,0 GHz
COMPRIMENTO CABO	0,46 m	8,00 m	12,00 m	8,00 m
HOT PLUG	Não	SIM	SIM	SIM
QTDE DISPOSITIVOS	2	1	16	4
PINAGEM	40/80	7	60/80	32
CONSUMO	5V	250mV	5V	800mV



# CONCEITOS DE HARDWARE

- **DISPOSITIVOS DE ENTRADA E SAÍDA**

- **HD (Hard Discs)**

- Um disco de estado sólido (*SSD - solid state disk*) que encaixa diretamente num slot de memória RAM DDR3 que estiver sobrando. Os contatos do slot de memória servem para a alimentação elétrica do SSD, para a comunicação é usado um cabo SATA que se conecta no centro do módulo. Sua velocidade de leitura chega a 700MB/s e a de gravação até a 250MB/s.
- Um disco de estado sólido (*solid state disk*) ainda tem o peso menor em relação aos discos rígidos, mesmo os mais portáteis; possui um consumo reduzido de energia; consegue trabalhar em ambientes mais quentes do que os HDs (cerca de 70°C); e, por fim, realiza leituras e gravações de forma mais rápida, com dispositivos apresentando 250 MB/s na gravação e 700 MB/s na leitura.



# CONCEITOS DE HARDWARE

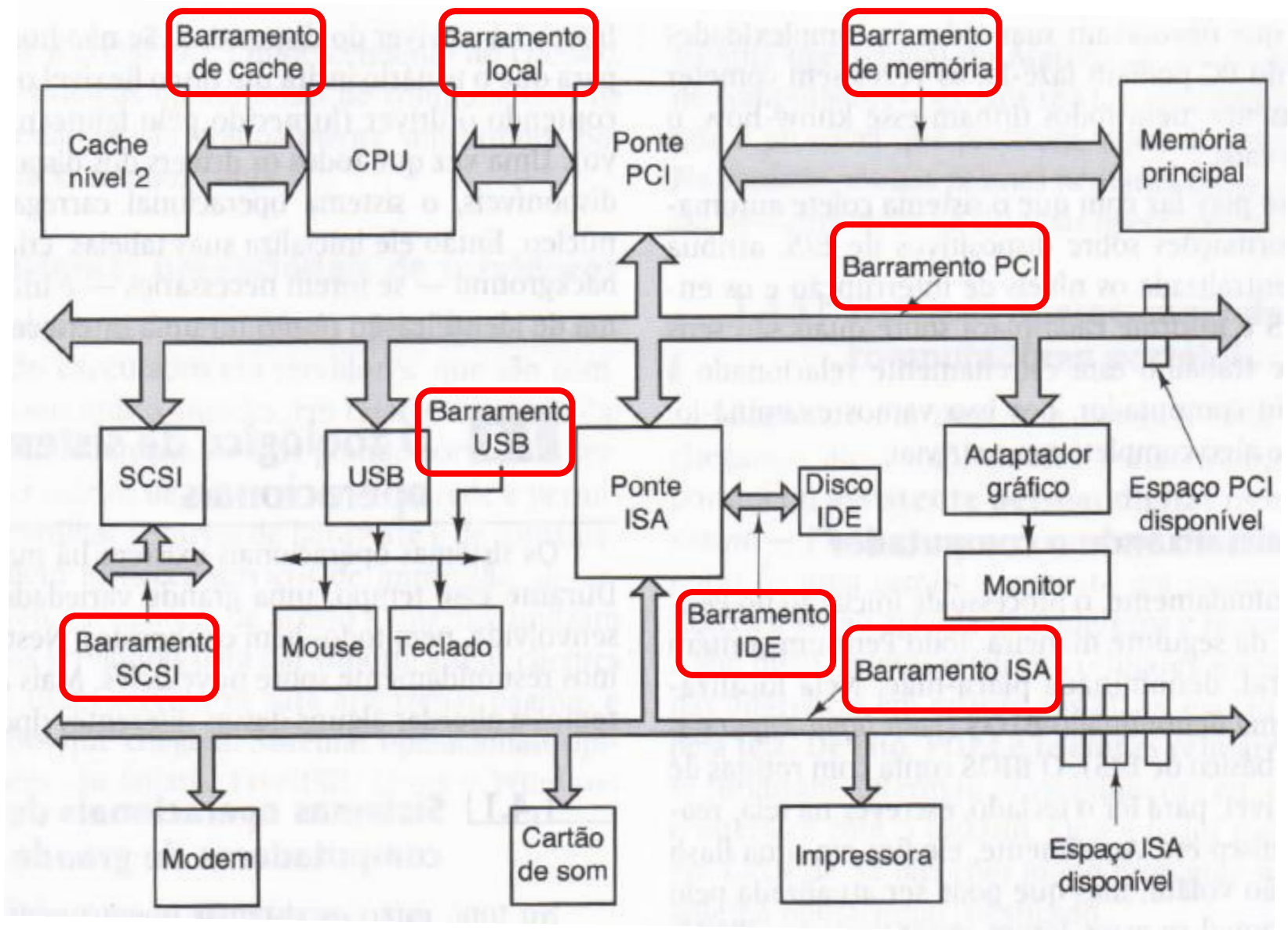
- **DISPOSITIVOS DE ENTRADA E SAÍDA**
- **Interfaces mais comuns:**
  - PS2, RS232C, Paralela / Centronics
  - USB, RGB , HDMI, Firewire, Etc.
  - IDE, SAS, Serial ATA , etc. (Não importa a tecnologia)
- **Fitas**
  - Mídia utilizada essencialmente para cópias de segurança (backup);
  - Muito lenta pois exige leitura sequencial;
  - Custo muito baixo e é removível;
  - Utiliza unidades manuais ou robotizadas para cópia ou leitura.

