

# Sistemas Operacionais

4º período

Professora: Michelle Hanne

# O que é um sistema operacional (SO)?



- Na era atual, os sistemas computacionais são constituídos de processadores, memórias (volátil e fixa), diversos tipos de disco (hard disk, pen drives, nuvem), teclado, conjunto multimídia (CD, DVD, som), monitor, impressoras, redes, interface com usuário, mouse, câmeras, scanners, entre outros. **Em suma, sistema computacional é um sistema muito complexo para desenvolvedores tanto de software quanto de hardware (WATSON, 2013).**

# O que é um sistema operacional (SO)?

O desenvolvedor de um software, como, por exemplo, o **Waze não precisa se preocupar com o gerenciamento de memória**; no mesmo sentido, um desenvolvedor de impressora ou de qualquer outro hardware não precisa se preocupar com o reconhecimento do hardware com relação ao dispositivo conectado.

Os sistemas operacionais têm a finalidade de controlar e gerenciar tanto o hardware, através de uma interface amigável, quanto o software (programas de usuários), por meio de diversos recursos disponíveis no sistema.

# O que é um sistema operacional (SO)?

O sistema operacional é um software de base, que provê um conjunto de aplicações que tem a finalidade de atuar de forma transparente entre os programas de usuários e os hardwares de qualquer dispositivo computacional, como computadores, celulares, tablets, televisão, etc.

# O que é um sistema operacional (SO)?

O sistema operacional é um software de base, que provê um conjunto de aplicações que tem a finalidade de atuar de forma transparente entre os programas de usuários e os hardwares de qualquer dispositivo computacional, como computadores, celulares, tablets, televisão, etc.

# O que é um sistema operacional (SO)?

## SO como gerente de recursos:

- Gerenciar o compartilhamento de recursos, como memória, hardware e processador
- Prover diversos serviços que simplifiquem as aplicações.
- Apresentar uma interface entre programas de aplicação e hardwares

## Em um sistema computacional os recursos são, via de regra, escassos e disputados

- Processador
- Memória
- Disco
- Impressora e outros periféricos
- Dados compartilhados

## Uso compartilhado ou concorrente de recursos

# SO como Gerenciador de Recursos

A **memória**, por exemplo, é um recurso exigido por qualquer aplicação. Um sistema operacional não pode permitir que uma aplicação tenha acesso à determinada área de memória utilizada por outra aplicação, ou seja, uma área de memória reservada por outra aplicação.

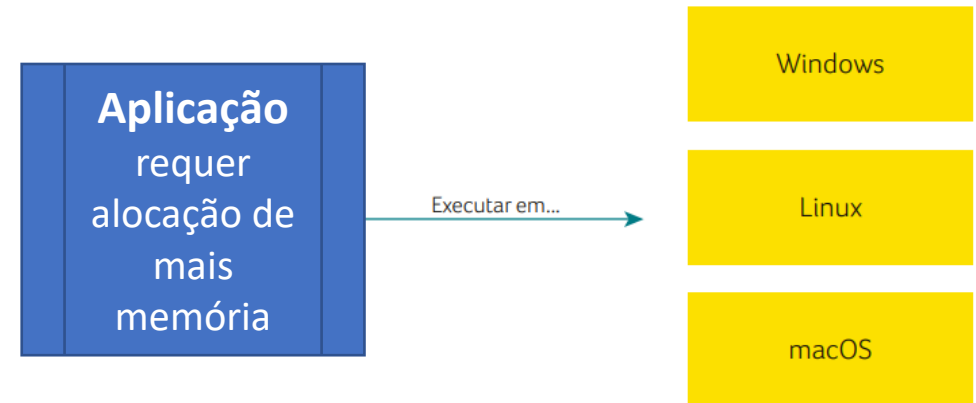
Se um programa malicioso – por exemplo, um vírus – acessar uma área de memória reservada por outra aplicação, poderá ocorrer um travamento do sistema ou, no mínimo, falhas constantes de resultados, que podem ser prejudiciais. Ex: “Tela Azul da Morte”, exemplo de acesso indevido às áreas de memórias compartilhadas

# SO como Provedor de Serviços

Serviço é qualquer tarefa (recurso) que seja utilizada por uma aplicação.

Uma aplicação pode, em determinados momentos, exigir mais poder de processamento, mais alocação de memória, mais controle de entrada e saída (I/O); enfim, pode demandar do sistema operacional os serviços necessários para que ela opere com eficiência.

Por exemplo, **editores de vídeo** demandam mais processamento, mais capacidade de memória e um melhor gerenciamento dos recursos pelo sistema operacional.



