

Marisangila Alves, MSc

`marisangila.alves@edu.sc.senai.br`

`marisangila.com.br`

2026/1

Fundamentos de Desenvolvimento de Sistemas

Noções de Sistemas Operacionais

Sumário

- 1 Introdução
- 2 Sistemas Operacionais Modernos
- 3 Windows
- 4 Linux
- 5 FreeBSD

- 6 MacOS
- 7 GUI vs CLI
- 8 Execução de Processos
- 9 Tarefas
- 10 Bibliografia

Introdução

O que é um sistema operacional?

Sua função é fornecer aos programas do usuário um modelo do computador melhor, mais simples e mais limpo, assim como lidar com o gerenciamento de todos os recursos.

(TANENBAUM, 2010)

O sistema operacional é uma camada de software entre o hardware e os programas que executam tarefas para os usuários.

(OLIVEIRA; CARISSIMI; TOSCANI, 2001)

- › Em resumo, sistemas operacionais oferecem:
 - › **Conveniência**;
 - › **Eficiência**.
- › Tem aspectos de baixo nível (*drivers*);
- › Tem aspectos de alto nível (*interface* e utilitários).



Figura 1: Estrutura de um computador.

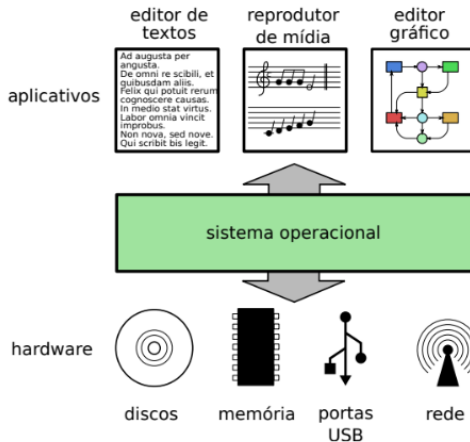


Figura 2: Estrutura de um computador (Maziero, 2019).

› Abstração de recursos

- » Prover interfaces de acesso aos dispositivos.
- » Tornar os aplicativos independentes do *hardware*.
- » Definir interfaces de acesso *homogêneas* para dispositivos com tecnologias distintas.

Exemplo: Abrir um arquivo.

- 1 Verificar se os parâmetros informados estão corretos (nome do arquivo, identificador do disco, buffer de leitura, etc.);
- 2 Verificar se o disco está disponível;
- 3 Ligar o motor do disco e aguardar atingir a velocidade de rotação correta;
- 4 Posicionar a cabeça de leitura sobre a trilha onde está a tabela de diretório;
- 5 Ler a tabela de diretório e localizar o arquivo ou subdiretório desejado;
- 6 Mover a cabeça de leitura para a posição do bloco inicial do arquivo;
- 7 Ler o bloco inicial do arquivo e depositá-lo em um buffer de memória.

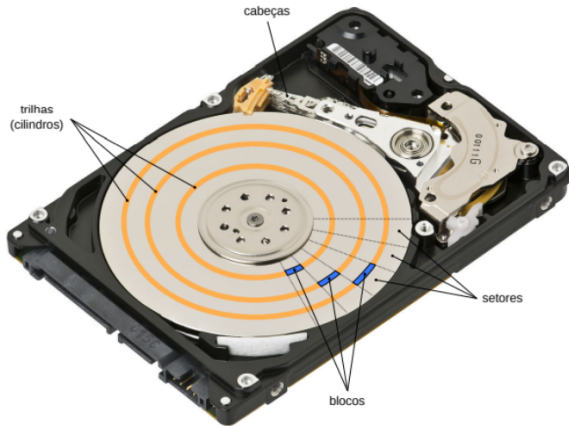


Figura 3: HD (Maziero, 2019).

