-

Sistemas Operacionais: Conceitos e Fundamentos

Este material apresenta os conceitos fundamentais de sistemas operacionais, abordando suas funcionalidades, categorias e evolução histórica. Veremos como os sistemas operacionais atuam como intermediários entre o hardware e os aplicativos, proporcionando abstração e gerência de recursos.

Agenda

01

Introdução e Conceitos Básicos

Definição de sistema operacional e seus objetivos fundamentais

02

Objetivos de um SO

Abstração e gerência de recursos

03

Funcionalidades

Gerência de processador, memória, dispositivos, arquivos e proteção

04

Categorias de SOs

Tipos de sistemas operacionais e suas características

05

Evolução Histórica

Marcos importantes no desenvolvimento dos sistemas operacionais

O que é um Sistema Operacional?

Um sistema de computação é constituído basicamente por **hardware** e **software**:

- O **hardware** inclui circuitos eletrônicos (processador, memória, portas de E/S) e periféricos (teclados, mouses, discos, etc.)
- O **software de aplicação** representa programas destinados ao usuário final (editores de texto, navegadores, jogos)

Entre os aplicativos e o hardware reside uma camada de software complexa denominada **Sistema Operacional (SO)**, que atua como intermediário entre ambos.

Estrutura de um Sistema de Computação

A figura ilustra a estrutura geral de um sistema de computação típico, mostrando a relação entre hardware, sistema operacional e programas aplicativos. O sistema operacional atua como uma camada intermediária que gerencia recursos e fornece abstrações para os aplicativos.

