

Sistemas Operacionais - 2025/1

Prof. Pedro Ramos
pramos.costar@gmail.com

Pontifícia Universidade Católica de Minas Gerais
ICEI - Departamento de Ciência da Computação

Avaliação / Datas importantes

- Cronograma e plano de ensino no Canvas
- Debate ainda será divulgado
- “Reservas” no cronograma -> Haverá aula!

Ao Longo do Semestre	Exercícios	15 Pts
22/04	PROVA 1	30 Pts
24/06	DEBATE de SO's	20 Pts
01/07	PROVA 2	30 Pts
04/07	PROVA 2a CHAMADA	30 Pts
08/07	REAValiação	60 Pts
??/??	ADA	5 Pts

Livro Texto: Livro do Tanenbaum

Sistemas operacionais modernos.

TANENBAUM, Andrew.

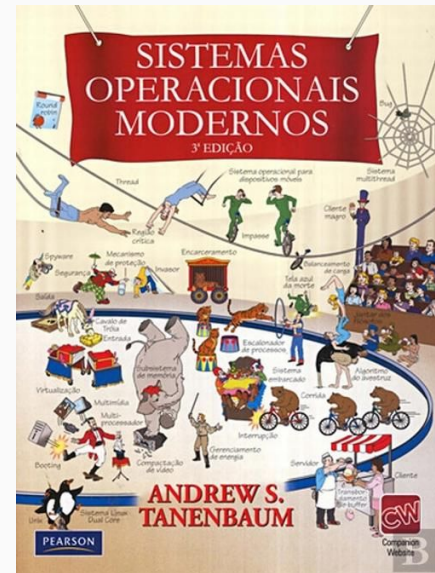
Outros livros:

- SILBERSCHATZ, Abraham et al. Fundamentos de sistemas operacionais: princípios básicos
- MACHADO, Francis; MAIA, Luiz Paulo. Arquitetura de Sistemas Operacionais
- CARISSIMI, Alexandre et al. Sistemas operacionais

Outros links úteis:

Harvard Course - <https://read.seas.harvard.edu/cs161/2020/>

Stanford Course - <https://web.stanford.edu/~ouster/cgi-bin/cs140-spring20/index.php>



CONTEÚDO DO CURSO

Aulas iniciais: nivelamento e revisão de arquitetura

- PROCESSOS E THREADS
 - Gerenciamento de processos e threads
 - Escalonamento
- GERENCIAMENTO DE MEMÓRIA
 - Como o OS faz o layout da memória virtual em memória física
- ARMAZENAMENTO E SISTEMAS DE ARQUIVOS
 - Tradicional - disco rígido (válido para outros dispositivos)
- SISTEMAS DISTRIBUÍDOS

O QUE É UM SISTEMA OPERACIONAL?

- ???

O QUE É UM SISTEMA OPERACIONAL?

- A definição mudou ao longo dos anos
- Sistema Operacional (SO):
 - INTERFACE entre o usuário e a arquitetura
 - Implementa uma máquina virtual que é (espera-se) mais fácil de programar do que hardware bruto
- O que é uma **máquina virtual**?

O QUE É UM SISTEMA OPERACIONAL?

- A definição mudou ao longo dos anos
- Sistema Operacional (S0):
 - INTERFACE entre o usuário e a arquitetura
 - Implementa uma máquina virtual que é (espera-se) mais fácil de programar do que hardware bruto
- O que é uma **máquina virtual**?
 - Quando você escreve um programa, você não faz ele para sua máquina específica - você escreve um programa para rodar no sistema operacional.
 - A máquina que o S0 apresenta para o usuário é + poderosa que a real

O QUE É UM SISTEMA OPERACIONAL?

- Nos anos 90, a MS argumentava que muitas coisas faziam parte de um SO.
 - Navegador?
 - Media player?



O QUE É UM SISTEMA OPERACIONAL?

- Nos anos 90, a MS argumentava que muitas coisas faziam parte de um SO.
 - Navegador?
 - Media player?

A MS começou a entregar SO's com várias aplicações embutidas.