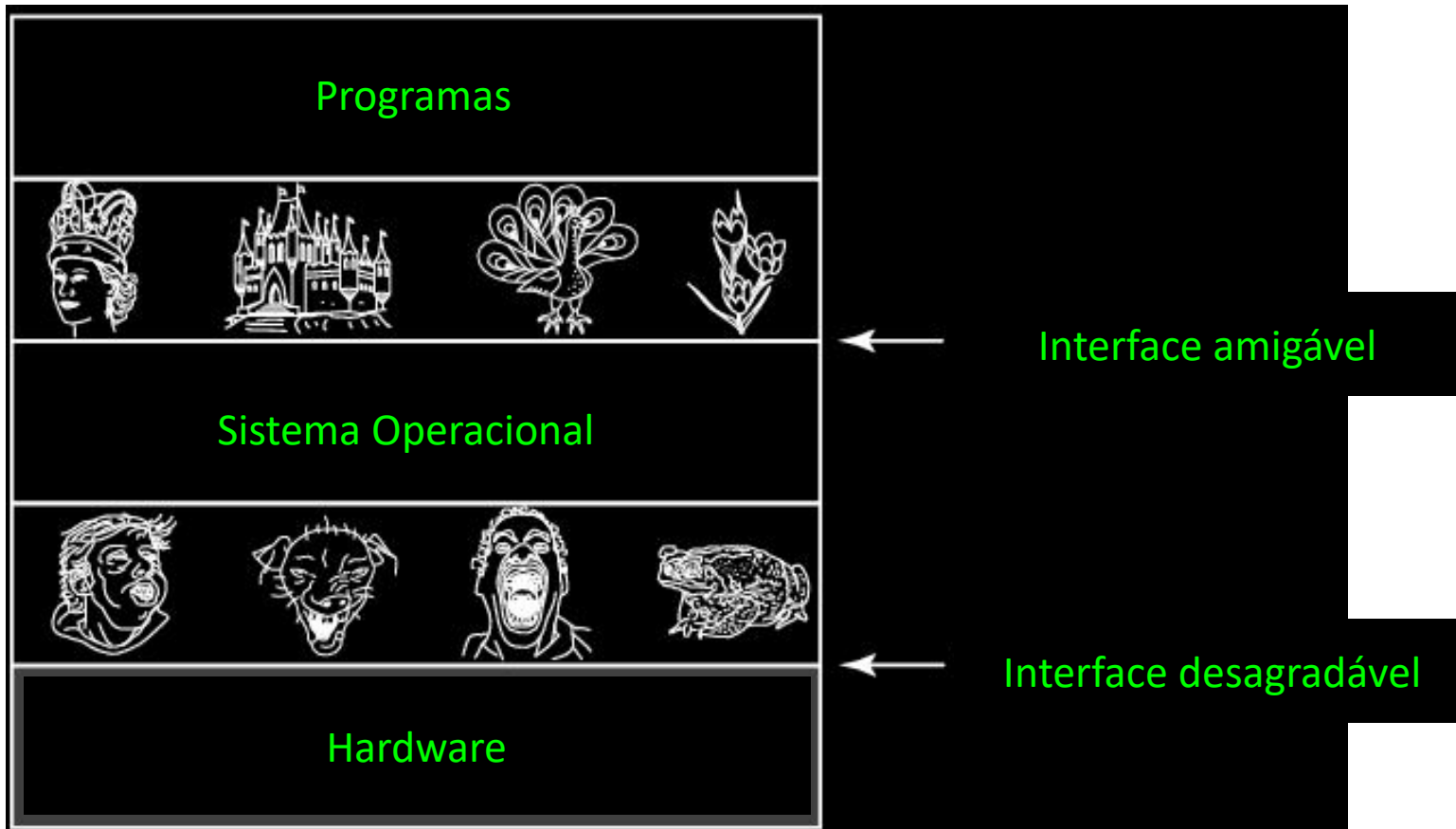


# SO como uma Máquina Virtual ou Estendida

Um usuário comum especifica um **computador não somente pelo hardware, mas pelo conjunto hardware + sistema operacional**. Esse conceito é definido pela simplicidade de execução de tarefas que o sistema operacional proporciona às aplicações ou usuário.

A interface **traduz instruções virtuais** em instruções nativas num processo de compilação em tempo de execução (*just-in-time*). Essa visão abstrata é chamada de **Máquina Virtual ou Estendida**.

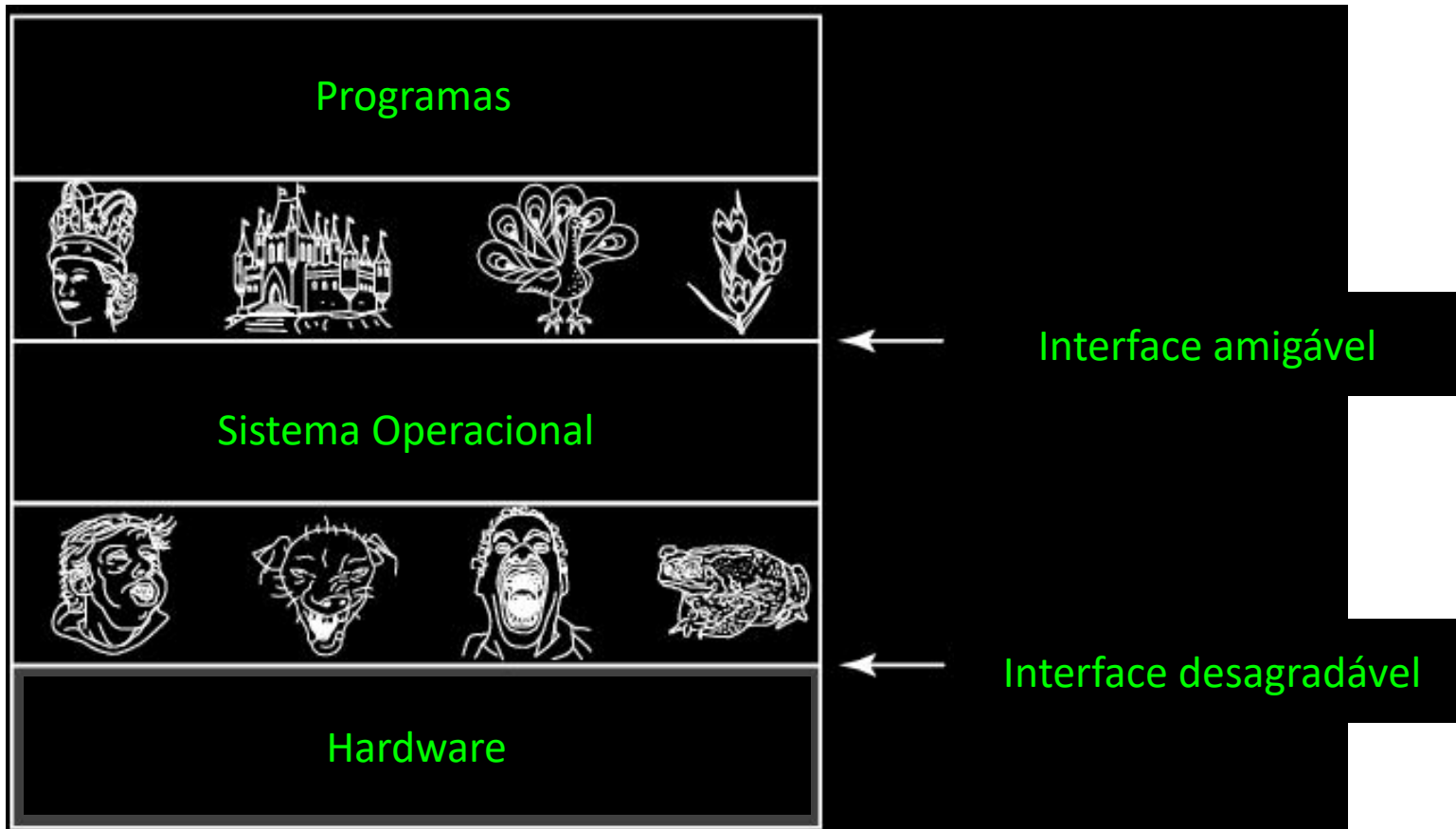
# SO como uma Máquina Virtual ou Estendida



- Por exemplo: ao instalar um programa Pacote Office, o usuário não se atenta ao processador ou tipo de memória, **pois o sistema operacional abstrai essa informação**, fazendo o elo entre a aplicação do usuário e o hardware da máquina.

Fonte: adaptado de Tanenbaum(2008)

# SO como uma Máquina Virtual ou Estendida



- Por exemplo: ao instalar um programa Pacote Office, o usuário não se atenta ao processador ou tipo de memória, **pois o sistema operacional abstrai essa informação**, fazendo o elo entre a aplicação do usuário e o hardware da máquina.

Fonte: adaptado de Tanenbaum(2008)

# Evolução Histórica dos Sistemas Operacionais

*A evolução histórica dos sistemas operacionais está diretamente relacionada à arquitetura dos computadores nos quais eles são executados.*

# Primeira Geração (1945 – 1955)

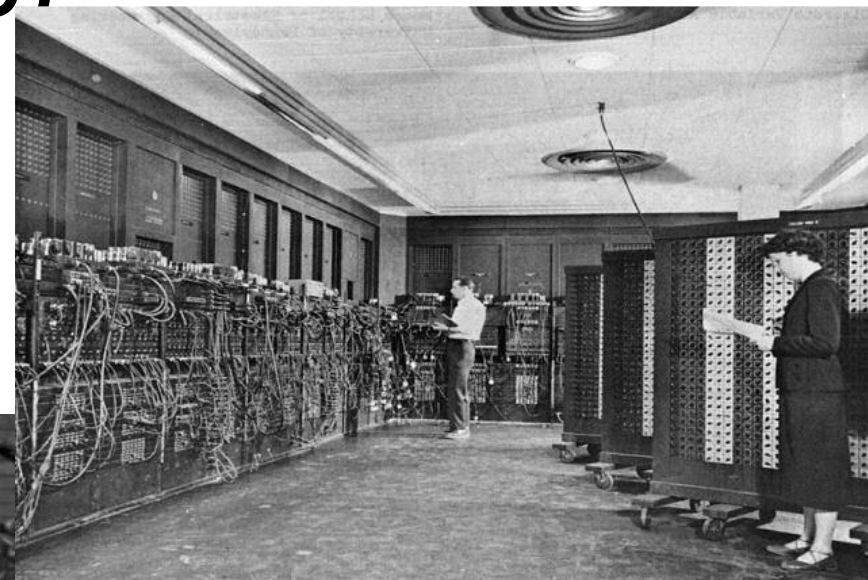
Computadores pos II Guerra Mundial (1945-1955), segundo Tanenbaum