Objetivos Fundamentais de um SO

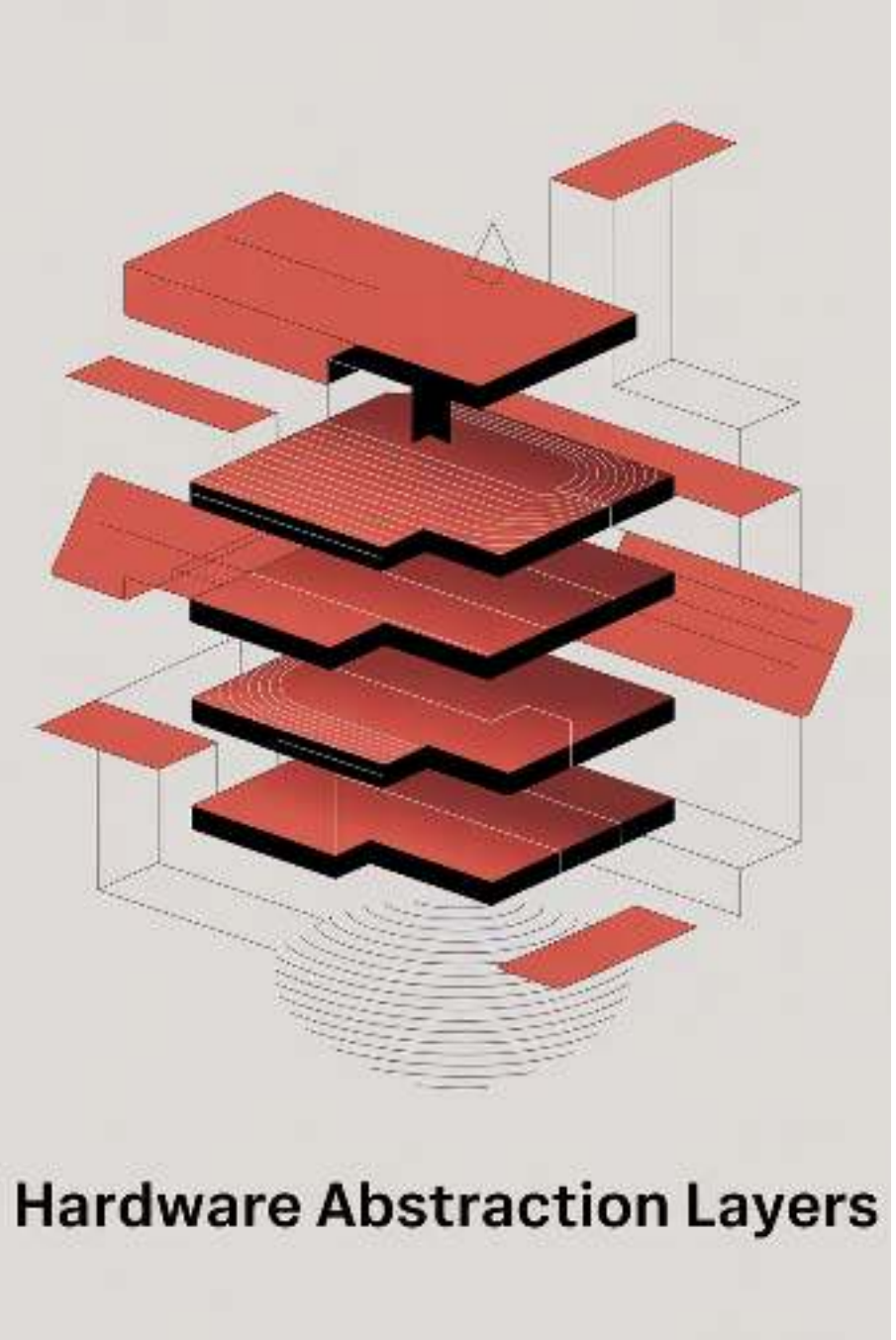
Abstração

Criar interfaces simplificadas para acesso aos recursos de hardware, ocultando a complexidade dos dispositivos físicos e suas diferenças tecnológicas.

Gerência

Definir políticas para o uso compartilhado dos recursos de hardware, resolvendo conflitos entre aplicativos e garantindo o uso eficiente dos recursos disponíveis.

Estes dois objetivos fundamentais norteiam todas as funcionalidades implementadas pelos sistemas operacionais modernos.



Abstração de Recursos

A abstração de recursos visa:

- Prover interfaces de acesso aos dispositivos mais simples que as interfaces de baixo nível
- Tornar os aplicativos independentes do hardware
- Definir interfaces de acesso homogêneas para dispositivos com tecnologias distintas

Exemplo: Para ler dados de um disco, um programador usa o conceito de **arquivo**, acessível através de operações como `open`, `read` e `close`, em vez de manipular diretamente portas de E/S e registradores.

Exemplo de Abstração: Acesso a Arquivos

Sem abstração (baixo nível)

1. Verificar parâmetros informados
2. Verificar disponibilidade do disco
3. Ligar motor do disco e aguardar velocidade correta
4. Posicionar cabeça de leitura sobre a trilha da tabela de diretório
5. Ler tabela de diretório e localizar o arquivo
6. Mover cabeça para posição do bloco inicial
7. Ler o bloco e depositá-lo em buffer de memória

Com abstração (alto nível)

O programador simplesmente usa:

```
arquivo = open("documento.txt");
dados = read(arquivo, tamanho);
close(arquivo);
```