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void){
4     FILE *arquivo = fopen("nome_arquivo.txt", "w");
5     if(arquivo == NULL)
6         printf("Não foi possível abrir o arquivo!\n");
7     fclose(arquivo);
8     return 0;
9 }
```

Código 1: Exemplo fopen() em Linguagem C.

Sistemas Operacionais Modernos



Windows

- A IBM era a maior empresa de computadores do mundo. Em 1981 lançou o IBM PC (Personal Computer) com processador Intel 8088;
- A Microsoft, fundada por Bill Gates e Paul Allen, inicialmente não possuía um sistema operacional para PCs;
- Licenciou e posteriormente adquiriu um sistema chamado 86-DOS de outra empresa, adaptando-o para o IBM PC. Assim surgiu o MS-DOS (Microsoft Disk Operating System);
- A Microsoft lançou o Microsoft Windows em 1985;

- › O Windows é um **software proprietário**, ou seja, seu código-fonte não é público e sua distribuição é controlada pela Microsoft;
- › A versão atual pertence à família **Windows 11**, utilizada em desktops, notebooks e ambientes corporativos;
- › Há uma versão para servidores (Windows Server);
- › Oferece menor liberdade e controle ao usuário;
- › É alvo frequente de atacantes e *malwares* devido à sua grande participação no mercado global.



Bill Gates



Windows IV

```
C:\DOS>chkdsk
Volume Serial Number is 3F76-4B5B

2,146,467,848 bytes total disk space
131,972 bytes in 2 hidden files
32,768 bytes in 1 directories
7,485,568 bytes in 124 user files
2,138,898,432 bytes available on disk

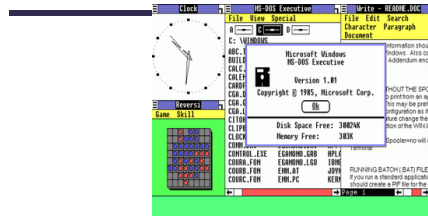
32,768 bytes in each allocation unit
65,585 total allocation units on disk
65,274 available allocation units on disk

655,360 total bytes memory
682,784 bytes free

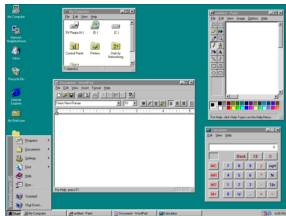
Instead of using CHKDSK, try using SCANDISK. SCANDISK can reliably detect
and fix a much wider range of disk problems. For more information,
type HELP SCANDISK from the command prompt.

