

# Aula 7

## SOs e shells

---

# MACo216 - Técnicas de Programação I

Professores: Alfredo, Daniel, Fabio e Kelly

**Departamento de Ciência da Computação**  
**Instituto de Matemática e Estatística**



# 1.

# Introdução aos Sistemas Operacionais

# Softwares

*Podem ser divididos grosseiramente em dois grupos:*

- ▷ **Programas de sistema** - gerenciam o funcionamento do computador
- ▷ **Programas aplicativos** - executam o trabalho que o usuário quer que seja feito

O **sistema operacional** é o principal programa de sistema de um computador.

# Principais funções de um sistema operacional

- ▷ **Máquina estendida ou máquina virtual** - fornece ao usuário interfaces que facilitam a interação com o hardware (e.g., serviços que os programas podem acessar por meio de chamadas ao sistema)
- ▷ **Gerenciador de recursos** - gerencia todas as "peças" que constituem um computador moderno (processadores, memórias, interfaces de rede, dispositivos de entrada e saída, etc.), se preocupando em prover uma alocação ordenada e controlada dos recursos para os vários programas que os disputam

# Motivações históricas para o desenvolvimento dos SOs

- ▷ **Primeira fase** - hardware muito mais caro que a mão de obra especializada

**Objetivo:** maximizar a taxa de utilização do hardware

- ▷ **Segunda fase (inversão)** - hardware mais barato, mão de obra mais cara

**Objetivo:** tornar o computador mais fácil de ser usado pelas pessoas

# Evolução dos sistemas de computadores

- ▷ **Computadores a válvulas** - sistemas sem SO algum
- ▷ **Computadores a transistores** - sistemas em lote (batch)
- ▷ **Circuitos integrados** - sistemas multiprogramados, sistemas de tempo compartilhado
- ▷ **Computadores pessoais** - sistemas gráficos, sistemas de redes, sistemas distribuídos

# Multiprogramação × multiprocessamento

- ▷ **Multiprogramação (ou multitarefa)** - diversos programas distintos executando em um mesmo processador
- ▷ **Multiprocessamento** - diversos processadores (em um mesmo sistema computacional) executando programas distintos ou cooperando na execução de um mesmo programa

# Componentes centrais de um sistema operacional

- ▷ **Escalonador de processos** - determina quando e por quanto tempo um processo é executado em um processador
- ▷ **Gerenciador de memória** - determina quando e como a memória é alocada aos processos e o que fazer quando a memória principal estiver cheia
- ▷ **Gerenciador de E/S** - atende às solicitações de entrada/saída de/para dispositivos de hardware
- ▷ **Gerenciador de comunicação interprocessos (IPC)** - permite que os processos se comuniquem uns com os outros
- ▷ **Gerenciador de sistemas de arquivos** - organiza coleções nomeadas de dados em dispositivos de armazenamento e fornece uma interface para acessar os dados nesses dispositivos

