

Engenharia de Computação



Arquitetura de Sistemas Operacionais

Introdução

Prof. Anderson Luiz Fernandes Perez

Prof. Martín Vigil

Universidade Federal de Santa Catarina

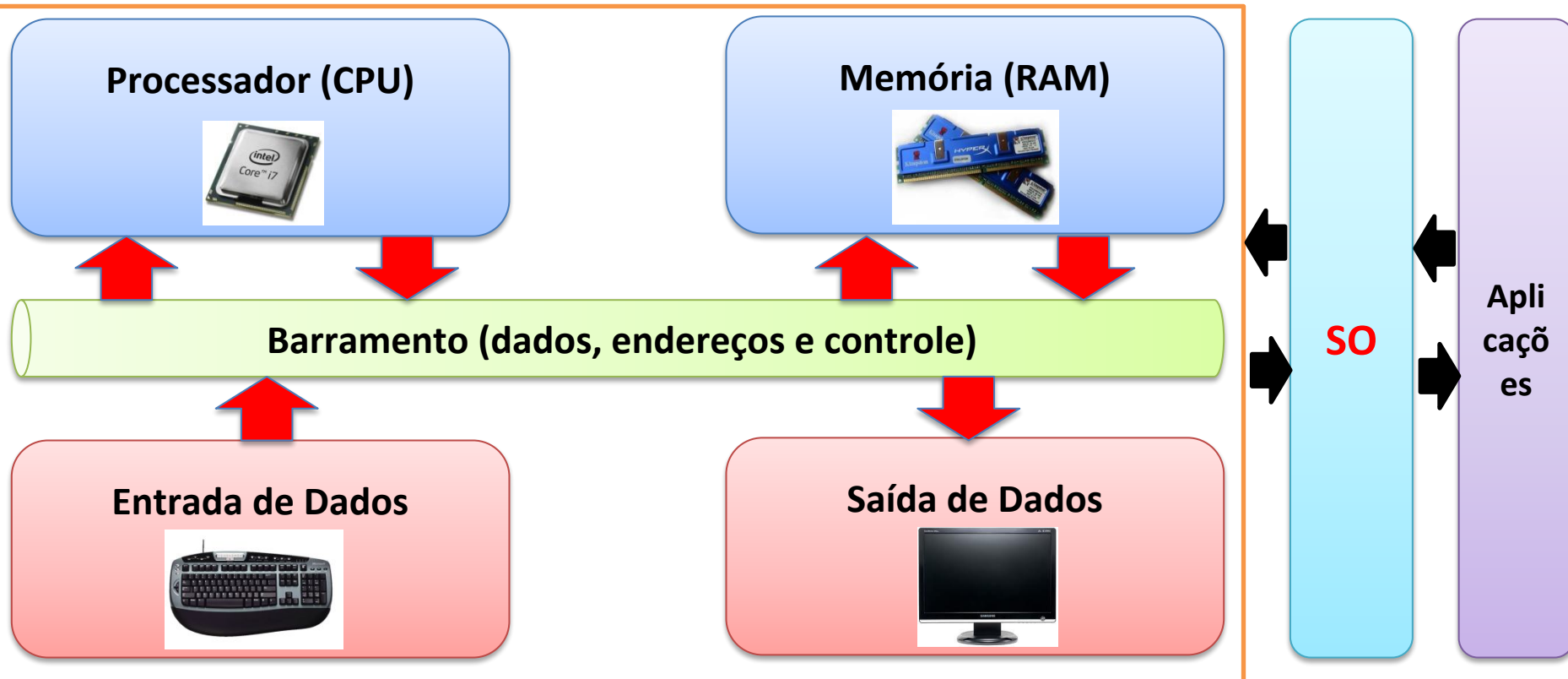
Campus Araranguá

Conteúdo

- Definição de Sistema Operacional
- Funções do Sistema Operacional
- Responsabilidade de um Sistema Operacional
- Arquitetura Interna de um Sistema Operacional
- Tipos de Sistemas Operacionais
- Chamadas de Sistema
- Histórico

Definição de Sistema Operacional

- Sistema Computacional



Definição de Sistema Operacional

- O que é um SO?
 - Um SO é um **software que controla diretamente o hardware e fornece uma infraestrutura para outros softwares.**
 - É o **software de base** da qual a maioria das aplicações depende.
 - **Gerencia o compartilhamento de recursos** entre entidades concorrentes.
 - **Fornece vários serviços comuns** que tornam as aplicações mais fáceis de escrever.

Definição de Sistema Operacional

- O que é um SO?

Um Sistema Operacional é composto de um ou mais programas que fornece um conjunto de serviços, o qual cria uma interface entre aplicações e o hardware do computador e que aloca e gerencia recursos compartilhados entre múltiplos processos.

Funções do Sistema Operacional



- Gerenciador de Recursos
 - O SO é responsável por todo o gerenciamento do hardware do sistema computacional.
 - O SO recebe solicitações de acesso a recursos por parte das aplicações e concede ou nega tais acessos.
 - Ao liberar o acesso a algum recurso, o SO deve alocar com cuidado o recurso, evitando que os programas não interfiram uns com os outros.