- Computadores valvulados
- Maquinas gigantescas
- SO praticamente inexistente: uso direto do hardware

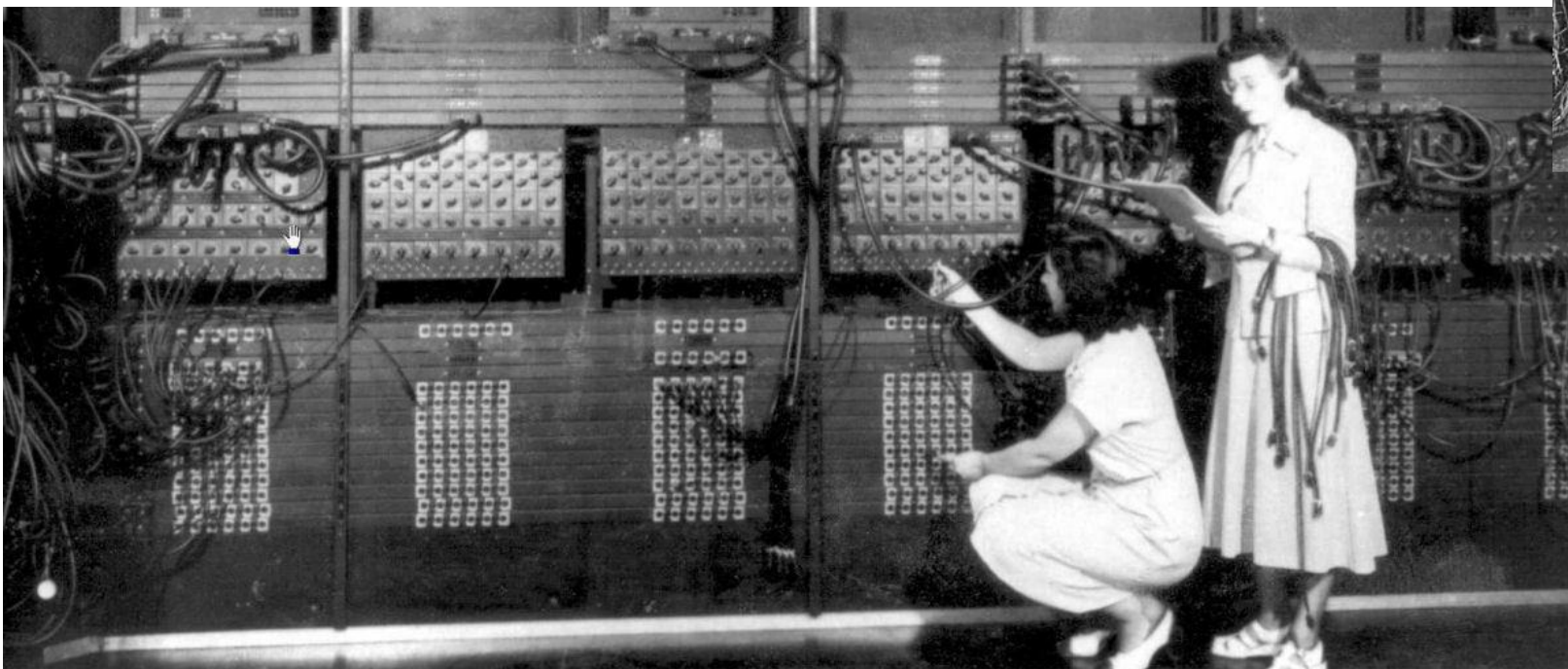
Naquela época (meados da década de 40), projeto, construção, programação, operação e manutenção eram realizados por um único grupo de pessoas. Não havia qualquer tipo de programação, nem Assembly. Toda e qualquer ligação e operação era eletromecânica, e a programação era em linguagem pura de máquina. Não havia o conceito de sistema operacional.

# Primeira Geração (1945 – 1955)

ENIAC (Electronic Numerical Integrator and Computer)



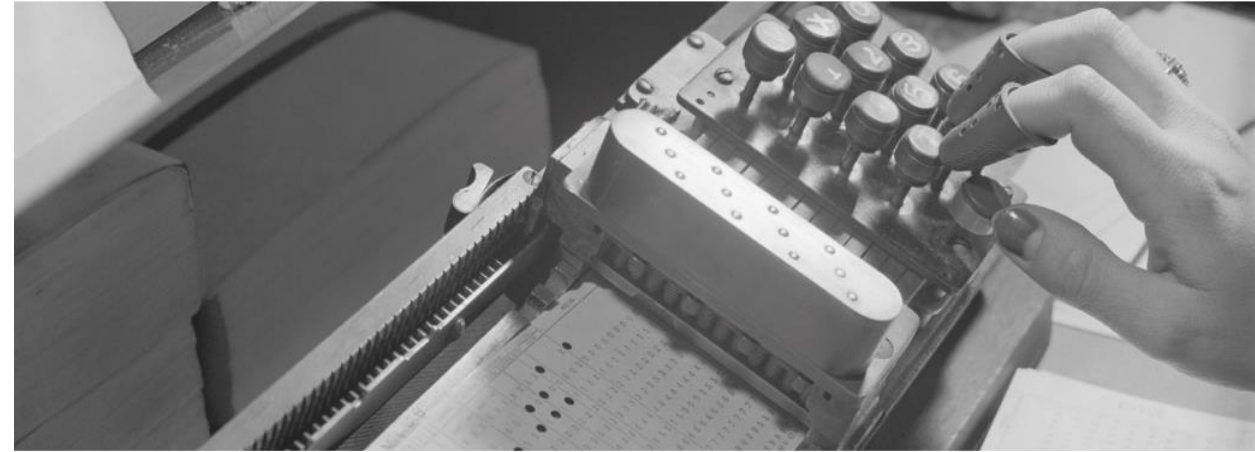
Computador criado para realização de cálculos balísticos – possuía 18 mil válvulas, 10 mil capacitores e 70 mil resistores; além disso, pesava 30 toneladas, consumia cerca de 140 kW e era capaz de realizar 5 mil adições por segundo.



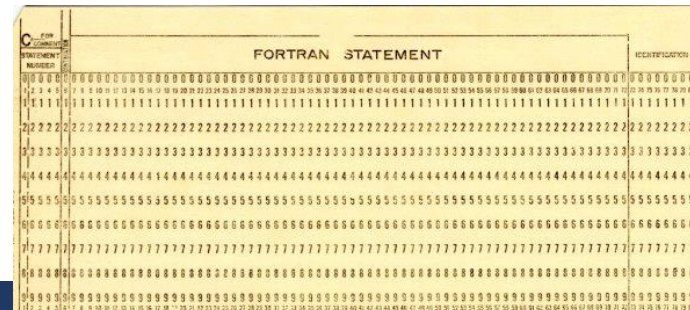


# Segunda Geração (1956-1965)

- Transistores e sistemas batch (lote)
- Programação em cartões perfurados
- Leitora de cartões perfurados
- SO: interpretador
- Lote de programas carregados de cartões para uma fita magnética
- SO: execução sequencial do lote
- **O FMS (Fortran Monitor System) e o IBSYS, sistema racional da IBM, eram os sistemas operacionais típicos dessa época.**



Fonte: SHUTTERSTOCK, 2018.