- ChromeOS: um navegador-SO.
 - *Isso torna o navegador parte de um SO?*



O QUE É UM SISTEMA OPERACIONAL?

VISÃO TRADICIONAL:

interface entre usuário e arquitetura

- esconde detalhes da arquitetura!

máquina virtual

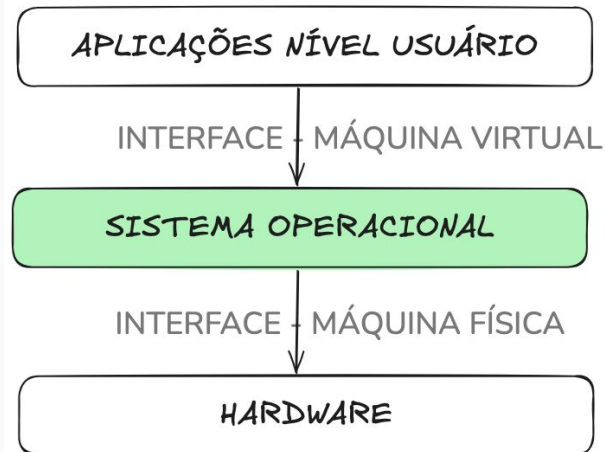
- + fácil de programar do que hardware bruto

ilusionista

- maior, mais rápido, confiável

governo

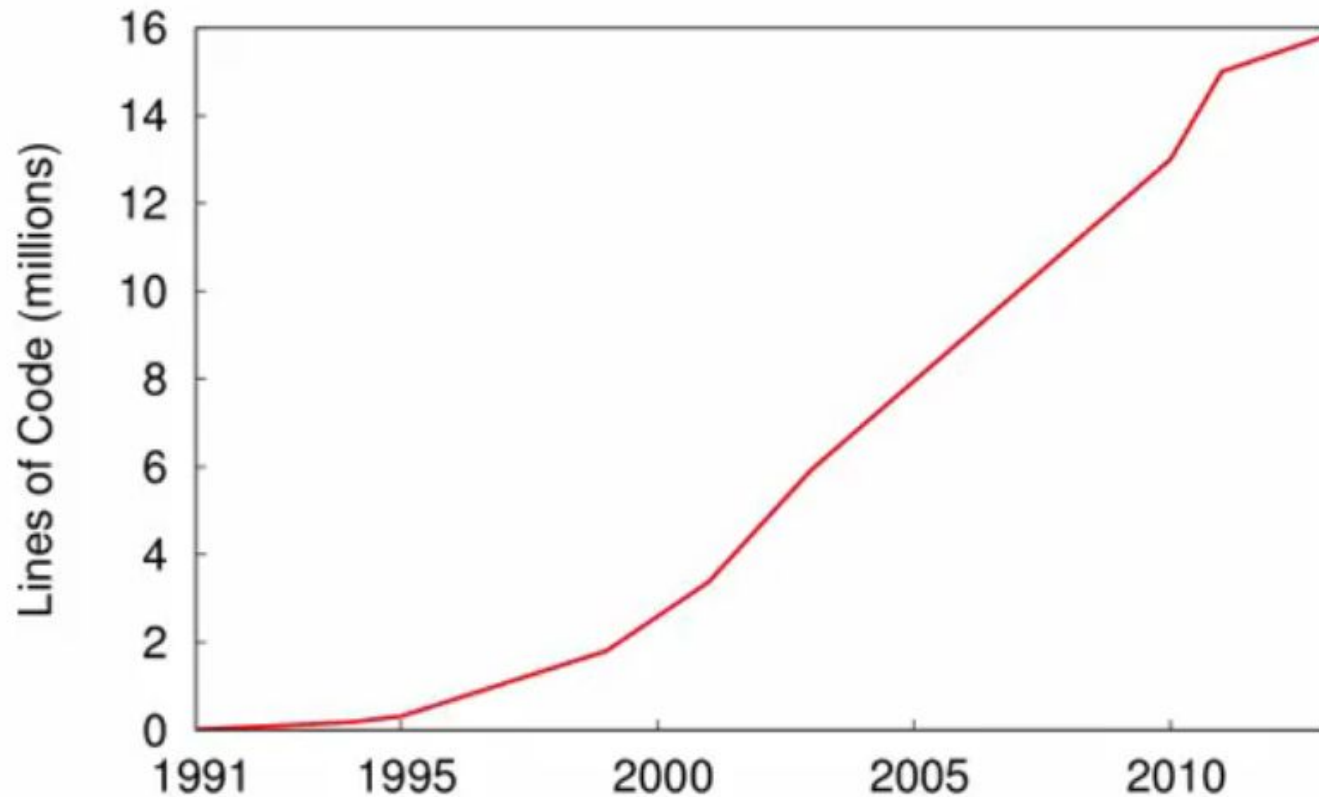
- divide recursos entre programas que competem
- cobra um imposto/taxa = *overhead*