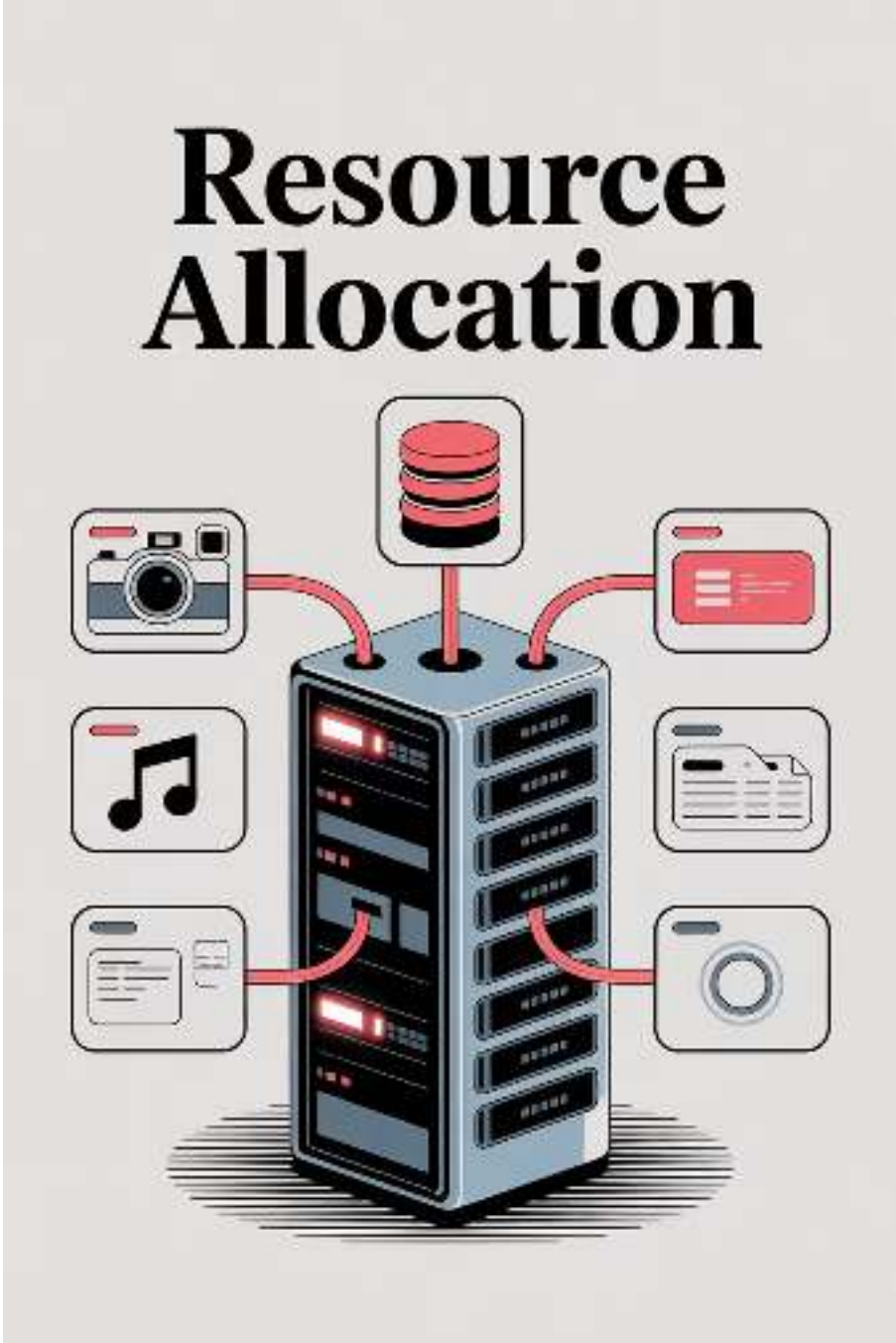
O sistema operacional se encarrega de todos os detalhes de baixo nível, independentemente da tecnologia do dispositivo (disco SATA, USB, CD, rede, etc.).

Gerência de Recursos

O sistema operacional define **políticas** para gerenciar o uso dos recursos de hardware pelos aplicativos, resolvendo disputas e conflitos.

- Distribuição justa de processadores entre tarefas em execução
- Alocação adequada de memória RAM para cada aplicação
- Acesso mutuamente exclusivo a recursos como impressoras
- Proteção contra monopolização de recursos (ex: ataques DoS)

A gerência de recursos garante estabilidade, segurança e desempenho adequado para todas as aplicações no sistema.



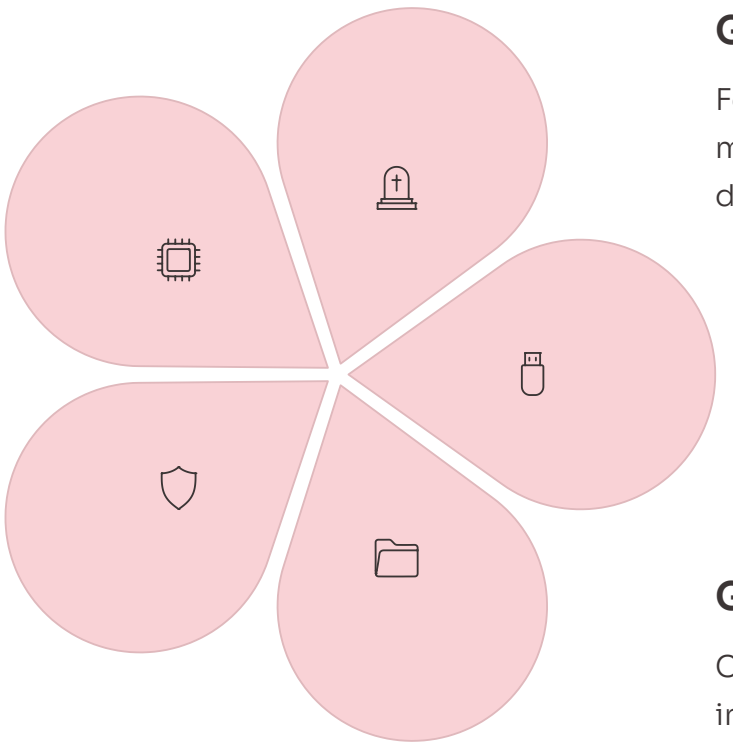
Funcionalidades de um SO

Gerência do Processador

Distribui a capacidade de processamento entre aplicações, criando a ilusão de processadores independentes para cada tarefa.