# CONCEITOS DE HARDWARE

- **BARRAMENTO**

- Barramento único – minicomputadores e no IBM PC Original;
- Aumento da velocidade dos processadores e memórias exigiu mais barramentos para tratar o tráfego de dados;
- Foram estabelecidos oito barramentos :
  - Cache
  - Local
  - Memória,
  - PCI – *Peripheral Component Interconnect*
  - SCSI / SATA - *Small Computer System Interface / Serial Advanced Technology Attachment*
  - USB - *Universal Serial Bus*
  - IDE - *Integrated Drive Electronics*
  - ISA - *Industry Standard Architecture (Barramento original do PC-IBM)*

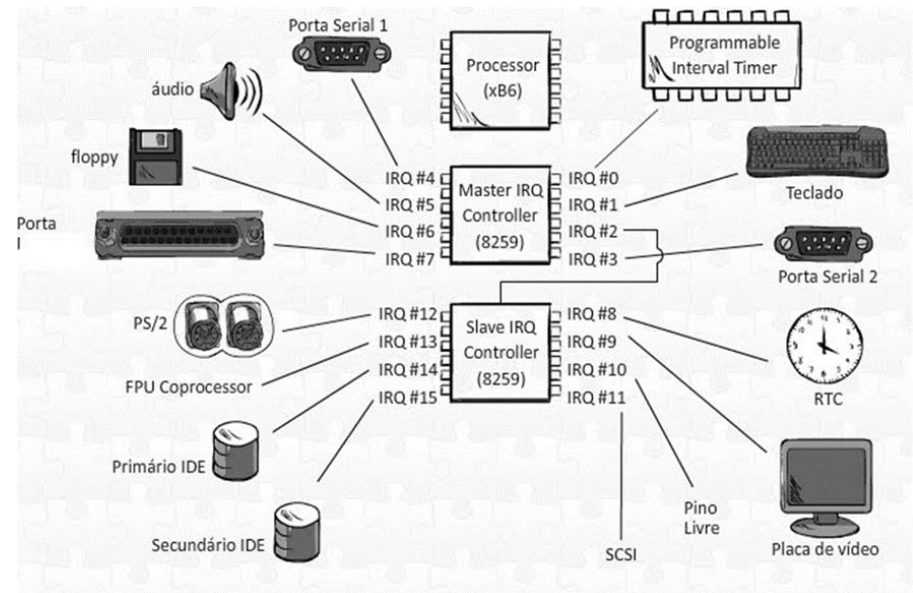
# CONCEITOS DE HARDWARE - Barramento





**● BARRAMENTO**

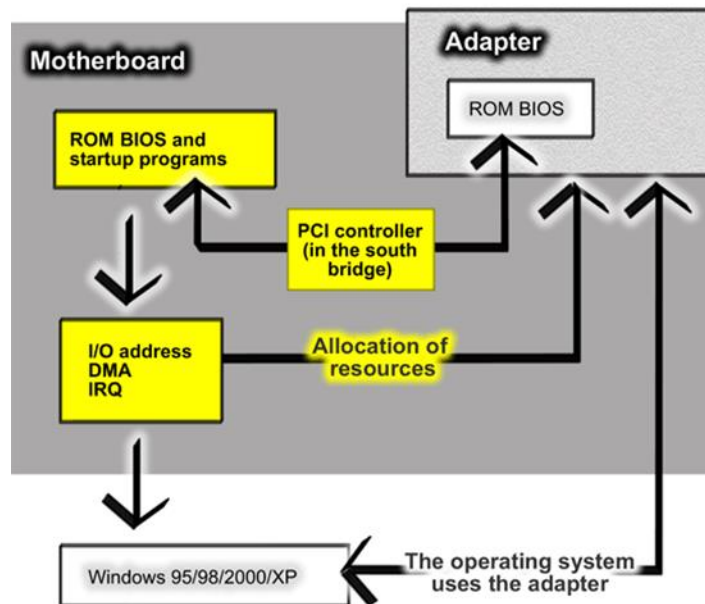
- Configuration Settings for the AT-GPIB/TNT**
- File Help
- Your AT-GPIB/TNT has been assigned Base I/O Address 0x280, No Interrupt, and DMA Channel 7. Use the illustrations below to configure your board to use these resources.
- 
- DRQ7  
DACK7  
DRQ6  
DACK6  
DRQ5  
DACK5  
IRQ15  
IRQ14  
IRQ12  
IRQ11  
IRQ10
- IRQ3  
IRQ4  
IRQ5  
IRQ6  
IRQ7  
IRQ9



# CONCEITOS DE HARDWARE

- **BARRAMENTO**

