

# MÓDULO SISTEMAS OPERACIONAIS

Curso Técnico em Redes / Infraestrutura

Carga Horária: 80H

Ambiente das práticas: VPS Linux Debian + ttyd via Chromebook

Versão: 2026

Professor: Marcos Brandão

---

## SUMÁRIO

### I UNIDADE — FUNDAMENTOS DOS SISTEMAS OPERACIONAIS

1. Introdução aos Sistemas Operacionais
  - 1.1 Conceitos fundamentais
  - 1.2 Hardware e Software
  - 1.3 Funções do Sistema Operacional
  - 1.4 Tipos de Sistemas Operacionais
2. Estrutura dos Sistemas Operacionais
  - 2.1 Kernel
  - 2.2 Shell
  - 2.3 Interface gráfica e terminal
  - 2.4 Serviços do sistema
3. Gerenciamento de Processador
  - 3.1 Processos
  - 3.2 Escalonamento
  - 3.3 Threads
  - 3.4 Monitoramento de CPU
4. Gerenciamento de Memória
  - 4.1 Memória RAM
  - 4.2 Memória virtual
  - 4.3 Swap
  - 4.4 Monitoramento de memória
5. Sistemas de Arquivos
  - 5.1 Conceitos
  - 5.2 Estrutura de diretórios
  - 5.3 Partições
  - 5.4 Permissões

---

## **II UNIDADE — SISTEMAS OPERACIONAIS MODERNOS**

- 6. Sistema Operacional Linux
    - 6.1 Distribuições Linux
    - 6.2 Terminal Linux
    - 6.3 Administração básica
    - 6.4 Serviços Linux
  - 7. Sistema Operacional Windows
    - 7.1 Estrutura do Windows
    - 7.2 Gerenciamento básico
    - 7.3 Serviços do Windows
    - 7.4 Ferramentas administrativas
  - 8. Administração de Usuários
    - 8.1 Usuários e grupos
    - 8.2 Controle de acesso
    - 8.3 Segurança básica
  - 9. Processos e Serviços
    - 9.1 Serviços Linux
    - 9.2 Serviços Windows
    - 9.3 Gerenciamento de tarefas
  - 10. Armazenamento e Backup
    - 10.1 Disco e partições
    - 10.2 Backup
    - 10.3 Compactação
  - 11. Projeto Integrador
-

# I UNIDADE

## CAPÍTULO 1 — INTRODUÇÃO AOS SISTEMAS OPERACIONAIS

### 1.1 O que é um Sistema Operacional?

Sistema Operacional (SO) é o software responsável por controlar o funcionamento do computador.

Ele faz a comunicação entre:

- hardware
- usuário
- aplicações

---

## Exemplos de Sistemas Operacionais

Sistema	Uso
Windows	desktops
Linux	servidores
Android	smartphones
iOS	celulares Apple

---

# Exemplo do funcionamento

Quando o usuário abre um navegador:

Usuário → Sistema Operacional → Hardware

O sistema operacional organiza:

- memória
  - processador
  - arquivos
  - dispositivos
- 

## 1.2 Hardware e Software

### Hardware

Parte física do computador.

---

### Exemplos

Hardware	Função
CPU	processamento
RAM	memória
SSD/HD	armazenamento
Placa de Rede	comunicação

---

# Software

Parte lógica.

---

## Exemplos

Software	Tipo
Linux	sistema operacional
Chrome	aplicativo
VSCode	editor

---

## 1.3 Funções do Sistema Operacional

O SO é responsável por:

- gerenciamento da CPU
  - gerenciamento de memória
  - gerenciamento de arquivos
  - gerenciamento de dispositivos
  - segurança
  - execução de programas
- 

## PRÁTICA — Identificando o Sistema Operacional

### Comando

`uname -a`

---

# Resultado esperado

Linux servidor 6.1.0-amd64 x86\_64 GNU/Linux

---

## Explicação

Parte	Significado
Linux	sistema operacional
x86_64	arquitetura
GNU/Linux	plataforma

---

# PRÁTICA — Descobrimo versão do Linux

## Comando

```
cat /etc/os-release
```

---

## Resultado esperado

```
PRETTY_NAME="Debian GNU/Linux 12"
```

---

## Explicação

Mostra a distribuição Linux instalada.

---

# CAPÍTULO 2 — ESTRUTURA DOS SISTEMAS OPERACIONAIS

## 2.1 Kernel

Kernel é o núcleo do sistema operacional.

Ele controla:

- memória
  - CPU
  - dispositivos
  - processos
- 

## Fluxo

Aplicativos

↓

Kernel

↓

Hardware

---

## 2.2 Shell

Shell interpreta comandos digitados pelo usuário.

---

## Exemplo

ls

---

## Explicação

O shell recebe o comando e solicita ao sistema operacional a execução.

---

## 2.3 Interface gráfica e terminal

Interface	Característica
GUI	gráfica
CLI	linha de comando

---

## Vantagens do terminal

- rapidez
  - automação
  - administração remota
- 

## PRÁTICA — Navegando no sistema

### Diretório atual

`pwd`

---

### Listando arquivos

`ls -l`

---

### Mudando diretório

`cd /etc`

---

## Explicação

<b>Comando</b>	<b>Função</b>
pwd	local atual
ls	listar
cd	navegar

---

# CAPÍTULO 3 — GERENCIAMENTO DE PROCESSADOR

## 3.1 Processos

Processo é um programa em execução.