Gerência de Proteção

Define claramente os recursos que cada usuário pode acessar e as formas de acesso permitidas.



Gerência de Memória

Fornece a cada aplicação uma área de memória própria, independente e isolada das demais aplicações.

Gerência de Dispositivos

Implementa a interação com cada dispositivo por meio de drivers e cria modelos abstratos para acesso.

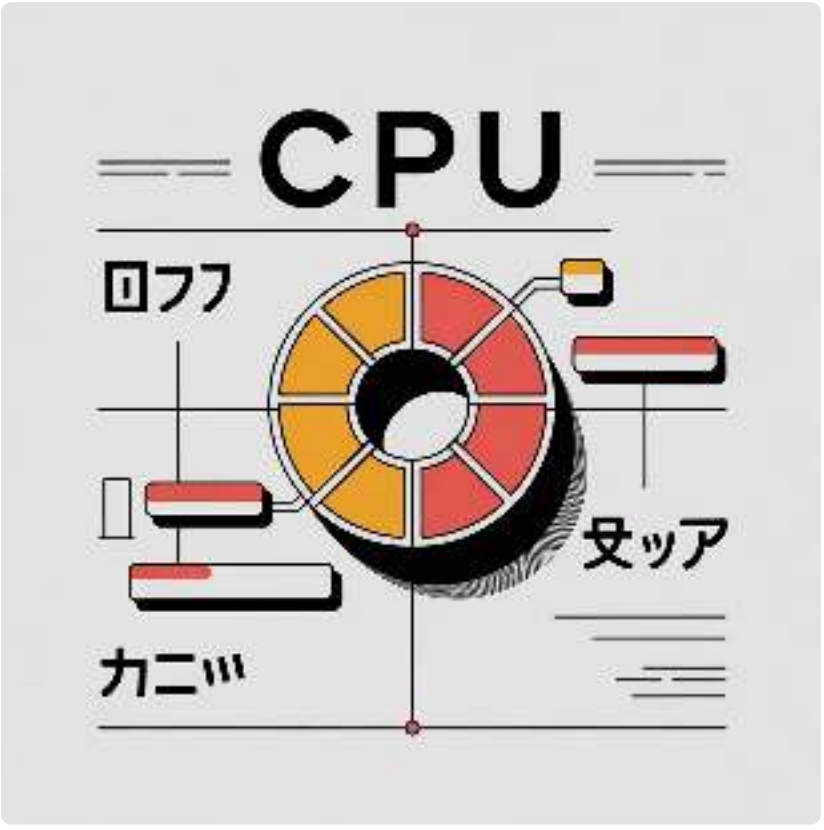
Gerência de Arquivos

Cria arquivos e diretórios, definindo sua interface de acesso e regras para seu uso.

Gerência do Processador

Também conhecida como gerência de processos, tarefas ou atividades, visa:

- Distribuir a capacidade de processamento de forma justa entre as aplicações
- Evitar que uma aplicação monopolize o processador
- Respeitar as prioridades definidas pelos usuários
- Prover a ilusão de que existe um processador independente para cada tarefa
- Fornecer abstrações para sincronizar tarefas interdependentes
- Prover formas de comunicação entre tarefas



A distribuição é feita de forma justa, mas não necessariamente igual, pois as aplicações têm demandas distintas de processamento.

Gerência de Memória

Isolamento

Fornece a cada aplicação uma área de memória própria, independente e isolada das demais aplicações e do sistema operacional.

Segurança

O isolamento melhora a estabilidade e segurança do sistema, impedindo que aplicações com erros ou maliciosas interfiram no funcionamento de outras.

Memória Virtual

Desvincula os endereços vistos por cada aplicação dos endereços físicos da memória RAM, permitindo que uma aplicação seja carregada em qualquer posição livre.

Expansão

Caso a memória RAM seja insuficiente, o sistema pode aumentá-la usando espaço em disco (swap), de forma transparente às aplicações.