C:\DOS>_
```

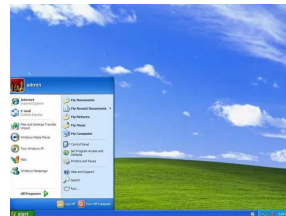
MS-DOS



Windows 1.0



Windows 95



Windows XP



Figura 4: Versão Atual, Windows 11.

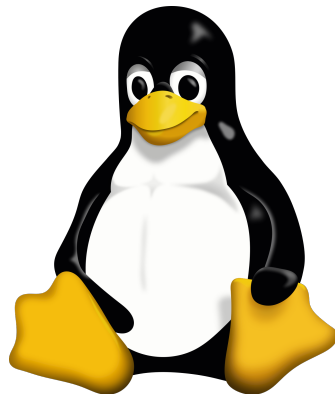
Linux

- › O **Linux** é um sistema operacional inspirado no sistema **Unix**.
- › O kernel Linux foi criado em 1991 por **Linus Torvalds**.
- › O kernel Linux é **open source**, permitindo que qualquer pessoa estude, modifique e distribua o código.
- › Possui uma grande **comunidade de desenvolvedores** e milhares de contribuições ao redor do mundo.

- › Existem muitas **distribuições Linux**.
- › O sistema **Android** utiliza o kernel Linux.
- › Muitas distribuições oferecem grande **liberdade e controle ao usuário**
- › Algumas distribuições podem possuir maior **curva de aprendizado**.



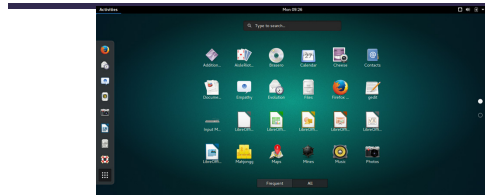
Linux Torvalds



Tux, Mascote do Linux.



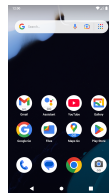
Mint



Ubuntu



Debian



Android

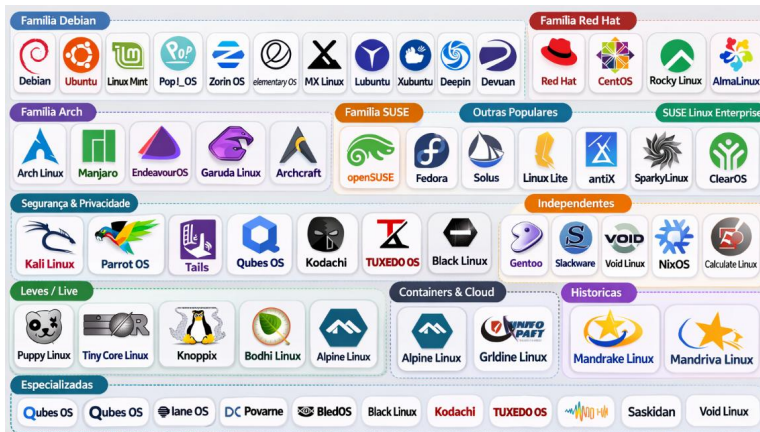
 CLICA!

-  Acesse: Distribuições Linux

 CLICA!

-  Acesse: Código Fonte Kernel Linux





FreeBSD

- › O **FreeBSD** é um sistema operacional derivado do tradicional **Unix**.
- › Ele utiliza a **licença BSD**, considerada uma licença permissiva de software livre.
- › O sistema é conhecido por sua **estabilidade, segurança e alto desempenho**.
- › Muitos sistemas modernos utilizam tecnologias derivadas do BSD.



Figura 5: BSD Daemon



Figura 6: Exemplo de sistema baseado em BSD.

MacOS

- › Desenvolvido pela empresa **Apple** para os computadores da linha *Mac*;
- › O sistema **macOS** da Apple possui base em tecnologias derivadas do BSD e do Unix;
- › A partir do macOS surgiram outros sistemas como:
 - › **iOS** (iPhone);
 - › **iPadOS**;
- › Otimizado para hardware específico;
- › Menor controle e liberdade para o usuário;
- › Melhor experiência de usuário;
- › Maior segurança para o usuário.



Steve Jobs





Figura 7: Exemplo de MacOS.



Figura 8: Exemplo de iOS.

Comparação entre sistemas operacionais I

Característica	Windows	Linux	BSD
Origem	MS-DOS / Microsoft	Inspirado no Unix	Derivado do Unix
Desenvolvedor	Microsoft	Comunidade / empresas	Comunidade BSD
Licença	Proprietária	Open Source (GPL)	BSD (permissiva)