Funções do Sistema Operacional



- Provedor de Serviços
 - É fundamental que o SO ofereça uma gama de serviços que facilite as tarefas da aplicação.
 - A maior parte dos detalhes sobre acesso a dispositivos de E/S (Entrada e Saída), alocação de memória e similares seja cuidada pelo SO.
 - Sob a ótica de um provedor de serviços é possível afirmar que os programas de usuários são executados sobre o SO.

Funções do Sistema Operacional



- Abstrações
 - O objetivo do SO é fornecer uma **abstração do hardware**, tornando **transparente para as aplicações de usuário** o tipo e as características do hardware disponível.
 - Para as aplicações de usuário (programas) a **combinação de hardware e SO é um computador** no qual a aplicação é executada.

Responsabilidade de um Sistema Operacional



- Um sistema operacional deve lidar com um série de recursos e para cada um aplicar **técnicas de gerenciamento específicas**.
- São **responsabilidade de um SO**:
 - Gerenciar processos;
 - Gerenciar a memória;
 - Gerenciar os dispositivos de E/S;
 - Prover um sistema de arquivos;
 - Controlar a segurança do sistema computacional;
 - Prover um mecanismo de comunicação (rede);
 - Prover um interface com o usuário.

Responsabilidade de um Sistema Operacional



- Gerenciar Processos
 - O principal recurso a ser gerenciado pelo SO é a CPU.
 - Processos (programas de usuário em execução) estão constantemente solicitando acesso a CPU.
 - Um SO, em geral, **fornece serviços para**:
 - Criar processos;
 - Destruir processos;
 - Alterar a prioridade de processos;
 - Comunicação entre processos;
 - Sincronização entre processos concorrentes.
 - O SO é responsável pela escolha de qual programa entrará em execução (**escalonamento de processos**) e qual processo deixará de ser executado (**troca de contexto**).

Responsabilidade de um Sistema Operacional



- Gerenciar a Memória
 - O gerenciamento de memória tem a ver com os pedidos de alocação e liberação de memória feitos pelos programas de usuário (processos).
 - O SO deve assegurar que os processos não interfiram uns nos outros e que não haja desperdício de memória.
 - Os serviços típicos oferecidos pelo gerenciador de memória são:
 - Solicitar memória adicional diretamente;
 - Solicitar memória indiretamente (quando se cria um novo processo);
 - Liberar memória (devolvê-la ao SO);
 - Solicitar áreas de memória para serem compartilhadas por mais de um processo.

Responsabilidade de um Sistema Operacional



- Gerenciar os Dispositivos de E/S
 - Uma das facilidades em se utilizar um SO é no desenvolvimento de novas aplicações.
 - O SO “esconde” os detalhes do hardware, fazendo com que o programador se preocupe somente com a lógica do negócio.
 - O serviços de gerenciamento de dispositivos de E/S fornecidos por um SO são:
 - Abrir um dispositivo e associá-lo a um processo;
 - Ler e gravar dados de/para um dispositivo;
 - Fechar e liberar um dispositivo;
 - Fornecer acesso exclusivo a um dispositivo.

Responsabilidade de um Sistema Operacional



- Prover um Sistema de Arquivos
 - A maioria das aplicações de **usuário necessitam armazenar dados de maneira persistente**, sendo assim o SO é responsável por prover um interface para armazenar e recuperar dados.
 - Os **serviços oferecidos por um sistema de arquivos** são:
 - Abrir e fechar um arquivo;
 - Ler um arquivo;
 - Gravar dados em um arquivo;
 - Procurar por um lugar aleatório dentro de um arquivo;
 - Modificar os arquivos.

Responsabilidade de um Sistema Operacional



- Controlar a Segurança do Sistema Computacional
 - Um SO deve prover mecanismos de segurança para evitar, entre outras, que:
 - processo encerre a execução de outro;
 - processos acessem dispositivos de E/S sem permissão para tal.
 - O SO deve agir como um porteiro, ou seja, deve autorizar ou não que um processo acesse qualquer um de seus subsistemas.

Responsabilidade de um Sistema Operacional



- Prover um Mecanismo de Comunicação (rede)
 - É comum que aplicações de usuário acessem a rede para trocar dados com outras aplicações.
 - O SO deve fornecer um serviço de protocolos para permitir que as aplicações se comuniquem.
 - Um **serviço de comunicação de dados deve permitir as seguintes operações:**
 - Estabelecer uma conexão com um serviço remoto;
 - Atender conexões de um cliente remoto;
 - Enviar e receber mensagens para/de um sistema remoto;
 - Fechar uma conexão com um sistema remoto.

Responsabilidade de um Sistema Operacional



- Prover um Interface com o Usuário
 - Um SO deve fornecer, quando necessário, uma interface amigável com o usuário.
 - Um SO projetado para um ambiente desktop que não tenha um boa interface gráfica terá seu uso limitado a um grupo específico de usuários.