Gerência de Dispositivos

Cada periférico do computador possui suas particularidades, mas existem problemas e abordagens em comum para o acesso aos periféricos.

A gerência de dispositivos (ou gerência de entrada/saída):

- Implementa a interação com cada dispositivo por meio de **drivers**
- Cria modelos abstratos que permitem agrupar dispositivos similares sob a mesma interface de acesso

Exemplo: Criar uma abstração única para dispositivos de armazenamento como:

- Pendrives
- Discos SATA ou IDE
- CDROMs
- Cartões de memória

Todos são vistos como um vetor de blocos de dados, independentemente de suas diferenças tecnológicas.

Gerência de Arquivos

Esta funcionalidade é construída sobre a gerência de dispositivos e visa criar arquivos e diretórios, definindo sua interface de acesso e as regras para seu uso.

Arquivos

Abstrações que representam conjuntos de dados armazenados, acessíveis através de operações como open, read, write e close.

Diretórios

Estruturas que organizam arquivos hierarquicamente, facilitando sua localização e acesso.

Além do armazenamento

Em muitos sistemas, arquivos e diretórios são usados para acessar recursos que não são de armazenamento, como conexões de rede e informações internas do sistema.

No sistema experimental *Plan 9*, por exemplo, todos os recursos do sistema operacional são vistos como arquivos.

Gerência de Proteção

Com computadores conectados em rede e compartilhados por vários usuários, é importante definir claramente os recursos que cada usuário pode acessar e as formas de acesso permitidas.

Definição

Definir usuários e grupos de usuários no sistema

Autenticação

Identificar os usuários que se conectam ao sistema através de procedimentos de verificação

Autorização

Definir e aplicar regras de controle de acesso aos recursos

Auditoria

Registrar o uso dos recursos pelos usuários para fins de verificação e contabilização

A gerência de proteção é fundamental para garantir a segurança e a privacidade em sistemas multiusuários e conectados em rede.

Política vs. Mecanismo

Política

Aspectos de decisão mais abstratos, que podem ser resolvidos por algoritmos de nível mais alto:

- Decidir a quantidade de memória para cada aplicação ativa
- Determinar qual o próximo pacote de rede a enviar
- Escolher qual processo deve receber o processador

Mecanismo

Procedimentos de baixo nível usados para implementar as políticas:

- Atribuir ou retirar memória de uma aplicação
- Enviar ou receber um pacote de rede
- Trocar o contexto de execução entre processos

Os mecanismos devem ser suficientemente genéricos para suportar mudanças de política sem necessidade de modificações. Esta separação traz grande flexibilidade aos sistemas operacionais.

Categorias de Sistemas Operacionais

Os sistemas operacionais podem ser classificados segundo diversos parâmetros e aspectos. Muitos sistemas se encaixam em mais de uma categoria.

1

Batch (de lote)

Processam tarefas sem interação direta com os usuários. Ex: IBM OS/360, VAX/VMS

2

De rede

Oferecem suporte à operação em rede, compartilhando recursos entre computadores

3

Distribuído

Os recursos estão disponíveis de forma transparente aos usuários. Ex: Amoeba, sistemas de nuvem

4

Multiusuário

Suportam a identificação do "dono" de cada recurso e impõem regras de controle de acesso

Estas são apenas algumas das categorias existentes. Veremos outras na próxima slide.

Mais Categorias de Sistemas Operacionais

1

Servidor

Permitem a gestão eficiente de grandes quantidades de recursos, impondo prioridades e limites de uso

2

Desktop

Voltados ao usuário doméstico e corporativo. Ex: Windows, MacOS, Linux

3

Móvel

Usados em equipamentos compactos, com foco em gestão de energia e conectividade. Ex: Android, iOS

4

Embarcado

Construídos para hardware com poucos recursos. Ex: LynxOS, TinyOS, Contiki, VxWorks

Cada categoria possui características específicas que atendem a diferentes necessidades e contextos de uso.

Sistemas de Tempo Real

São sistemas nos quais o tempo é essencial. Sua característica principal é ter um comportamento temporal **previsível**, com tempo de resposta conhecido no melhor e no pior caso.

Tempo Real Crítico

(Hard real-time) A perda de um prazo pode causar graves consequências humanas, econômicas ou ambientais.

Ex: Controle de turbina de avião, freio ABS

Tempo Real Não-Crítico

(Soft real-time) A perda de um prazo é perceptível e degrada o serviço, sem maiores consequências.