Código-fonte	Fechado	Aberto	Aberto
Uso comum	Desktop e empresas	Servidores e sistemas embarcados	Servidores e redes
Distribuições	Versões do Windows	Ubuntu, Debian, Fedora etc.	FreeBSD, OpenBSD, NetBSD
Personalização	Limitada	Muito alta	Alta
Exemplos derivados	Windows Server	Android	macOS, iOS

Tabela 1: Comparação entre famílias de sistemas operacionais

Nota:

QUAL É O MELHOR SISTEMA OPERACIONAL?

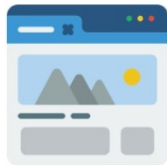
DEPENDENTE!

Nota:

A forma como um sistema operacional coleta dados de uso (telemetria) e trata a privacidade do usuário deve ser considerada no momento da escolha do sistema.

GUI vs CLI

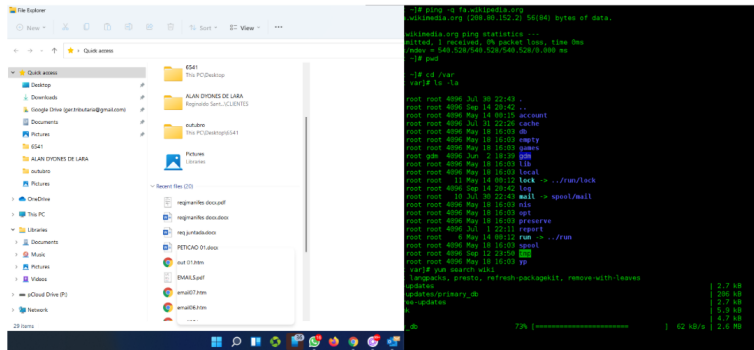
- **GUI (Graphical User Interface):** Interface que permite ao usuário interagir com o sistema por meio de elementos gráficos como janelas, ícones e botões.
- **CLI (Command-line Interface):** Interface que permite ao usuário interagir com o sistema digitando comandos em texto por meio de um console ou terminal.



VS.



Interfaces II





O que é um processo?

Processo é uma abstração de um programa em execução.

(TANENBAUM, 2010)

(TANENBAUM, 2010) Um processo é apenas uma instância de um programa em execução, incluindo os valores atuais do contador do programa, registradores e variáveis.

Atenção!

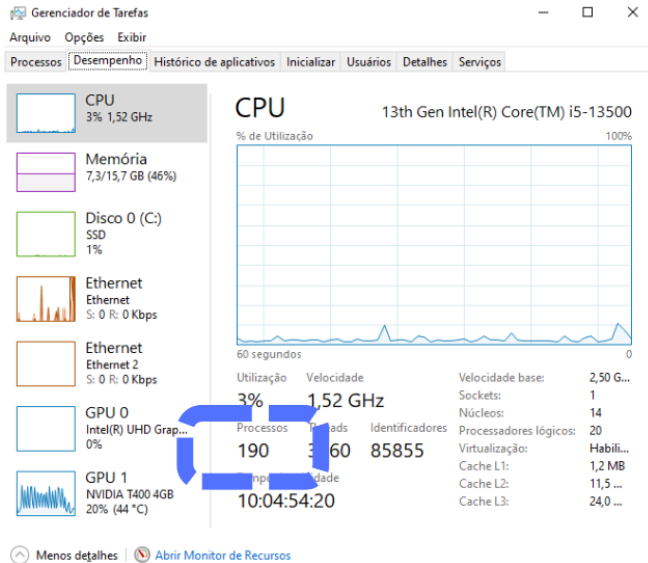
Processo $\neq$ Programa

- › **Programa**: Entidade estática;
- › **Processo**: Seu estado muda a medida em que avança sua execução.

Nota: (Analogia:)

Podemos pensar também em receitas e cozinheiros (Maziero, 2019).

- 1 Pode-se dizer que um programa é o equivalente de uma “receita de torta” dentro de um livro de receitas (um diretório) guardado em uma estante (um disco) na cozinha (o computador).
- 2 Essa receita de torta define os ingredientes necessários (entradas) e o modo de preparo (programa) da torta (saída).
- 3 A ação de “executar” a receita, providenciando os ingredientes e seguindo os passos definidos na mesma, é a tarefa propriamente dita.
- 4 A cada momento, o cozinheiro (o processador) está seguindo um passo da receita (posição da execução) e tem uma certa disposição dos ingredientes e utensílios em uso (as entradas e variáveis internas da tarefa).