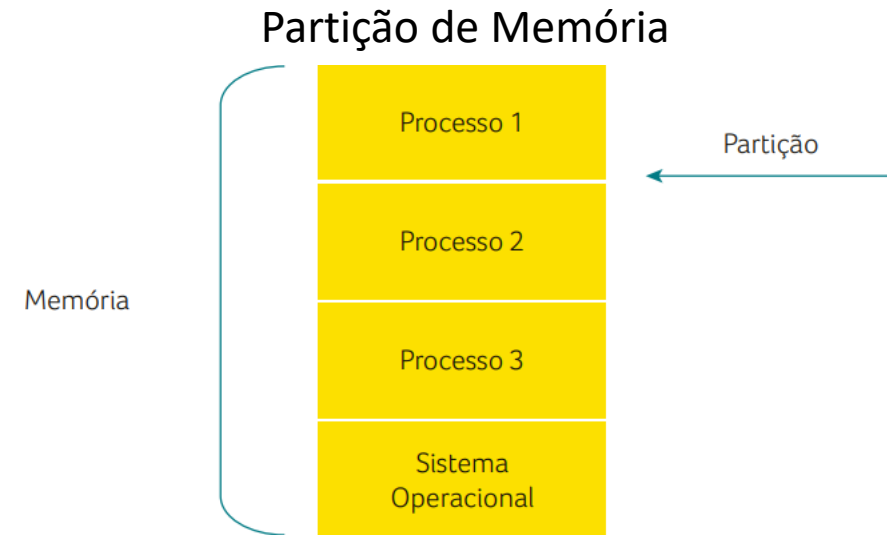


**Assembly e Fortran** são as linguagens originadas na época que permitiram que os programas deixassem de **ser meramente fios e conectores diretamente executados no hardware** e que propiciaram o processo de desenvolvimento dos primeiros programas.

# Terceira Geração (1965-1980)

- Circuitos integrados e multiprogramação
- Diminuição de tamanho e de custo
- Surgimento das “famílias” de máquinas
- Evolução do hardware possibilitou a multiprogramação.

Os computadores baseados em palavras (**IBM-7094**) eram utilizados para fins científicos; os baseados em caracteres (**IBM-1491**), para fins comerciais por bancos e companhias de seguros. Então, a IBM criou o **System/360**, que contemplava ambas as necessidades.



O **UNIX**, criado pelo laboratório Bell, da AT&T, também foi lançado na mesma época, originalmente para os computadores PDP-7 baseados no sistema MULTICS (*Multiplexed Information and Computer Service*) e, posteriormente, reescrito em linguagem C, com portabilidade adicionada.



# Terceira Geração (1965-1980)

- **Evolução:** time-sharing ou tempo compartilhado
- Usuários conectados por terminais
- SO dedica uma fatia de tempo do processador para cada usuário ou processo
- **Utilizado até os dias de hoje**

## E o SO, o que mudou?

- Gerenciamento de memória
- Escalonamento de processos
- Prioridades
- Concorrência e compartilhamento...

UNIVAC Uniscope 300



# Quarta Geração (1981-1990)

- O desenvolvimento dos **circuitos integrados para integração em larga escala (LSI)** e para integração em **muito larga escala (VLSI)** permitiu que projetos de **diminuição e de custos mais baixos** dos computadores fossem viabilizados, tornando o mercado impulsionado pelos **mini e superminicomputadores**.
- Os sistemas operacionais apresentaram interfaces mais amigáveis.
- Surge, então, o **Microsoft Disk Operating System (MS-DOS)**, sistema operacional da fabricante Microsoft que dominou o mercado de **SO para computadores pessoais**, assim como houve a expansão do sistema operacional **UNIX**.

IBM-PC



Fonte: SHUTTERSTOCK, 2018.

# Quarta Geração (1981-1990)

- Computadores pessoais: Apple e IBM PC
- **E o SO dos microcomputadores?**
  - Computador pessoal: precisa gerenciar vários usuários?
  - Um usuário: vai fazer varias tarefas ao mesmo tempo?
  - Microsoft e o MS-DOS
  - Surgimento e importância do Linux
- No meio da década de 80, com a difusão das redes de computadores, surgiram os sistemas operacionais de rede e os sistemas operacionais distribuídos



# Quinta Geração (1991 aos dias atuais)

A partir do início da década de 1990, evoluções no hardware e nos programas, bem como a convergência **entre redes e telecomunicação**, permitiram um desenvolvimento crescente na capacidade de processamento, armazenamento e transmissão de dados, acrescido do boom da internet e de melhorias significantes nos processadores.

Com isso, os sistemas operacionais baseados em interfaces gráficas foram consolidados.

Surgimento de novos dispositivos, novos sistemas wireless e na integração entre esses dispositivos (internet of things) e na computação em nuvem, linguagens concorrentes, naturais e paralelas, etc.

# Quinta Geração (1991 aos dias atuais)

## Ultra *Large Scale Integration* (ULSI)

- De computadores pessoais a servidores

## Evolução do TCP/IP - Abertura comercial da Internet e surgimento da Web

## Miniaturização: surgimento de dispositivos móveis

## E o SO?

- Sistemas cliente/servidor
- Sistemas multiusuário
- Interfaces gráficas
- Interfaces intuitivas
- Sensibilidade ao toque
- Segurança
- Virtualização
- Processamento distribuído
- Sistemas de alta disponibilidade: clusters

# Quinta Geração (1991 aos dias atuais)

## Ultra *Large Scale Integration* (ULSI)

- De computadores pessoais a servidores

## Evolução do TCP/IP - Abertura comercial da Internet e surgimento da Web

## Miniaturização: surgimento de dispositivos móveis

## E o SO?

- Sistemas cliente/servidor
- Sistemas multiusuário
- Interfaces gráficas
- Interfaces intuitivas
- Sensibilidade ao toque
- Segurança
- Virtualização
- Processamento distribuído
- Sistemas de alta disponibilidade: clusters

# Quinta Geração (1991 aos dias atuais)

## Ultra *Large Scale Integration* (ULSI)

- De computadores pessoais a servidores

## Evolução do TCP/IP - Abertura comercial da Internet e surgimento da Web

## Miniaturização: surgimento de dispositivos móveis

## E o SO?

- Sistemas cliente/servidor
- Sistemas multiusuário
- Interfaces gráficas
- Interfaces intuitivas
- Sensibilidade ao toque
- Segurança
- Virtualização
- Processamento distribuído
- Sistemas de alta disponibilidade: clusters

# Quinta Geração (década de 2010)

Popularização de tablets e smartphones

Consolidação da computação em nuvem

- IaaS (Infrastructure as a Service)
- SaaS (Software as a Service)

E o SO?