Ex: Software de reprodução de mídia

Evolução Histórica dos Sistemas Operacionais

1

Anos 40

Sem sistema operacional: programas executavam diretamente sobre o hardware, controlando todos os aspectos do computador.

2

Anos 50

Surgem "bibliotecas de sistema" para facilitar o acesso aos periféricos e programas "monitores" para auxiliar na carga e descarga de aplicações.

3

1961

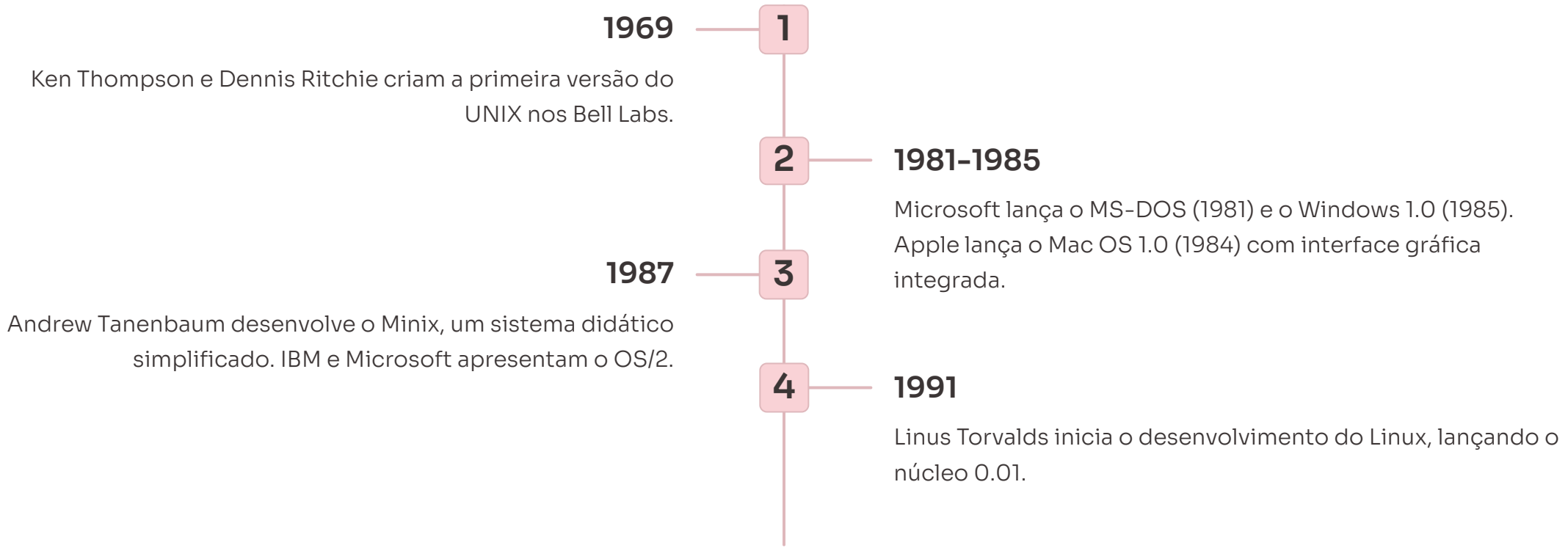
CTSS (Compatible Time-Sharing System) - primeiro sistema com compartilhamento de tempo, desenvolvido no MIT.