Arquitetura Interna de um Sistema Operacional

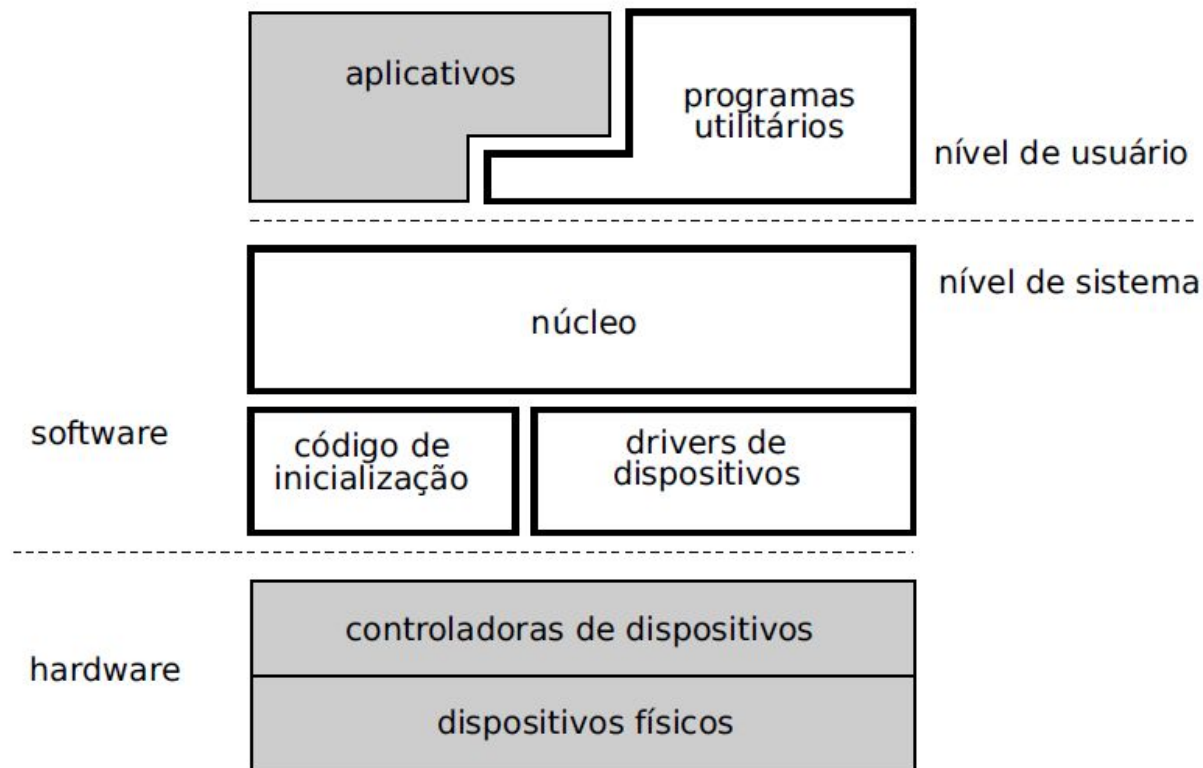


- Um SO é composto por diversos componentes com objetivos e funcionalidades complementares.
- Os componentes mais comuns presentes em um SO são:
 - Kernel (núcleo)
 - Drivers
 - Código de inicialização
 - Programas utilitários

Arquitetura Interna de um Sistema Operacional



- Integração entre os Componentes de um SO



Fonte: Maziero, 2010.

Arquitetura Interna de um Sistema Operacional



- Um SO pode ser organizado de diferentes formas, ou seja, as partes que o compõem podem ser implementadas da seguinte maneira:
 - Monolítico
 - Em Camadas
 - Micronúcleo (microkernel)
 - Máquinas Virtuais

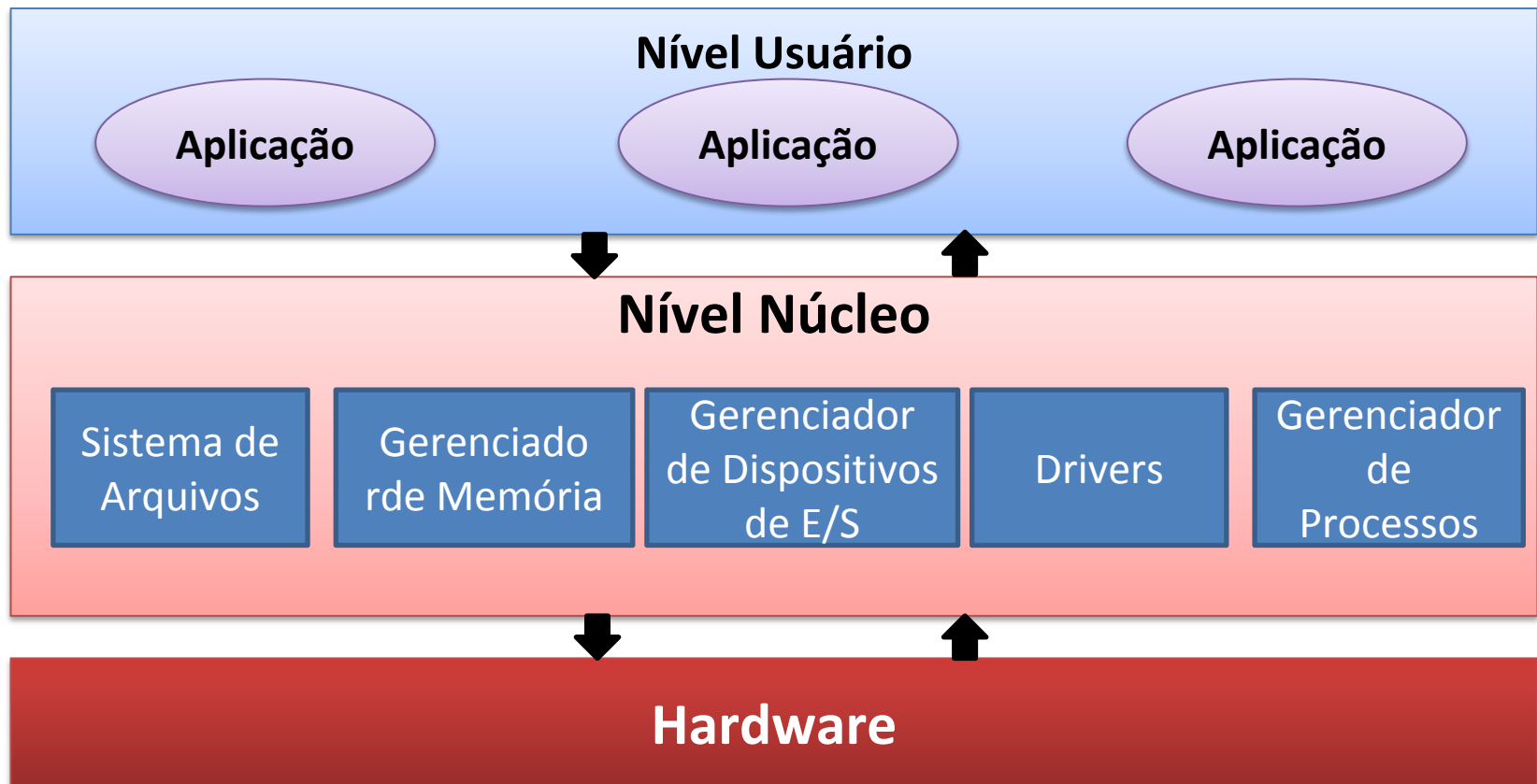
Arquitetura Interna de um Sistema Operacional