- Interface mais intuitiva
- Integração total com redes de comunicação
- Portabilidade
- Nascimento e morte: Symbian, WinMobile e WinPhone,
- Blackberry
- Consolidação: Android, iOS



# “O Zoológico dos Sistemas Operacionais”

TANENBAUM, (2016)

- Computadores de grande porte (mainframes)
- Servidores e multiprocessadores
- Computadores pessoais
- Computadores portáteis/móveis
- Sistemas embarcados
- Nós sensores
- Sistemas de tempo real
- Smartcards

## Desafios Atuais:

- **Compatibilidade:** resiliente ao avanço da tecnologia?
- **Portabilidade:** funciona em varias plataformas?
- **Interoperabilidade:** permite ambientes heterogêneos?
- **Escalabilidade:** e capaz de usufruir das capacidades de
- **Processamento?**

# SO: Estrutura e Conceitos

# SO: Estrutura e Conceitos

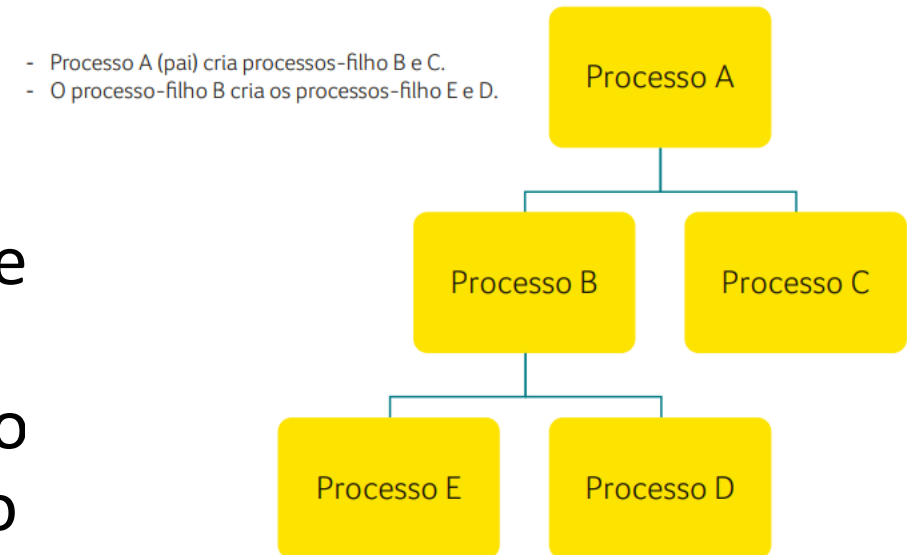
A interface de um sistema operacional proporciona **transparência entre a aplicação do usuário e o sistema operacional**, pois é formada por um conjunto de **instruções estendidas**. Essas instruções são conhecidas como *system calls* e definem implementações variadas, pois **os processadores mais atuais executam instruções em níveis de privilégio diferentes**: nível de usuário (modo usuário) e nível de kernel (modo protegido).

# SO: Estrutura e Conceitos

## Processos:

- São tarefas executadas no núcleo do processador, ou seja, no programa em execução.
- Associados a um processo estão um endereçamento de memória e um número de identificação. Cada processo (processo-pai) pode criar um processo-filho, e cada processo filho pode criar outro processo-filho, gerando assim uma estrutura de árvore de processos

Estrutura de Árvore de um Processo



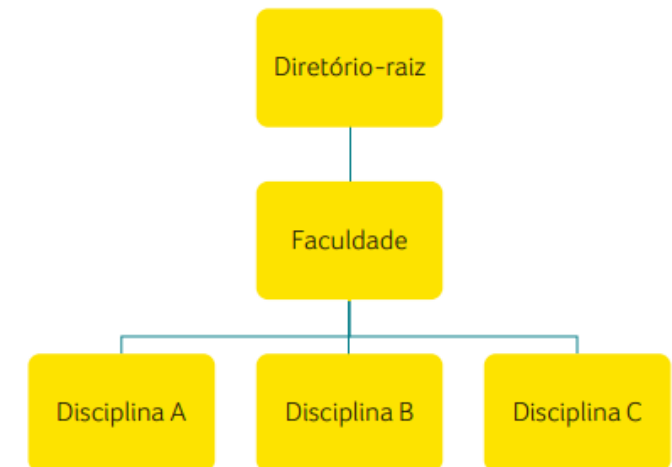
# SO: Estrutura e Conceitos

## Arquivos:

- É qualquer informação que possa ser armazenada em um dispositivo eletrônico ou computador, como som, imagem, texto, figura, vídeo, etc. Os sistemas operacionais possuem a capacidade de organização desses arquivos em locais chamados diretórios.

### Estrutura de Árvore de Arquivos

O caminho da disciplina C:  
diretório raiz/faculdade/disciplina C



# SO: Estrutura e Conceitos

## Shell:

- é o interpretador de comando utilizado dentro do sistema operacional. O usuário utiliza essa interface para fazer conexão com o sistema operacional; sendo assim, toda vez que um sistema operacional é iniciado, o shell também é iniciado.

# SO: Estrutura e Conceitos

## Kernel:

- O kernel (núcleo do sistema operacional) é composto por essas diversas rotinas de tratamento de interrupções, gerenciamento de memória, operações de entrada/saída, escalonamento e controle de processos, etc.

## Chamadas de sistema (*system calls*):

- um mecanismo pelo qual um processo pode solicitar um dos serviços, como interrupção de um processo em execução.

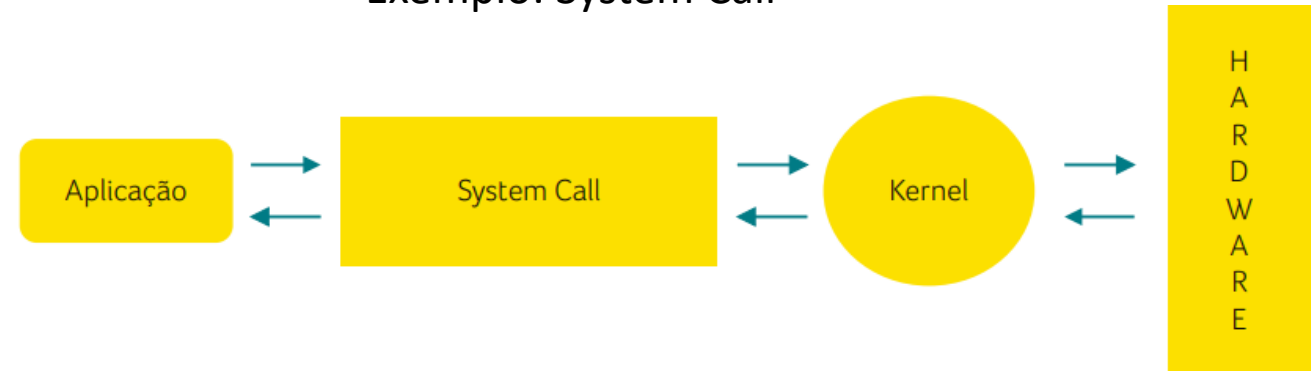
# SO: Estrutura e Conceitos

As rotinas da chamada de sistema (*system call*) podem ser divididas em três grupos: gerência de processos, gerência de memória, gerência de entrada/saída.

## Principais rotinas da chamada de sistema (*system call*)

Gerência de processo
Criação e eliminação de processos.
Alteração das características de processos.
Sincronização de processos.
Comunicação entre processos.
Gerência de memória
Alocação e desalocação de memória.
Gerência de entrada/saída
Operações de I/O.
Manipulação de arquivos e diretórios.

## Exemplo: System Call



O núcleo do sistema operacional opera em modo kernel para que se permita ao sistema controlar todos os recursos e solucionar todos os problemas de qualquer nível de privilégio.