4

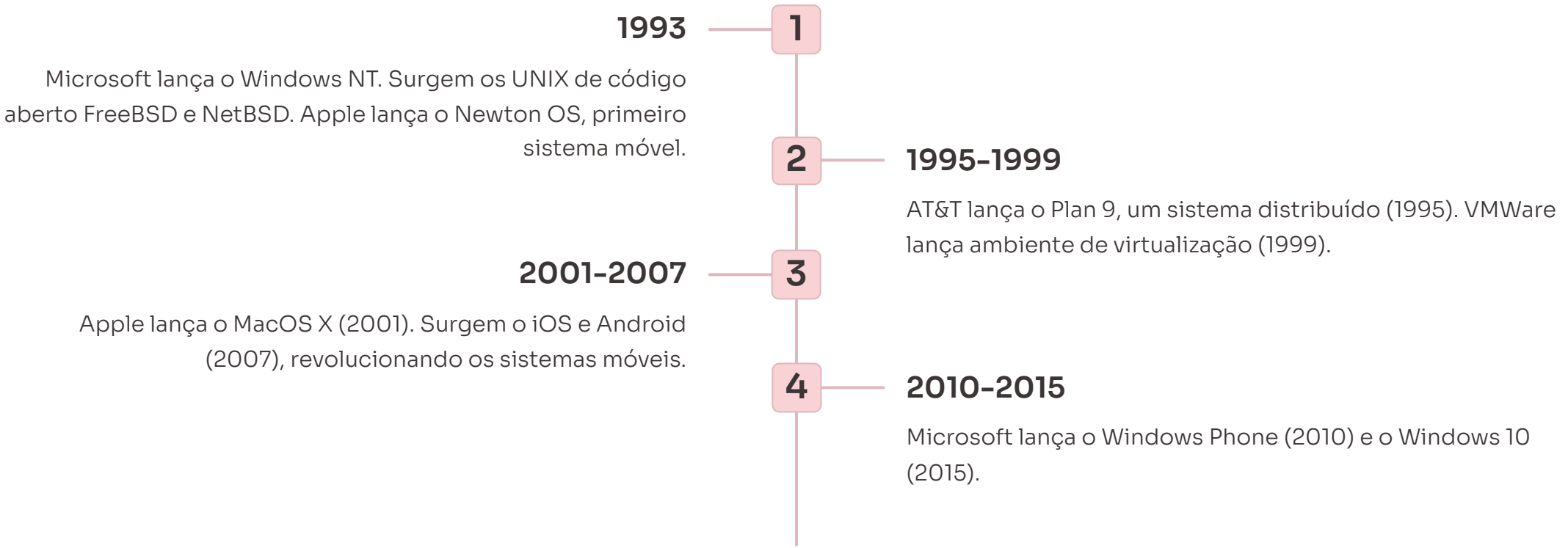
1965

IBM lança o OS/360 e inicia-se o projeto Multics (MIT, GE e Bell Labs), que influenciaria sistemas futuros.

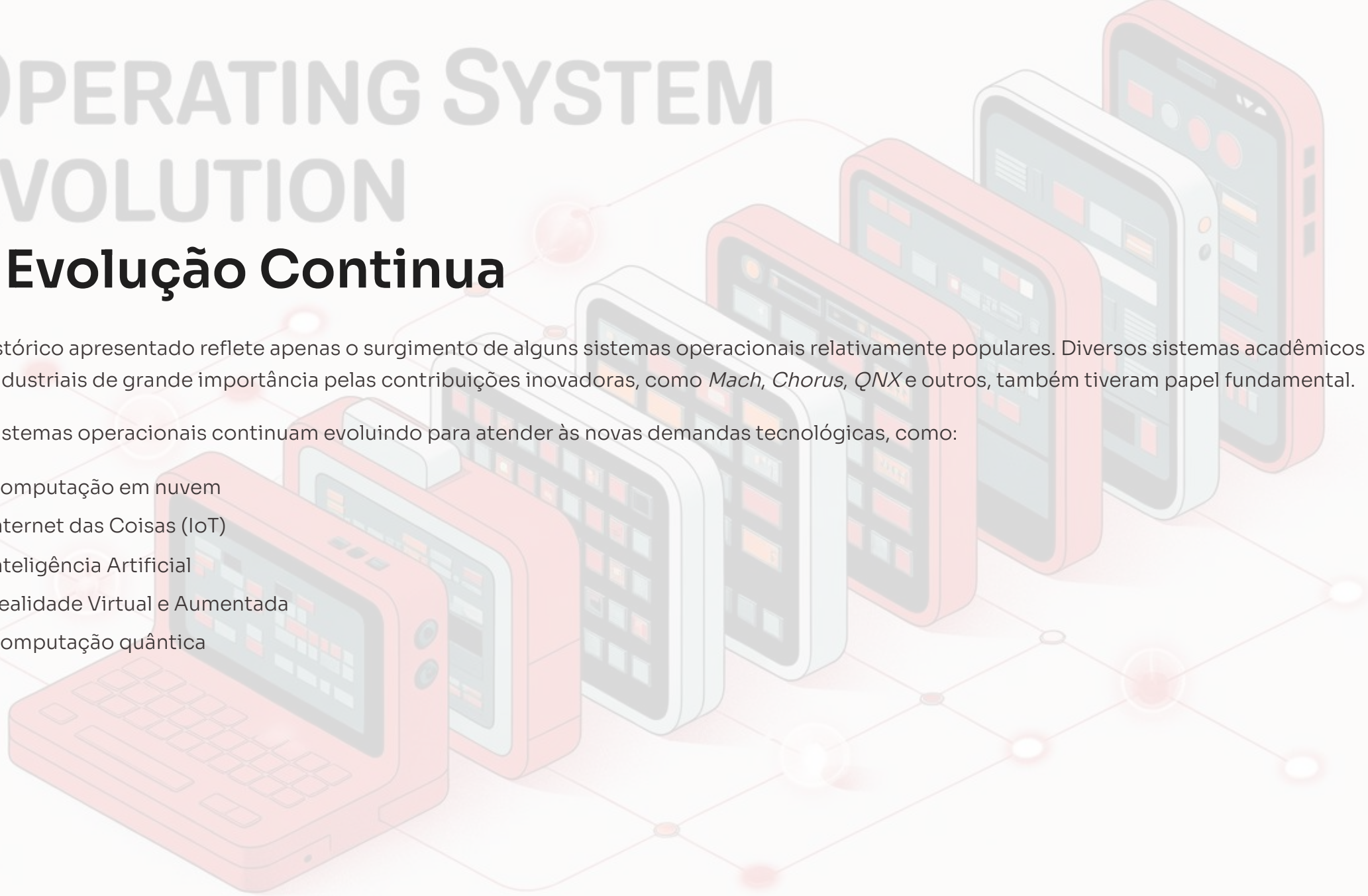
Evolução Histórica (continuação)