SISTEMAS OPERACIONAIS HOJE

- Demandas em SO's estão crescendo
- Novos espaços para aplicações (web, cloud)
- Hardware evoluindo rapidamente
- 50 anos atrás:
 - Linux - SO de código aberto
- Você pode contribuir para um SO!
- SO's móveis (Android e iOS)

TAMANHO DO KERNEL DO LINUX



FUNCIONALIDADES DE UM SISTEMA OPERACIONAL

- **SERVIÇOS**

- O SO provê serviços “padrões” (a interface) que são implementados pelo hardware.
 - Sistema de arquivos e comunicação com o disco
 - Memória virtual
 - Redes
 - Escalonamento de CPU

- **COORDENAÇÃO**

- O SO coordena múltiplas aplicações e usuários para atingir **imparcialidade** e **eficiência** (vazão).
 - concorrência, proteção de memória, redes, segurança

FUNCIONALIDADES DE UM SISTEMA OPERACIONAL

- **SERVIÇOS**

- O SO provê serviços “padrões” (a interface) que são implementados pelo hardware.
 - Sistema de arquivos e comunicação com o disco
 - Memória virtual
 - Redes
 - Escalonamento de CPU

- **COORDENAÇÃO**

- O SO coordena múltiplas aplicações e usuários para atingir **imparcialidade** e **eficiência** (vazão).
 - concorrência, proteção de memória, redes, segurança



trade-off

PORQUÊ ESTUDAR SISTEMAS OPERACIONAIS?

- **abstração**
 - como o SO dá ao usuário a impressão de ter memória infinita, múltiplas CPUs, recursos, computação na nuvem?
- **projeto de sistemas**
 - como fazer tradeoffs entre
 - performance e conveniência do SO;
 - performance e simplicidade do SO;

O SO É O PONTO DE INTERSECÇÃO ENTRE SOFTWARE E HARDWARE.

CONSTRUINDO SISTEMAS LARGOS

O S0 é um exemplo de um sistema largo.

OBJETIVO: RÁPIDO. CONFIÁVEL. ESCALÁVEL.

Para construir esse tipo de sistema, precisamos saber:

- Sobre cada computador:
 - Detalhes arquiteturais
 - C/C++
 - Gerenciamento de memória e localidade
 - Concorrência e escalonamento
 - Discos, redes, sistemas de arquivos
- Através de *clusters* de computadores:
 - Arquiteturas de servidor
 - Computação distribuída, sistemas de arquivos distr.