- Sistema Operacional Monolítico
 - Todos os módulos do núcleo que se inter-relacionam operam em modo núcleo.
 - O sistema é mais compacto e tem melhor desempenho uma vez que qualquer componente do núcleo pode acessar os demais sem quaisquer restrições.
 - A desvantagem é que qualquer erro causado por um componente do núcleo pode levar todo o sistema ao colapso.
 - Outra desvantagem é a manutenção do sistema que se torna mais complexa.

Arquitetura Interna de um Sistema Operacional

- Sistema Operacional Monolítico



Arquitetura Interna de um Sistema Operacional



- Sistema Operacional em Camadas
 - Nesse modelo o SO é estruturado em diferentes camadas desde a de mais baixo nível, a que controla do hardware, até a camada de mais alto nível, que controla as aplicações de usuário.
 - Cada camada é especializada em uma determinada função do SO.
 - A desvantagem do modelo em camadas é o baixo desempenho devido a troca de comunicação entre as camadas do sistema.

Arquitetura Interna de um Sistema Operacional



- Sistema Operacional em Camadas



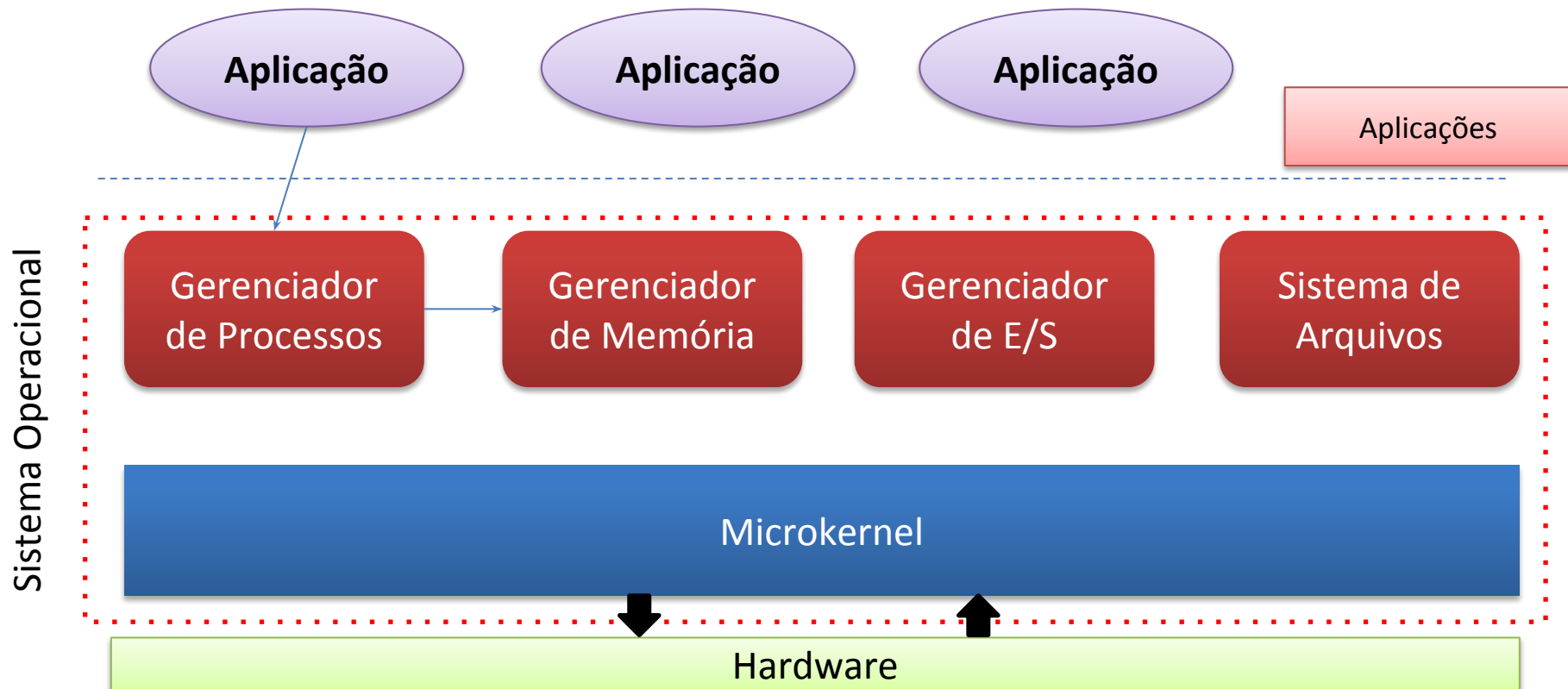
Arquitetura Interna de um Sistema Operacional