# Estrutura Interna de um SO

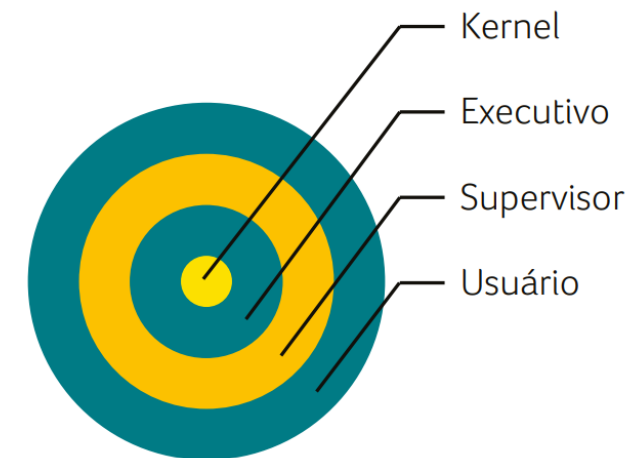
## Sistemas Monolíticos

- Essa estrutura é a mais comum de ser encontrada, pois qualquer procedimento pode interagir com outro procedimento em qualquer parte do programa. Os procedimentos são **compilados** e, depois, linkados como um só
- programa; caso um erro aconteça, o sistema é paralisado. O **MS-DOS, da Microsoft**, é um exemplo de estrutura de sistema operacional monolítico Unix-like (solaris, Linux).

## Sistema em Camadas

- No sistema de estrutura em camadas, o **sistema operacional é dividido em camadas sobrepostas**. Os módulos são compostos por um conjunto de rotinas que podem ser usadas por outros módulos. O sistema **MULTICS**, por exemplo, utilizava o mesmo conceito de camadas, porém em camadas circuncêntricas, sendo o privilégio maior para a camada mais interna.

### Sistema Operacional em Camadas Circuncêntricas



# Estrutura Interna de um SO

## Exonúcleos

- É um programa na camada inferior do sistema operacional, sendo executado em modo kernel, que cria máquinas virtuais alocando recursos para elas, sem que uma máquina virtual utilize recursos de outra máquina virtual. O IBM VM/370, por exemplo, utilizava esse tipo de estrutura.

## Máquinas Virtuais:

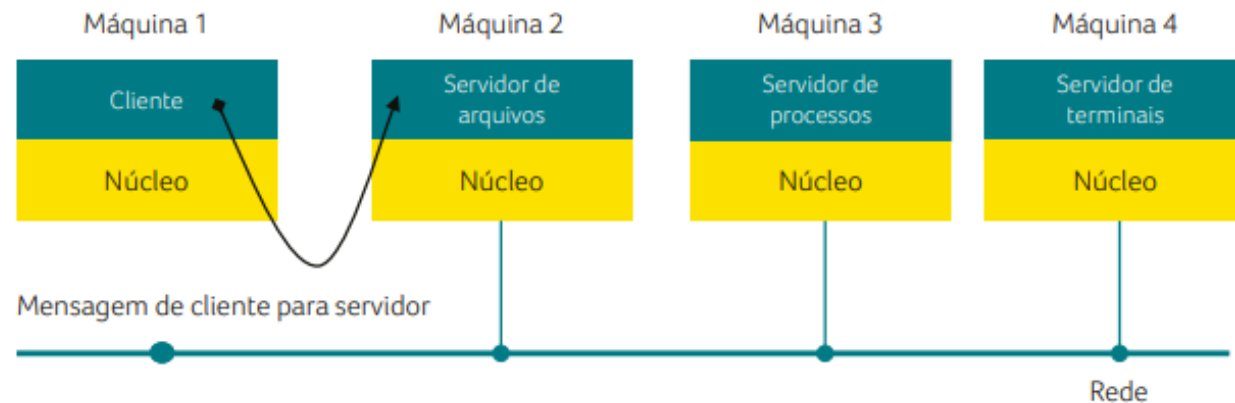
- A máquina virtual fornece todos os recursos que uma máquina real **fornece ao usuário ou aplicação, porém virtualmente**. No sistema de máquina virtual, um sistema operacional pode suportar outros sistemas operacionais sendo executados sob o SO principal

Aplicação A	Aplicação B	Aplicação C
SO convidado (Windows)	SO convidado (Linux)	SO convidado (macOS)
SO host (máquina virtual – VMware)		
Hardware		

# Estrutura Interna de um SO

## Cliente-Servidor

- A estrutura cliente-servidor possui um núcleo com o mínimo de rotinas, removendo o máximo de códigos possível em modo núcleo. **Quando os serviços são solicitados, os códigos de camadas superiores transferem-se para as camadas inferiores.**
- Quando um serviço é solicitado por um usuário (cliente), um bloco de informações é enviado à camada inferior (servidor), que executa a tarefa e retorna uma resposta.



# Tipos de Sistemas Operacionais

## Monotarefa/monoprogramável:

- sistema desenvolvido nos primórdios dos computadores. Máquinas que eram utilizadas **por apenas um usuário e todos os recursos eram exclusivos a uma única tarefa.**

## Multitarefa/multiprogramação:

- sistema desenvolvido para evolução do sistema monotarefa, é mais complexo e eficiente. Nesse sistema, **as tarefas são compartilhadas entre diversos usuários**, e os recursos são compartilhados entre as diversas tarefas executadas no processador.

## Múltiplos processadores:

- sistema em **duas ou mais unidades de processamento trabalhado em conjunto.** Destacam-se pelo grau de comunicação entre essas unidades de processamento, compartilhamento de memória e dispositivos de entrada e saída.