CONSTRUINDO SISTEMAS LARGOS

O SO é um

OBJETIVO

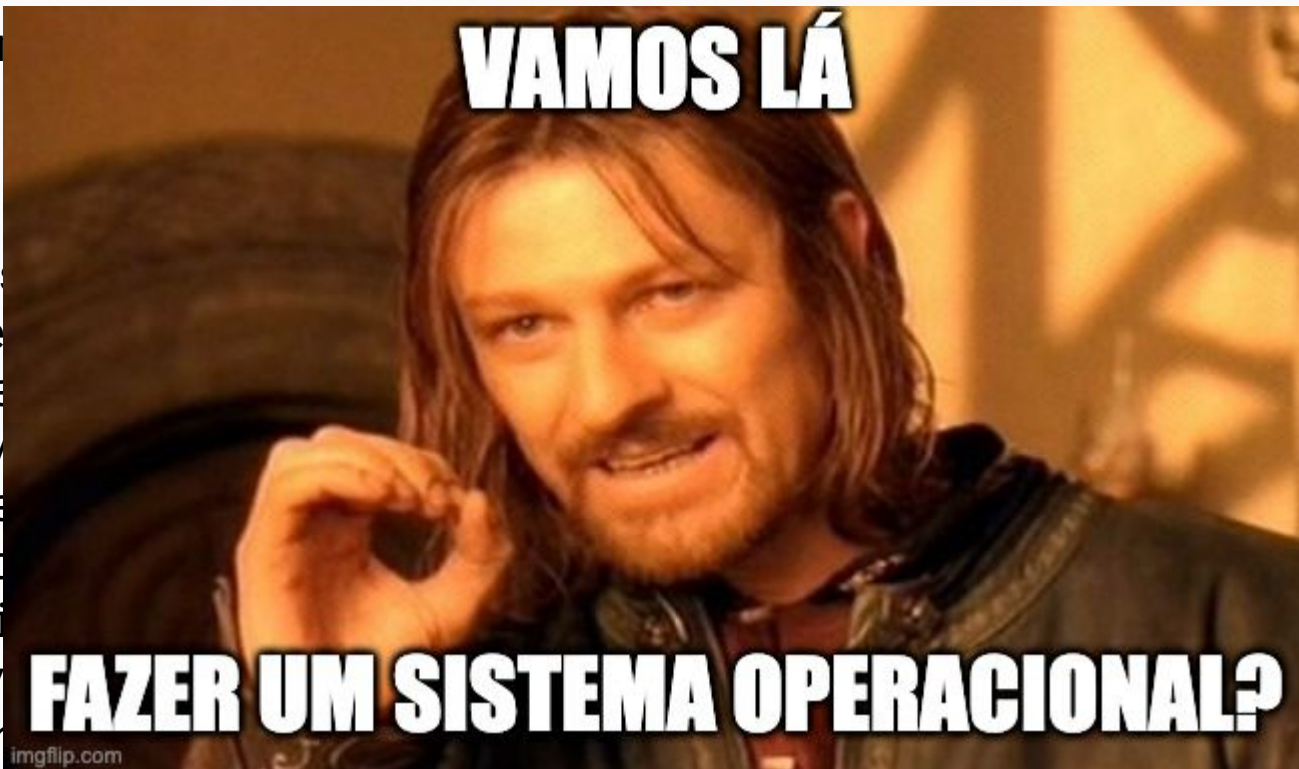
Para cons

- Sobre

- De
- C/
- Ge
- Co
- Di

- Atrav

- Ar
- Computação distribuída, sistemas de arquivos distr.



HISTÓRIA

- De mainframes à sistemas web em 9 slides

1. COMPUTADOR DE USO ÚNICO

- HARDWARE: **caro**, “MÃO DE OBRA” HUMANA: **barata**
- um usuário por vez em 1 único console
 - interagindo com o programa enquanto ele roda
 - speedcoding (John Backus 1953)
- computador **executa 1 função por vez**
 - **não tem sobreposição de computação com I/O** (entrada/saída)
- usuário precisa estar no console para debugar
- múltiplos usuários = uso ineficiente da máquina

2. PROCESSAMENTO EM “BATCH”

- Executar múltiplos trabalhos em “batch”:
 - Carregar o programa
 - Rodar
 - Printar resultados, dump do estado da máquina
 - Repetir
- Usuários submetiam trabalhos (programas) em cartões ou fita
- Humano agenda e organiza a sequência de programas
- O SO carrega e roda esses programas todos de uma só vez

... melhorou, mas ainda é limitado.