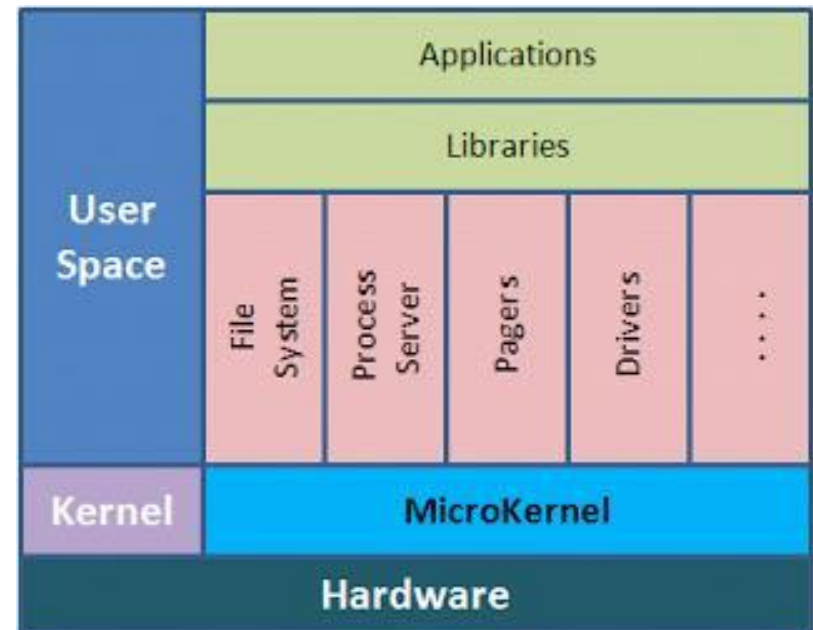
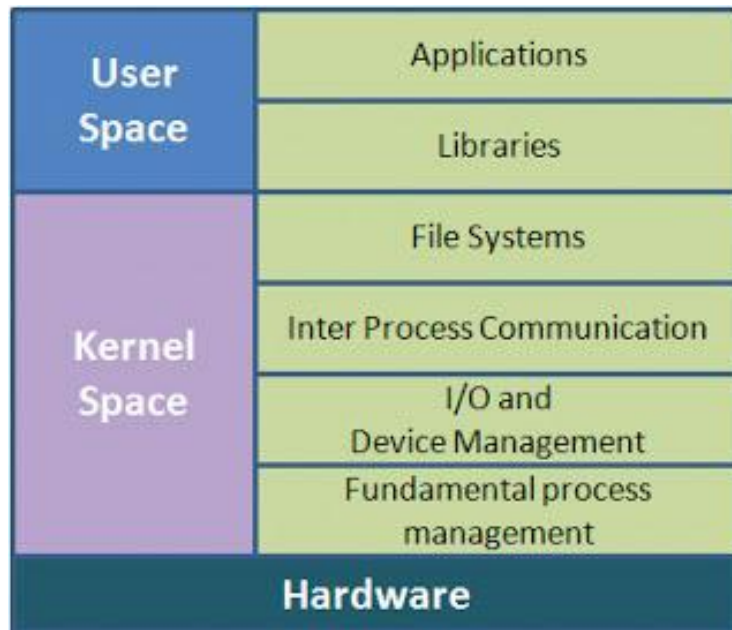
- Sistema Operacional com Micronúcleo
 - Nesta abordagem **somente os código de baixo nível permanecem no núcleo**, o restante das funcionalidades são implementadas fora do núcleo.
 - A **comunicação entre as diversas parte dos SO é feita por meio de troca de mensagens** (abordagem *cliente-servidor*).
 - A vantagem é que **se um módulo sofrer alguma falha, esta não se alastra por todo o sistema**.
 - A **desvantagem é a quantidade de mensagens trocadas** entre os módulos para se fazer uma operação simples do tipo abrir um arquivo, por exemplo.