- O *Plug and Play* faz com que o sistema colete automaticamente informações sobre dispositivos de E/S, atribua de maneira centralizada os níveis de interrupção e os endereços de E/S e informe cada placa sobre quais são seus números.
- Esse trabalho está estreitamente relacionado à inicialização do computador





# CONCEITOS DE HARDWARE

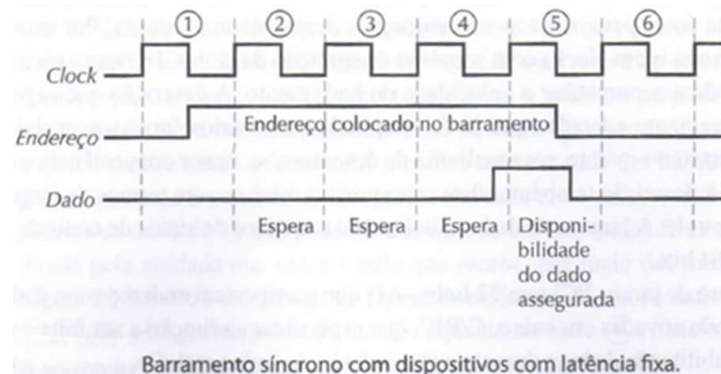
## ● INICIALIZANDO O COMPUTADOR

- Na placa-mãe temos um programa denominado **BIOS** (*basic input output system*). Ele conta com rotinas de E/S de baixo nível, para ler o teclado, escrever na tela, realizar a E/S no disco etc. Instalado em uma *flash* RAM, que é não volátil, mas que pode ser atualizada pelo usuário ou pelo sistema operacional.
- Quando o computador é ligado, o BIOS é iniciado e começa a executar uma verificação dos recursos do equipamento (Físico x Configurações existentes):
  - Memória RAM, Teclado, Mouse, etc.
  - Barramento ISA tem periféricos conectados?
  - Barramento PCI tem equipamentos conectados?
  - Existem dispositivos legados (antes do *Plug and Play*)?
  - Qual a sequência de carga do SO (HD, CD, FD, ROM)?
  - Não existindo problemas, inicializa o SO (Windows, Linux, etc.)
  - O SO checa na BIOS se os dispositivos conectados tem driver instalado, se não tiver pede ao usuário para informar onde busca-lo.