3. I/O (ENTRADA/SAÍDA) & COMPUTAÇÃO AO MESMO TEMPO

Antes: máquina espera I/O terminar para poder começar a executar. -> *Ineficiente pois a CPU tinha tempo ocioso.*

Agora:

- CPU executa outras coisas enquanto espera I/O
- Adição de "buffer"
 - Espaço a ser preenchido com dados antes da saída (output)
- sistema de interrupções
 - eventos de entrada/saída (I/O) disparam um sinal de interrupção

... melhorou, mas ainda é um programa por vez.

4. MULTIPROGRAMAÇÃO

- Familiar até hoje: **executar vários programas ao mesmo tempo.**
- Ideia básica:
 - Executa 1 programa até I/O;
 - Executa outro programa até I/O; etc...;
- O SO gerencia as **interações entre I/O e os programas.**
 - Qual programa rodar (**escalonamento**)
 - **Protege a memória** do programa dos outros programas
 - Decide qual programa **retorna quando a CPU está disponível**

... ok, mas o SO tá ficando complexo...

COMPLEXIDADE DE UM SO

- + funcionalidades, + complexidade
- Primeiras falhas:
 - Multics (GE & MIT)
 - anunciado em 1963 e lançado em 1969
 - OS/360 lançado com 1000 bugs conhecidos
- Daí a necessidade de tratar a projeção de um SO como algo científico - não somente um problema de engenharia a ser resolvido de qualquer forma.
- *Essa necessidade de que precisamos gerenciar a complexidade de um SO, nos levou a...*

5. O RENASCIMENTO (1970's)

- HARDWARE: barato, MÃO DE OBRA HUMANA: cara
- Usuários compartilham o sistema via terminais
- A era UNIX
 - Multics:
 - exército de programadores, 6 anos
 - excessivamente grande e complexo
 - UNIX:
 - 3 programadores, 2 anos
 - menor, mais simples
 - "shell": comandos que podem compor (cd / | ls | grep)
- PORÉM: Tempo de resposta longo, e fenômeno *thrashing*

6. REVOLUÇÃO INDUSTRIAL DOS ANOS 80

- HARDWARE: muito barato
- MÃO DE OBRA HUMANA: muito cara
- Uso dos PC's
 - IBM PC: 1981
 - Macintosh: 1984
- SO's simples: DOS e MacOS
- Não tinha multiprogramação, concorrência, proteção de memória, memória virtual, nada.
- Depois: adicionaram redes, compartilhamento de arquivos, imprimir remotamente
- GUI + SO == "WIMP"

7. ERA MODERNA (1990 - 2010)

- Demandas de processamento aumentam
- SO's "reais" em PC's
 - NT(1991), Mac OS X, Linux
- SO's estão presentes em diversas modalidades:
 - **tempo-real**: operadores de avião e torres de controle.
 - **sensores e sistemas embarcados** - computadores com pouco poder computacional, pouca bateria
 - **paralelismo** - múltiplos processadores em uma mesma máquina
 - **distribuído** - múltiplos processadores através da rede (nuvem)

TENDÊNCIA DE ARQUITETURA

- EM 50 ANOS, 99% DE TODOS OS COMPONENTE EM TODOS OS COMPUTADORES SERÃO 9 ORDENS DE MAGNITUDE MAIS RÁPIDOS, LARGOS E BARATOS

Isso é sem precedência em qualquer área de negócio.

Exemplos:

Transporte - em 200 anos viemos de 16km/h em cavalos para 1600km/h em aviões supersônicos

Comunicação - em 200 anos fomos da velocidade de **Transporte** para quase a velocidade da luz - 7 ordens de magnitude.

CONVERGÊNCIA (2010 -> Hoje)

- Lei de MOORE - estamos atingindo um platô
- Novas funcionalidades:
 - Múltiplos núcleos. NVIDIA RTX 5090 tem 20 mil núcleos a 2.2 GHZ
 - Memórias não confiáveis
 - SSD's
 - Limitações de energia, SO's ecologicamente amigáveis
 - IA' (Inteligência Artificial)
 - segurança, confiança, privacidade
 - AI-Agents? (*Precisaremos de mais memória RAM nos PCs?*)

Convergência entre GPU's (processamento), dados (web) e IA preditiva => qual tipo de SO surgirá?