Arquitetura Interna de um Sistema Operacional

- Sistema Operacional com Micronúcleo



Monolítico vs. Microkernel



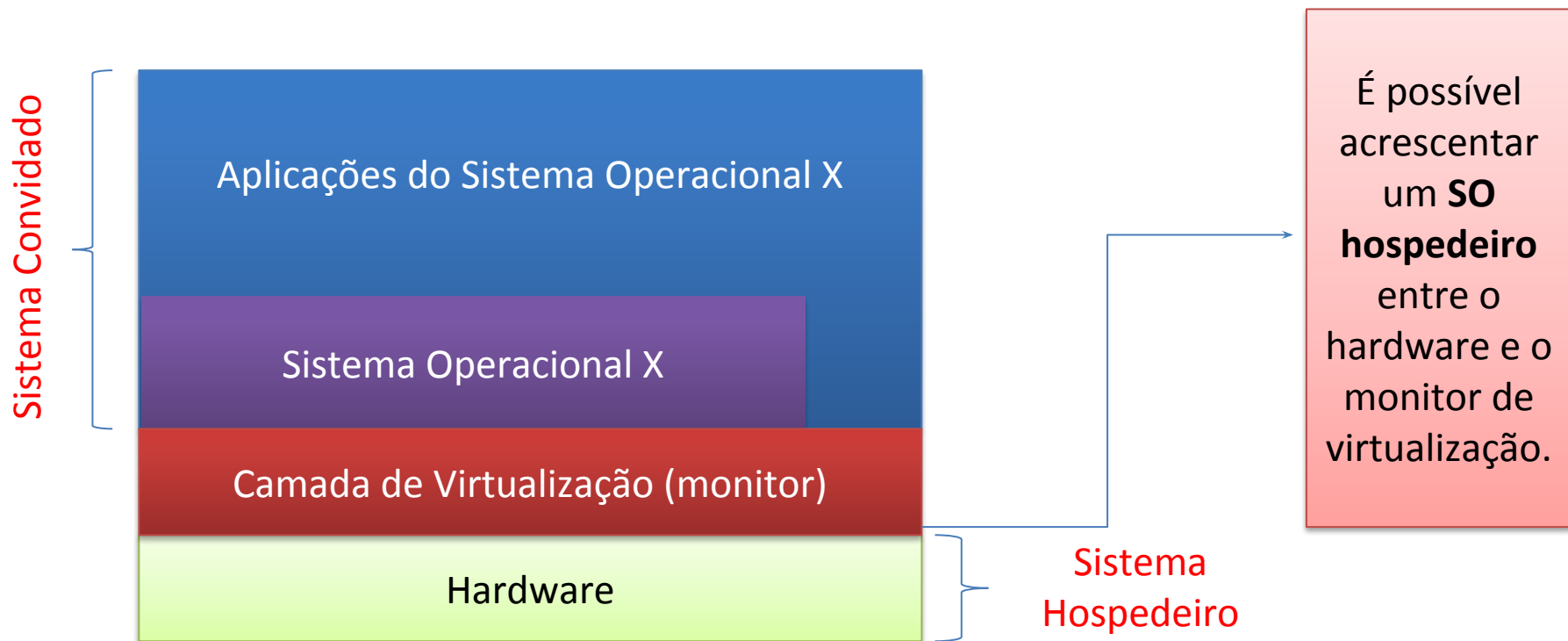
Arquitetura Interna de um Sistema Operacional



- Sistema Operacional com Máquina Virtual
 - Uma máquina virtual **permite um acoplamento entre interfaces distintas** permitindo que um sistema projetado para a máquina X possa executar também na máquina Y.
 - Um **sistema baseado em máquina virtual consiste de três partes:**
 1. Um sistema real (hospedeiro);
 2. Uma camada de virtualização (monitor de virtualização);
 3. Um sistema convidado que executa sobre o sistema virtualizado.

Arquitetura Interna de um Sistema Operacional

- Sistema Operacional com Máquina Virtual



Arquitetura Interna de um Sistema Operacional



- Sistema Operacional com Máquina Virtual
 - O ambiente de máquina virtual pode ser dividido em duas grandes famílias:
 - **Máquinas Virtuais de Aplicação:** são destinado a suportar apenas um processo ou aplicação convidada específica. Exemplo: máquina virtual Java (JVM).
 - **Máquinas Virtuais de Sistema:** construídos para suportar sistemas operacionais convidados completos, com aplicações convidadas executando sobre eles. Exemplos: VMWare, VirtualBox e Xen.

Tipos de Sistemas Operacionais



- Os sistemas operacionais podem ser classificados de acordo com o conjunto de serviços oferecidos e a aplicação ao qual se destina.
- Os SO's podem ser classificados em:
 - Batch (lote)
 - Rede
 - Distribuído
 - Multiusuário
 - Desktop
 - Servidor
 - Embarcado
 - Tempo Real

Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Batch (*lote*)
 - Esse tipo de SO se caracteriza por organizar em uma *fila* todos os programas a serem executados.
 - Um programa era executado **sem qualquer interação com o usuário**.
 - Exemplos de SO do tipo lote:
 - **OS/360**
 - **VMS**

Tipos de Sistemas Operacionais



- Sistema Operacional do tipo Rede
 - Esse tipo de SO **deve possuir suporte para operações em redes de computadores.**
 - Deve **permitir o acesso a recursos remotos** e permitir o compartilhamento de seus próprios recursos a outros computadores da rede.
 - Exemplo de SO do tipo rede:
 - Windows NT
 - Windows 7
 - Linux e variações
 - Unix e variações
 - **Obs.: A maioria dos SO's atuais podem ser considerados de rede.**

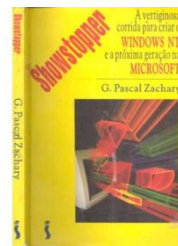
Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Rede

- Curiosidade

- O Windows NT (*New Technology*) foi o precursor dos sistemas operacionais ditos de arquitetura moderna da Microsoft.