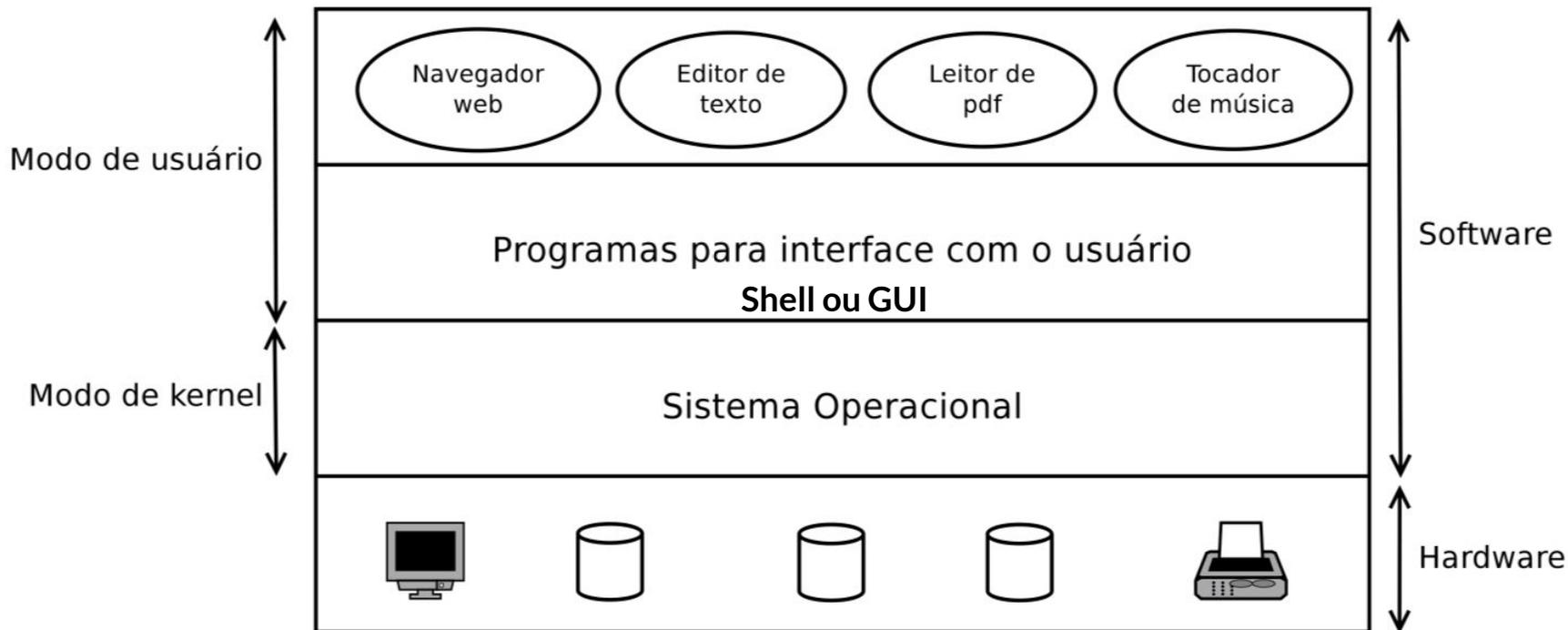
# Modo privilegiado ou supervisor (kernel) × modo usuário

- ▶ A maioria das CPUs possui dois modos de funcionamento, controlados por 1 bit de um dos registradores de uso específico:
- ▶ **Modo supervisor**
  - Nesse modo, a CPU pode executar qualquer instrução do seu conjunto de instruções e usar qualquer atributo do seu hardware.  
⇒ O sistema operacional é executado em modo supervisor, por isso tem acesso a todo o hardware.
- ▶ **Modo usuário**
  - Esse modo permite a execução de apenas um subconjunto das instruções e um subconjunto dos atributos de hardware. De forma geral, todas as instruções que envolvem entrada/saída e proteção de memória são inacessíveis nesse modo.  
⇒ Os programas aplicativos são executados em modo usuário.

# Arquiteturas de sistemas operacionais

- ▷ **Monolítica** - cada componente do SO é contido no núcleo (kernel) e pode comunicar-se diretamente com qualquer outro (por meio de chamadas à função) [arquitetura mais antiga e mais comum]
- ▷ **Em camadas** - agrupa em camadas os componentes que realizam tarefas similares
- ▷ **Microkernel** - fornece apenas um número pequeno de serviços (para manter o núcleo pequeno e escalável)
- ▷ **Cliente-servidor** - implementa muitas das funcionalidades do SO como processos de usuários (divididos entre processos clientes e processos servidores). O núcleo se encarrega da comunicação entre clientes e servidores

# Como tudo isso se conecta?



# 2.

## Interpretores de comandos (shells)

# Interpretador de comandos

É um programa que implementa uma interface de linha de comando. Essa, por sua vez, permite que usuários submetam comandos a um programa de computador por meio de linhas sucessivas de texto (ou seja, as linhas de comando)

**Shell = interpretador de comandos para um sistema operacional**

- É a interface primária existente entre um usuário e um sistema operacional quando o usuário não está usando uma interface gráfica
- por meio de um Shell, um usuário pode fazer um uso “intenso” das funcionalidades providas por um SO

# Um “parênteses” sobre o termo shell

- ▶ **Em Computação ...**

... o termo shell geralmente é usado para designar qualquer programa que atue como uma casca (= camada externa) entre usuários e: o kernel de um SO, ou outros programas, ou até mesmo linguagens. Sob essa perspectiva, mesmo um programa gráfico pode ser chamado de shell. Exemplo de shell gráfico: o Explorer.exe (versões do Windows).