```
File Edit View Terminal Tabs Help

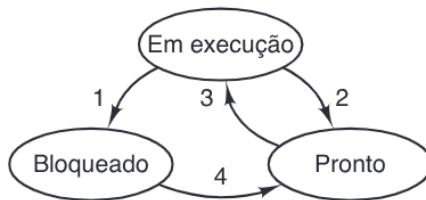
1[||||| 31.4%] 2[||||| 40.3%] 3[||||| 29.4%] 4[|||||||72.2%]
Mem[||||||| 6.25G/15.4G Tasks: 127, 736 thr, 123 kthr; 2 runni
Swp[||||| 0K/2.00G Load average: 1.54 1.84 1.66
Uptime: 01:43:57

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1 root 20 0 23088 13844 9236 S 0.0 0.1 0:02.23 /sbin/init sp
308 root 19 -1 66996 19984 18960 S 0.0 0.1 0:01.16 /usr/lib/syst
364 root 20 0 29960 8064 4864 S 0.0 0.0 0:00.49 /usr/lib/syst
687 systemd-re 20 0 21668 12416 10240 S 0.0 0.1 0:00.86 /usr/lib/syst
795 root 20 0 305M 7636 6868 S 0.0 0.0 0:00.08 /usr/libexec/
796 root 20 0 2720 1664 1536 S 0.0 0.0 0:00.31 /usr/sbin/acp
799 avahi 20 0 8748 4352 3968 S 0.0 0.0 0:02.78 avahi-daemon:
801 root 20 0 13508 6656 6144 S 0.0 0.0 0:00.04 /usr/libexec/
802 root 20 0 9428 2688 2560 S 0.0 0.0 0:00.01 /usr/sbin/cro
803 messagebus 20 0 11092 6272 4352 S 0.0 0.0 0:02.31 @dbus-daemon
811 root 20 0 82920 3968 3712 S 0.0 0.0 0:00.93 /usr/sbin/irq
822 root 20 0 305M 7636 6868 S 0.0 0.0 0:00.00 /usr/libexec/
823 root 20 0 305M 7636 6868 S 0.0 0.0 0:00.23 /usr/libexec/
826 root 20 0 40668 22784 11648 S 0.0 0.1 0:00.22 /usr/bin/pyth
832 root 20 0 82920 3968 3712 S 0.0 0.0 0:00.00 /usr/sbin/irq
834 polkitd 20 0 309M 11652 8020 S 0.0 0.1 0:00.26 /usr/lib/polkit
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice + F9Kill F10Quit
```

1

```
htop
```

Execução de Processos



1. O processo é bloqueado aguardando uma entrada
2. O escalonador seleciona outro processo
3. O escalonador seleciona esse processo
4. A entrada torna-se disponível

Figura 9: Diagrama de estados de um processo (TANENBAUM, 2010).

- › **Nova:** a tarefa está sendo preparada para executar.
- › **Pronta:** a tarefa está esperando pelo processador.
- › **Executando:** a tarefa está executando suas instruções.
- › **Suspensa:** a tarefa aguarda algum evento externo.
- › **Terminada:** a tarefa encerrou ou foi abortada.

Tarefas

O que o sistema operacional faz? I

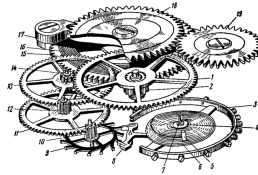
- › Quando um processo executa instruções na CPU.
- › Quanta memória RAM e onde o processo executa.
- › Gerenciamento de arquivos.
- › Gerenciamento de usuário, permissões.
- › Gerenciamento de entrada e saída.

Capítulo 1 (Maziero, 2019)

SISTEMAS OPERACIONAIS: CONCEITOS E MECANISMOS

PROF. CARLOS A. MAZIERO

DINF - UFPR



Bibliografia

MAZIERO, Carlos Alberto. **Sistemas Operacionais: Conceitos e Mecanismos [recurso eletrônico]**. Curitiba: DINF - UFPR, 2019. ISBN 978-85-7335-340-2.

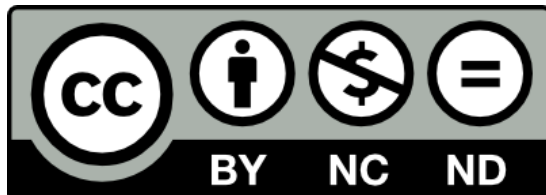
OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. **Sistemas Operacionais**. 2. ed. Porto Alegre: Sagra-Luzzatto, 2001.

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.



MAZIERO, Carlos Alberto. **Sistemas Operacionais: Conceitos e Mecanismos [recurso eletrônico]**. Curitiba: DINF - UFPR, 2019. ISBN 978-85-7335-340-2.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Marisangila Alves, MSc

`marisangila.alves@edu.sc.senai.br`

`marisangila.com.br`

2026/1

Fundamentos de Desenvolvimento de Sistemas

Noções de Sistemas Operacionais