Em 1994 o Jornalista G. Pascal Zachary escreveu o livro ***Showstopper! The breackneck race to create Windows NT and the next generation at Microsoft***, relatando o dia a dia da equipe de desenvolvedores do Windows NT. Esse livro foi lançado no Brasil em 1995 pelo editora Siciliano sob o título: ***Showstopper – A vertiginosa corrida para criar o Windows NT e a próxima geração na Microsoft***.



Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Distribuído
 - Em um SO distribuído os recursos de cada máquina estão disponíveis globalmente, de forma transparente aos usuários.
 - Exemplos de SO's do tipo distribuído:
 - Amoeba
 - Mach
 - Clouds

Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Multiusuário
 - Esse tipo de SO permite que **vários usuários acessem os recursos disponíveis**.
 - É importante que o SO tenha algum método de **identificar e autenticar os usuários**, permitindo ou negando o acesso a recursos do sistema.
 - Exemplos de SO's do tipo multiusuário:
 - **A maioria dos SO's disponíveis atualmente são multiusuários.**

Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Desktop
 - São sistemas projetados para usuários finais, ou seja, o usuário que irá utilizar aplicativos como: planilhas eletrônicas, editor de texto, acesso a internet etc.
 - Devem possuir uma **interface gráfica amigável** e de fácil utilização.
 - Exemplos de SO's do tipo desktop:
 - **Windows**
 - **Linux**
 - **MacOS**

Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Servidor
 - São SO's projetados para permitir a **gestão eficiente de grande quantidade de recursos**, tais como: discos, impressoras, processadores etc).
 - Exemplos de SO's do tipo servidor:
 - Unix
 - FreeBSD
 - OpenBSD

Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Embarcado
 - São sistemas projetados para executar e gerenciar hardware com poucos recursos computacionais.
 - Estão presentes em aplicações industriais, automotivas e *smartphones*.
 - Exemplos de SO's do tipo embarcado:
 - Linux Embedded
 - FreeRTOS
 - EPOS
 - Symbian
 - Android

Tipos de Sistemas Operacionais

- Sistema Operacional do tipo Tempo Real
 - São sistemas **projetados para atender certas restrições temporais**, ou seja, **seu tempo de resposta deve ser conhecido a priori** no melhor e no pior caso de operação.
 - Os SO's de tempo real podem **ser do tipo *soft real time* (brando) e *hard real time* (críticos)**.
 - Exemplos de SO's do tipo tempo real:
 - RT-Linux
 - VxWorks
 - QNX

Chamadas de Sistema

- Chamadas de sistema ou chamadas ao sistema **são rotinas implementadas no SO para a execução de funções específicas.**
- Essas rotinas **são executadas pelas aplicações de usuário** quando estas necessitam de algum serviço do SO.
- Todo sistema operacional possui uma API (*Application Programmer Interface*) que implementa um conjunto de chamadas de sistemas.

Chamadas de Sistema

- Exemplo de chamada de sistema
 - Segue alguns exemplos de chamadas de sistema no SO Unix:
 - **fork()** –cria um novo processo fazendo uma cópia do processo existente
 - **exit()** – encerra o processo solicitante
 - **open()** – abre um arquivo para leitura/escrita
 - **read()** – lê dados de um arquivo aberto
 - **write()** – escreve dados em um arquivo aberto

Chamadas de Sistema

- Exemplo de chamada de sistema
 - Programa para ler dados de um arquivo em disco.

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #include <fcntl.h>

4.  #define BUFFER_SIZE 100

5.  int main()
6.  {
7.      char buffer[BUFFER_SIZE];
8.      int fd, n;

9.      fd = open("dados.txt", O_RDONLY);

10.     if (fd < 0) {
11.         printf("Erro ao abrir o arquivo\n");
12.         exit(0);
13.     }

14.     n = read(fd, buffer, BUFFER_SIZE);
15.     printf("Dados lidos...: %s\n", buffer);