- ▶ **Em SOs “originados” do Unix ...**

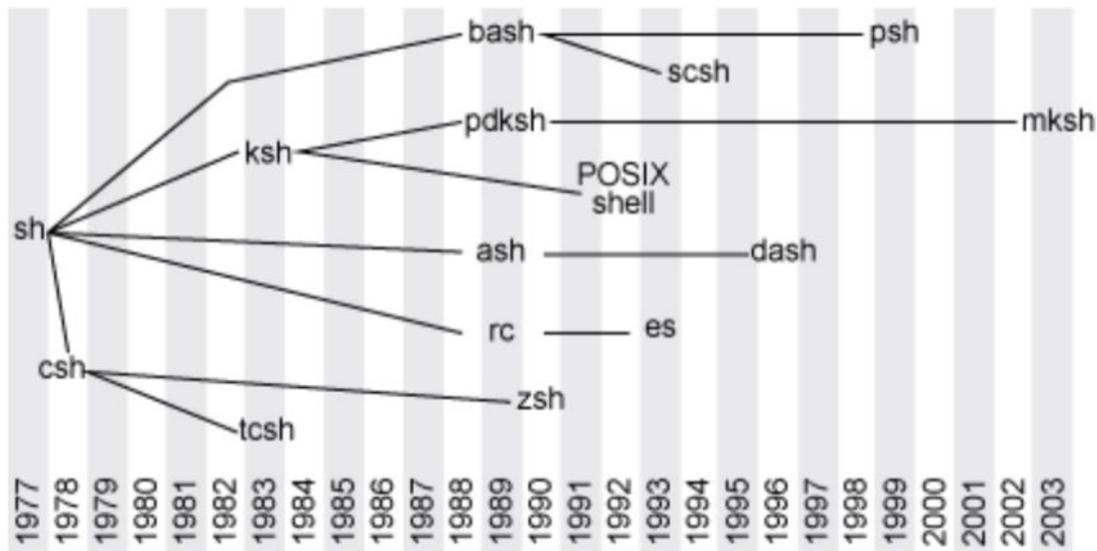
... shell assume um significado mais específico – o de interpretador de comandos em modo texto para o SO. Esse é o significado referenciado neste curso.

# Shells

Existem diversas implementações de Shells:

- ▶ **no Windows:** command.com e cmd.exe (Command), PowerShell, . . .
- ▶ **no Unix (e seus derivados):** sh (Bourne shell), csh (C shell), bash (Bourne-Again shell), ksh (Korn shell), tcsh (TENEX C shell) zsh (Z shell) ...

# Arvore genealógica dos Shells do GNU/Linux



Fonte:

<https://developer.ibm.com/tutorials/l-linux-shells/>

# Primeiro contato no GNU/Linux

Via emulador de terminal no modo gráfico ou via modo texto (ctrl+alt+F1, ..., F6)

```
Debian GNU/Linux 10 nebula tty1 Identificação do computador e do terminal
nebula login: lidenbrook Prompt de login
Password:
Last login: Thu Sep 10 17:16:20 -03 2020 from 192.168.0.10 on pts/1
Linux nebula 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1+deb10u1 (2020-04-27) x86_64

The programs included with the Debian GNU/Linux system are free software; Mensagem de login
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
lidenbrook@nebula:~$ Prompt de comandos date
dom set 13 13:30:08 -03 2020
lidenbrook@nebula:~$ uname -a
Linux nebula 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1+deb10u1 (2020-04-27) x86_64 GNU/Linux
lidenbrook@nebula:~$ _
```

# Material recomendado

- ▷ Capítulo 1 do livro Operating Systems: Design and Implementation, de Tanenbaum e Woodhull [disponível na biblioteca do IME]
- ▷ Artigo “Evolution of shells in Linux”, de M. Tim Jones <https://developer.ibm.com/tutorials/l-linux-shells/>