Evolução Histórica (continuação)



A Evolução Continua



O histórico apresentado reflete apenas o surgimento de alguns sistemas operacionais relativamente populares. Diversos sistemas acadêmicos ou industriais de grande importância pelas contribuições inovadoras, como *Mach*, *Chorus*, *QNX* e outros, também tiveram papel fundamental.

Os sistemas operacionais continuam evoluindo para atender às novas demandas tecnológicas, como:

- Computação em nuvem
- Internet das Coisas (IoT)
- Inteligência Artificial
- Realidade Virtual e Aumentada
- Computação quântica

Pontos-Chave sobre Sistemas Operacionais

Camada Intermediária

O sistema operacional atua como intermediário entre hardware e aplicativos, abstraindo complexidades e gerenciando recursos.

Objetivos Fundamentais

Abstração (simplificar interfaces) e gerência (distribuir recursos) são os dois pilares que sustentam todas as funcionalidades do SO.

Funcionalidades Interdependentes

Gerência de processador, memória, dispositivos, arquivos e proteção são funcionalidades que trabalham em conjunto e são interdependentes.

Diversidade de Categorias

Existem diversos tipos de sistemas operacionais, cada um com características específicas para atender diferentes necessidades e contextos.

Exercícios para Fixação

1. Quais os dois principais objetivos de um sistema operacional?
2. Por que a abstração de recursos é importante para os desenvolvedores de aplicações? Ela tem alguma utilidade para os desenvolvedores do próprio sistema operacional?
3. A gerência de tarefas permite compartilhar o processador, executando mais de uma aplicação ao mesmo tempo. Identifique as principais vantagens trazidas por essa funcionalidade e os desafios a resolver para implementá-la.

1. O que caracteriza um sistema operacional de tempo real? Quais as duas classificações de sistemas operacionais de tempo real e suas diferenças?
2. Relacione afirmações aos respectivos tipos de sistemas operacionais: distribuído, multi-usuário, desktop, servidor, embarcado ou de tempo-real.
3. Identifique afirmações incorretas sobre os diversos tipos de sistemas operacionais.

Estes exercícios ajudarão a consolidar os conceitos apresentados e a verificar sua compreensão sobre os fundamentos dos sistemas operacionais.

Obrigado!

Referências

- A. Arpaci-Dusseau, R. Arpaci-Dusseau, et al. Transforming policies into mechanisms with InfoKernel. In *19th ACM Symposium on Operating Systems Principles*, October 2003.
- Corbató et al. Compatible Time-Sharing System, 1962.
- Tanenbaum et al. Amoeba, 1991.
- Pike et al. Plan 9, 1993.
- Rashid et al. Mach, 1989.
- Rozier and Martins. Chorus, 1987.

Para mais informações, consulte a bibliografia recomendada e os recursos online disponíveis.