# CONCEITOS DE HARDWARE

## ● COMUNICAÇÃO NO BARRAMENTO

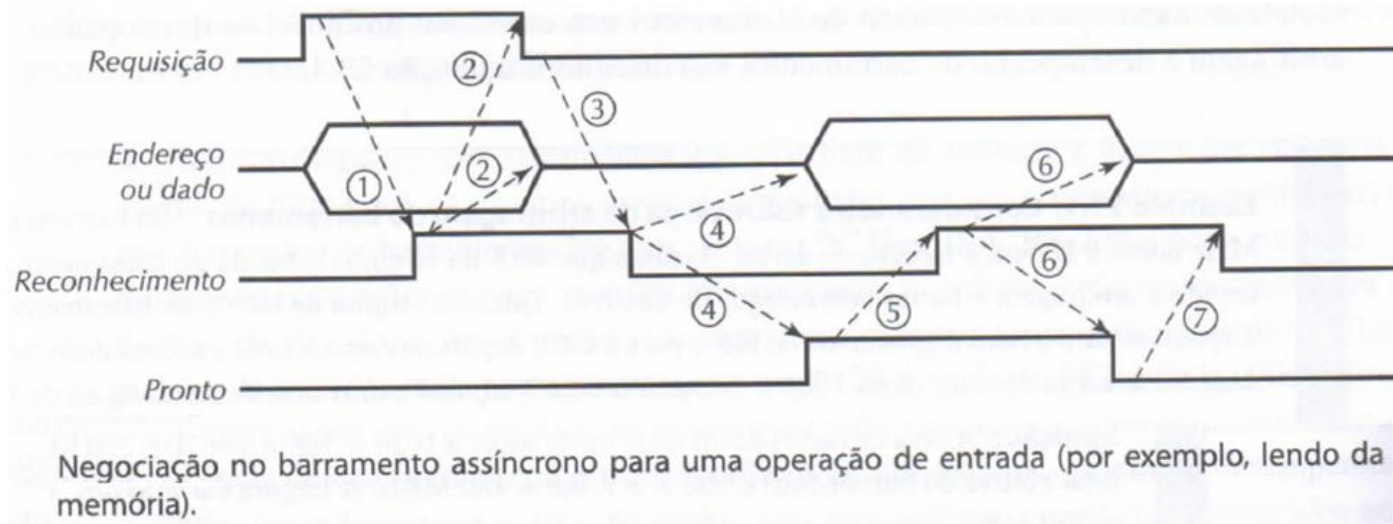
- Deve existir um protocolo de comunicação.
- O protocolo dependerá do tipo do barramento. **Síncronos** ou **Assíncronos**.
- Barramento Síncrono: um sinal de *clock* é parte do barramento e os eventos ocorrem em ciclos específicos do clock de acordo com um escalonamento pré-definido (*protocolo do barramento*).
- Por exemplo, um endereço de memória pode ser colocado no barramento em um ciclo, com a memória precisando responder com o dado no quinto ciclo de *clock*.
- Barramentos síncronos são adequados para um n° pequeno de dispositivos de velocidades iguais ou comparáveis comunicando-se a curtas distâncias (do contrário a carga do barramento e a distorção do *clock* se tomam problemáticas).



# CONCEITOS DE HARDWARE

- **COMUNICAÇÃO NO BARRAMENTO**

- Barramentos assíncronos podem acomodar dispositivos de velocidades variadas comunicando a longas distâncias, dado que a temporização fixa do barramento síncrono é substituída por um protocolo de negociação.
- Uma sequência típica de eventos em um barramento assíncrono no caso de entrada de dados, ou acesso de leitura na memória é mostrado na figura abaixo.



# Sistemas Operacionais

- DÚVIDAS