# Múltiplos Processadores

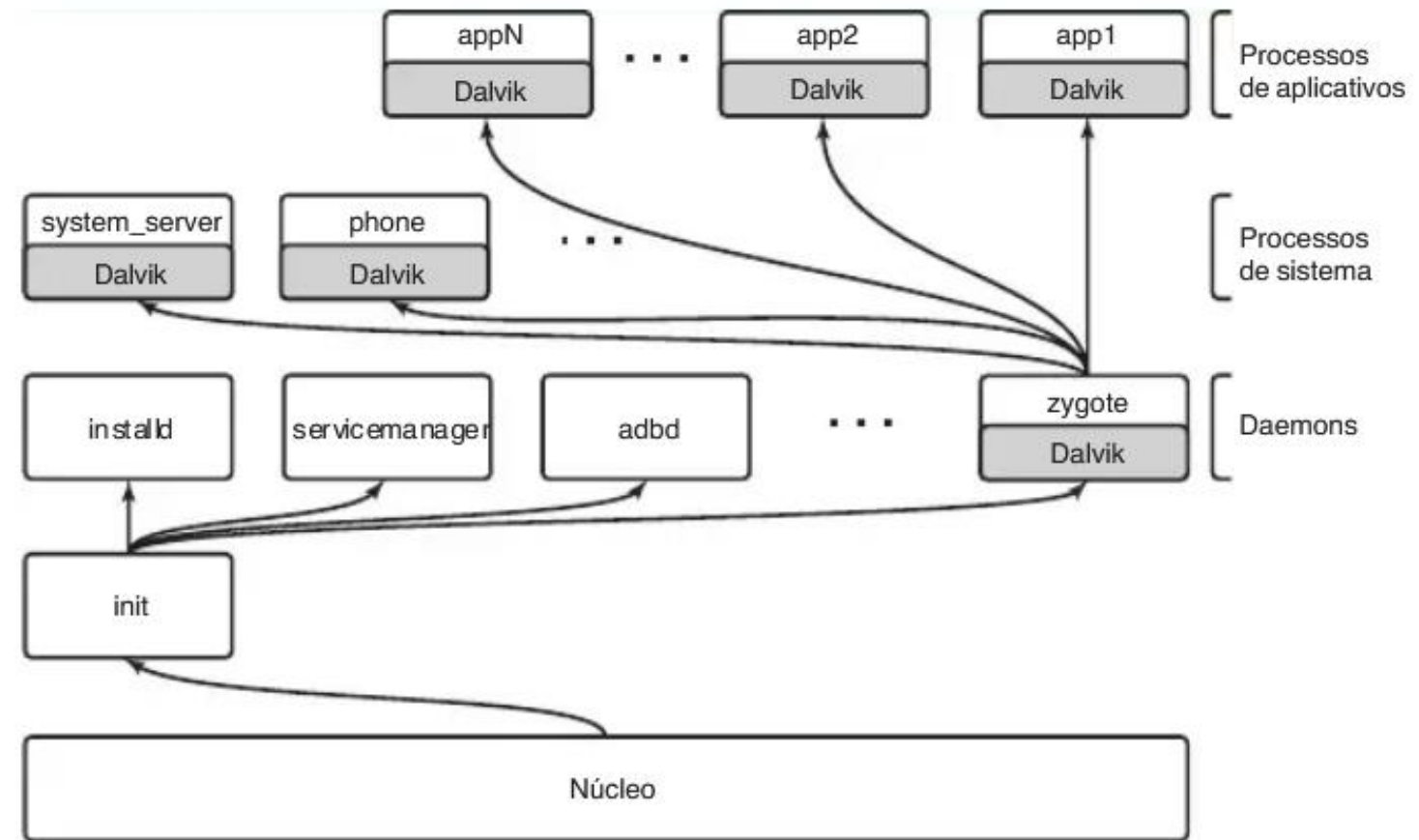
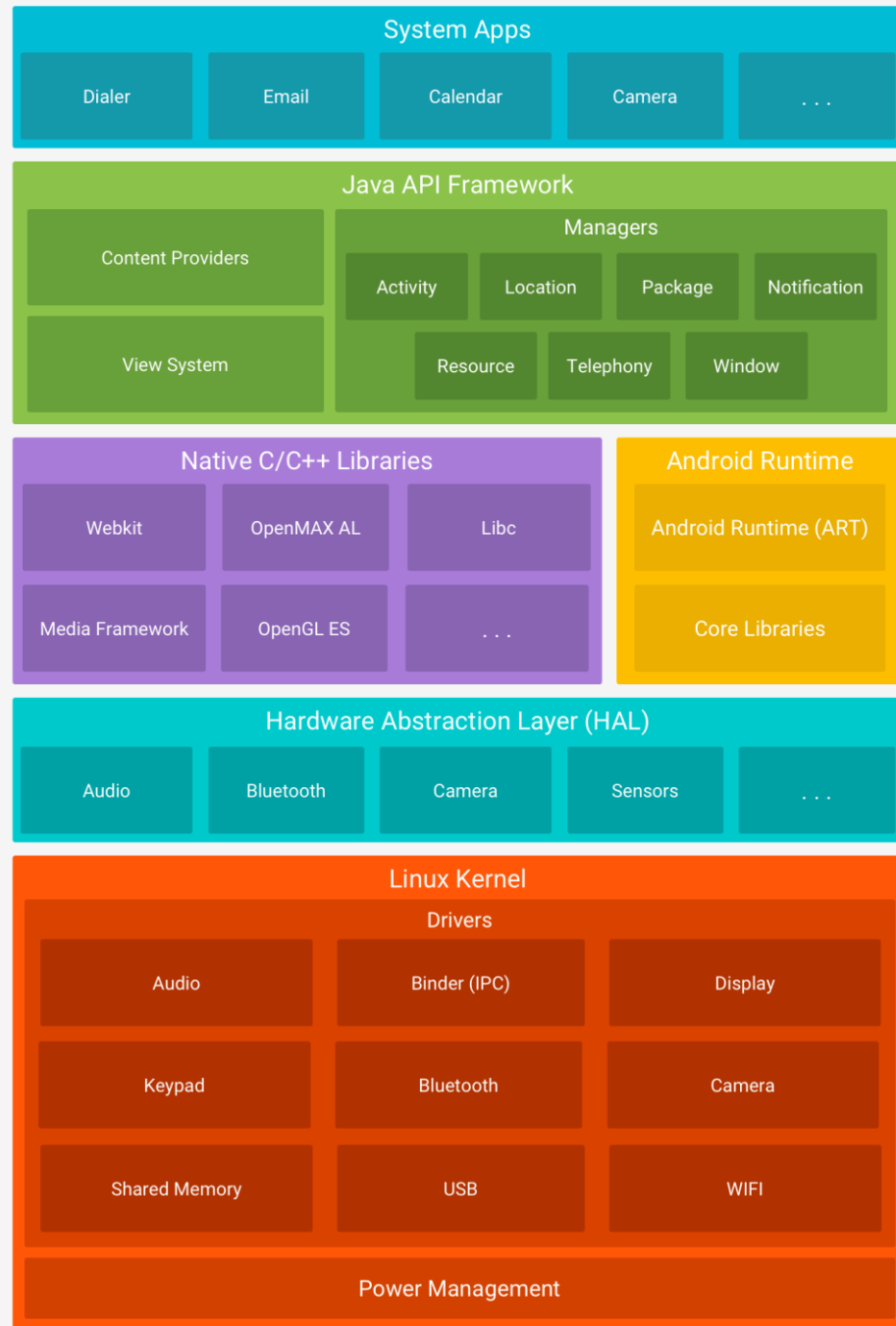
Esse sistema pode ser definido como **fortemente** ou **fracamente** **acoplado**.

- O sistema **fortemente acoplado** pode ser **simétrico** (quando dois processadores compartilham uma única memória) e **assimétrico** (quando um processador é definido como mestre, e o segundo processador, escravo). Qualquer operação será determinada pelo **processador mestre**: caso este falhe, nada acontece; caso precise de mais recursos, estes serão solicitados ao escravo.
- O sistema **fracamente acoplado** tem a característica de **dois ou mais sistemas de computação interligados**: rede ou distribuído. Cada sistema possui seu próprio sistema computacional: memória, processador, armazenamento, dispositivos de entrada e saída.