16.     return 0;
17. }
```

Histórico

- Décadas de 40 e 50
 - Não havia o conceito de sistema operacional.
 - Cada programa executava sozinho e detinha todo o controle da máquina.
 - O programador era o responsável por todo o controle do hardware.
 - Aparecimento das bibliotecas de sistema para facilitar a implementação de aplicações de usuário.
 - Surgimento de um programa monitor para auxiliar a carga e descarga de aplicações de usuário.

Histórico

- Décadas de 60 e 70
 - Surgimento do conceito de tempo compartilhado.
 - Lançamento do OS/360 pela IBM.
 - Desenvolvimento dos sistemas Multics.
 - Desenvolvimento da linguagem de programação C, base de desenvolvimento de muitos SO's.
 - Desenvolvimento do sistema Unix.

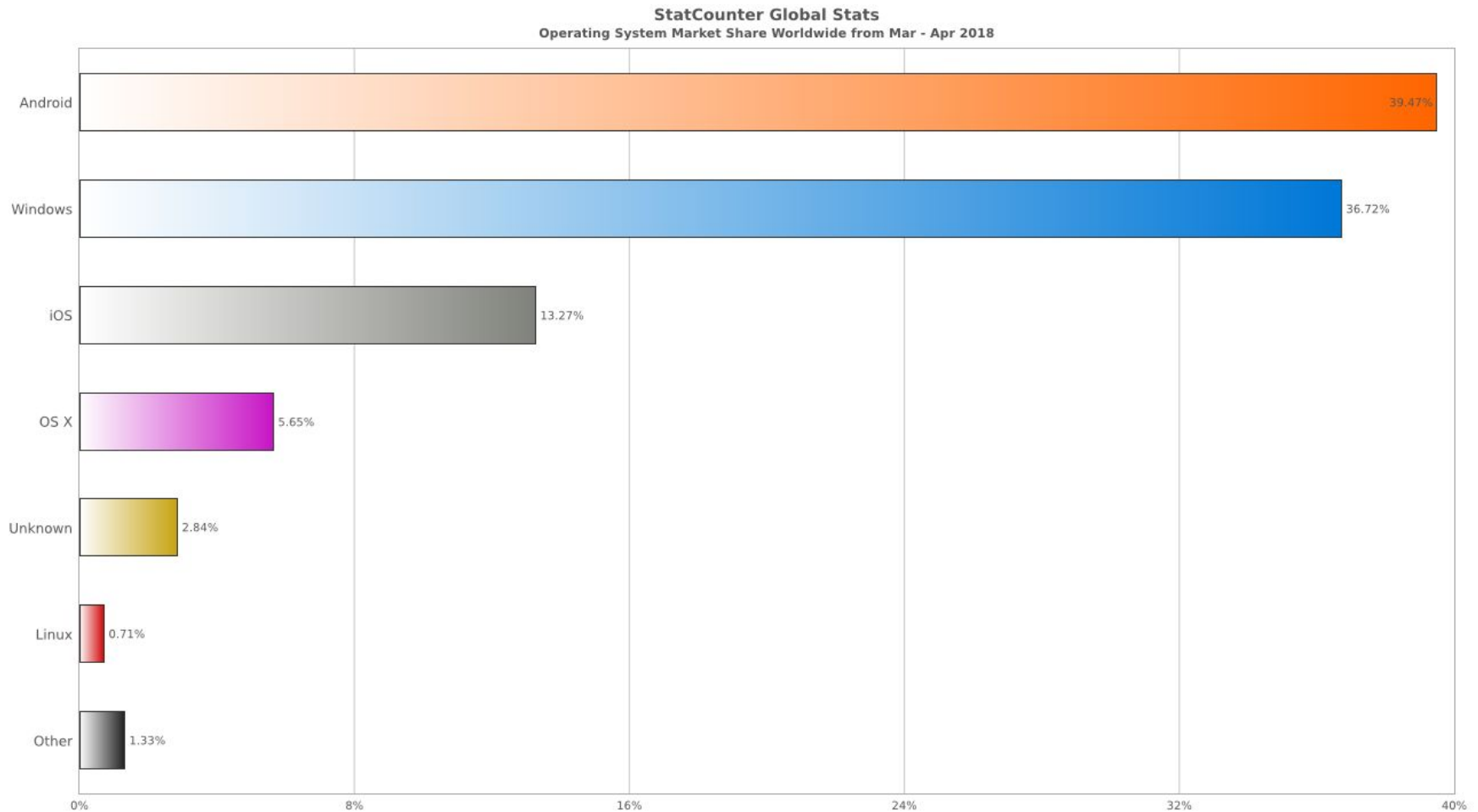
Histórico

- Décadas de 80 e 90
 - Lançamento da primeira versão do sistema operacional MS-DOS pela Microsoft.
 - A Apple lança a versão 1.0 do Macintosh OS.
 - Lançamento do Microsoft Windows 1.0.
 - Desenvolvimento do Minix e dos OS/2.
 - Desenvolvimento do Linux.
 - Unix de código aberto: FreeBSD e NetBSD.
 - Lançamento do Windows NT.

Histórico

- Década de 2000
 - Lançamento do MacOS X.
 - Lançamento do Windows XP, Vista, 7 e 8.
 - Lançamento do kernel versão 2.6 do Linux.
 - Lançamento do iOS 1 (2007)
 - Lançamento do Android 1.0 (2008)
 - ...

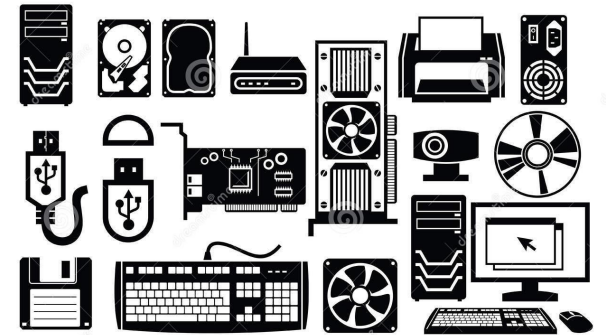
Uso dos SO's



Revisão



Sistema
operacional
(= Software)



Revisão

- Arquitetura interna
 - Núcleo (kernel)
 - Código de inicialização
 - Drivers
- Distintos tipos de SO
 - Mais comuns: desktop, multiusuário e embarcado

Revisão

- Chamadas de sistema
 - Funcionalidades de SW oferecidas
 - Interface de programação do SO