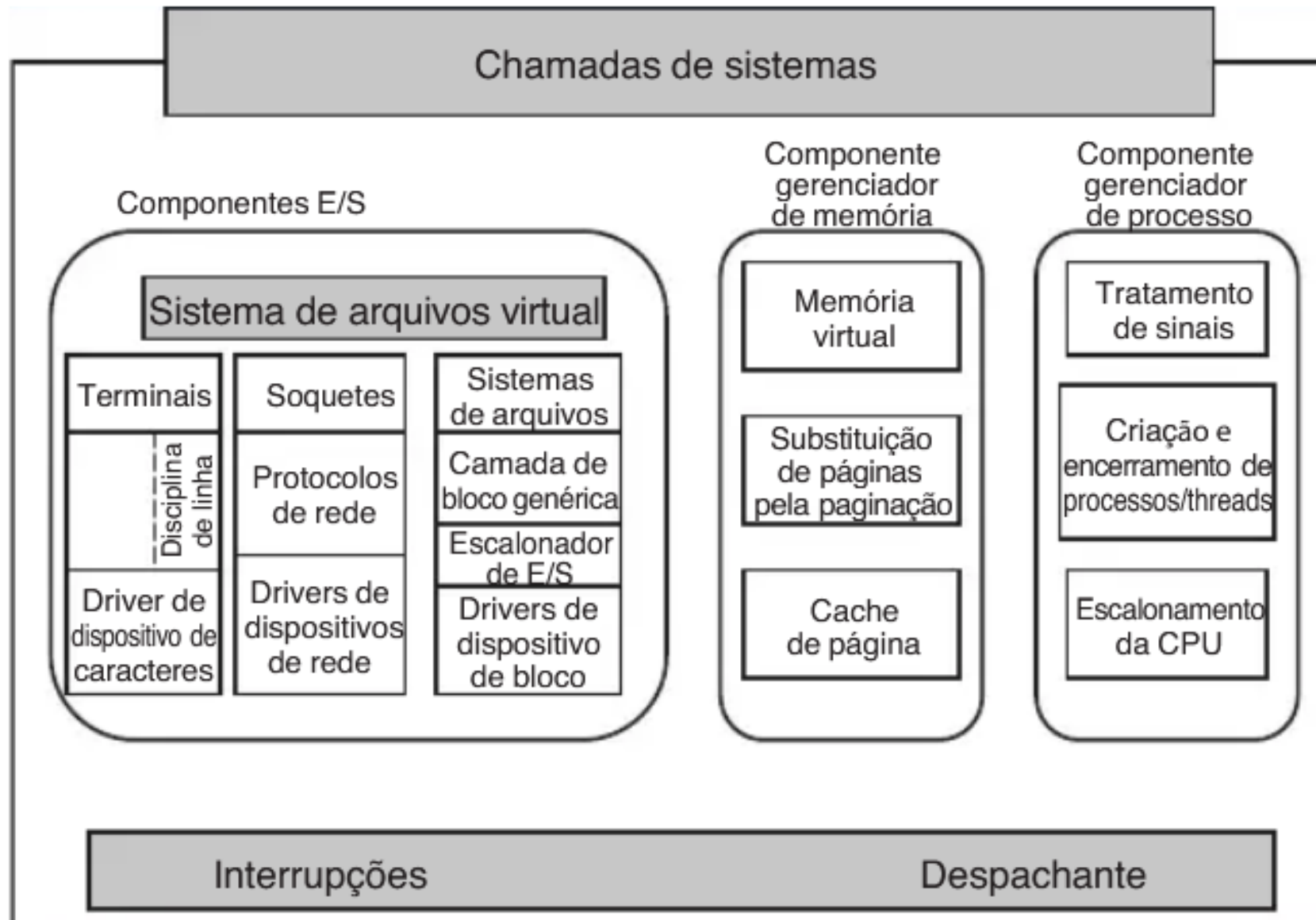
## Sistemas Operacionais e suas características

Sistema operacional	Características
Windows	Multitarefa Multiusuário Interface gráfica padronizada Licença para uso Baseado em bibliotecas (.dll) Empresa: Microsoft
iOS	Multitarefa Multiusuário Interface gráfica padronizada Licença para uso Empresa: Apple
Android	Licença de uso: GPL/GNU Interface Baseado em <i>kernel</i> Linux Empresa: Google
Linux	Multitarefa Multiusuário Licença de uso: GPL/GNU Interface variada Baseado em arquivo texto (.txt)

# Arquitetura Android

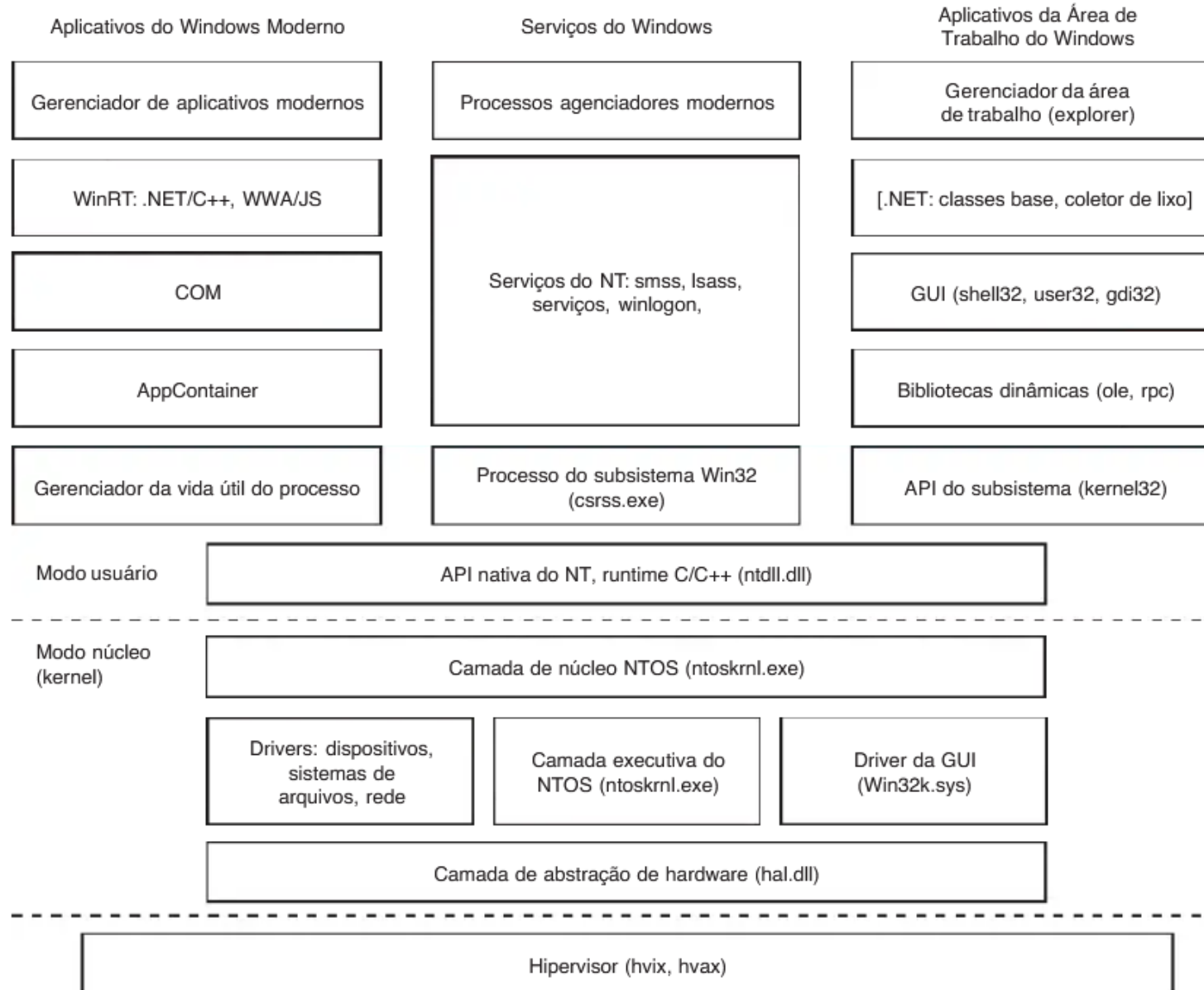


# Arquitetura Linux





# Arquitetura Windows



# Referências

OLIVEIRA, R.; CARISSIMI, A; TOSCANI, S: **Sistemas Operacionais**

PATTERSON, D.A., HENNESSY, J.L., **Organização e Projeto de Computadores**, Editora LTC, Quarta Edição, 2014.

SILBERSCHATZ, A. et al: Fundamentos de Sistemas Operacionais

TANENBAUM, Andrew S., **Organização Estruturada de Computadores**, Prentice Hall, Quinta Edição.

MONTEIRO, M. A., **Introdução à Organização de Computadores**, Editora LTC.