---

## Exemplos

<b>Process</b>	<b>Função</b>
o	
nginx	servidor web
ssh	acesso remoto
chrome	navegador

---

## Visualizando processos

### Comando

ps aux

---

# Resultado esperado

Lista de processos ativos.

---

## Explicação

**Coluna**   **Significado**

USER   usuário

PID   identificador

%CPU   uso da CPU

---

## 3.2 Escalonamento

O sistema operacional decide qual processo usará a CPU.

---

## Objetivos

- desempenho
  - equilíbrio
  - multitarefa
- 

## 3.3 Threads

Threads são subdivisões de processos.

Permitem executar tarefas simultaneamente.

---

# PRÁTICA — Monitorando CPU

## Comando

top

---

## Informações exibidas

- uso CPU
  - memória
  - processos
  - carga do sistema
- 

## Instalando htop

apt install htop -y

---

## Executando

htop

---

## Explicação

Interface visual para monitoramento.

---

# CAPÍTULO 4 — GERENCIAMENTO DE MEMÓRIA

## 4.1 Memória RAM

Armazena dados temporariamente durante execução.

---

### Características

- rápida
  - volátil
  - temporária
- 

## 4.2 Memória Virtual

Usa espaço do disco como complemento da RAM.

---

### Swap

Área reservada para memória virtual.

---

## Verificando memória

### Comando

```
free -h
```

---

# Resultado esperado

Mem: 2Gi used 1Gi free 1Gi

---

## Explicação

Campo	Função
used	utilizada
free	livre
swap	memória virtual

---

## Verificando swap

swapon --show

---

# CAPÍTULO 5 — SISTEMAS DE ARQUIVOS

## 5.1 Conceito

Sistema de arquivos organiza dados no disco.

---

## Exemplos

Sistema	Uso
ext4	Linux
NTFS	Windows
FAT32	pendrive

---

## 5.2 Estrutura Linux

Diretório	Função
/home	usuários
/etc	configurações
/var	logs
/tmp	temporários

---

## 5.3 Partições

Discos podem ser divididos em partições.

---

## Verificando discos

### Comando

lsblk

---

## Resultado esperado

```
sda  
├─sda1  
└─sda2
```

---

## Verificando espaço

df -h

---

# Explicação

Mostra:

- espaço usado
  - espaço livre
  - partições
- 

## 5.4 Permissões

Linux usa permissões para controlar acesso.

---

## Exemplo

`rwxr-xr--`

---

## Significado

Letra	Permissã
	o
r	leitura
w	escrita
x	execução

---

## Alterando permissões

### Comando

`chmod 755 script.sh`

---

## II UNIDADE

# CAPÍTULO 6 — SISTEMA OPERACIONAL LINUX

## 6.1 Distribuições Linux

Distribuições são versões do Linux.

---

### Exemplos

Distribuição	Uso
Debian	servidores
Ubuntu	desktop
Kali	segurança

---

## 6.2 Terminal Linux

O terminal é essencial para administração.

---

## Atualizando sistema

### Comando

`apt update`

---

# Atualizando pacotes

```
apt upgrade -y
```

---

## 6.3 Serviços Linux

Serviços executam tarefas em segundo plano.

---

### Exemplos

- ssh
  - nginx
  - docker
- 

### Verificando serviço

#### Comando

```
systemctl status ssh
```

---

### Resultado esperado

```
active (running)
```

---

### Reiniciando serviço

```
systemctl restart ssh
```

---

# CAPÍTULO 7 — SISTEMA OPERACIONAL WINDOWS

## 7.1 Estrutura do Windows

Windows possui:

- interface gráfica
  - gerenciador de tarefas
  - serviços
  - registro
- 

## Ferramentas administrativas

Ferramenta	Função
Gerenciador de Tarefas	processos
services.msc	serviços
diskmgmt.msc	discos

---

## 7.2 Gerenciador de Tarefas

Mostra:

- CPU
  - RAM
  - processos
  - rede
- 

## 7.3 CMD e PowerShell

Interfaces de linha de comando do Windows.

---

## Exemplo CMD

ipconfig

---

## Exemplo PowerShell

Get-Process

---

# CAPÍTULO 8 — USUÁRIOS E SEGURANÇA

## Usuários Linux

Cada usuário possui:

- login
  - senha
  - permissões
- 

## Criando usuário

### Comando

adduser aluno1

---

## Grupos

Facilitam controle de permissões.

---

## Criando grupo

```
groupadd suporte
```

---

## Adicionando usuário ao grupo

```
usermod -aG suporte aluno1
```

---

## Verificando grupos

```
groups aluno1
```

---

# CAPÍTULO 9 — PROCESSOS E SERVIÇOS

## Matando processo

### Comando

```
kill PID
```

---

## Exemplo

```
kill 1024
```

---

## Serviços automáticos

### Habilitando serviço

```
systemctl enable nginx
```

---

# Verificando logs

```
journalctl -xe
```

---

## CAPÍTULO 10 — ARMAZENAMENTO E BACKUP

### Compactação

#### Criando backup

```
tar -czvf backup.tar.gz /home
```

---

#### Restaurando

```
tar -xzvf backup.tar.gz
```

---

### Agendamento de backup

#### Cron

```
crontab -e
```

---

#### Exemplo

```
0 2 * * * tar -czf /backup/home.tar.gz /home
```

---

#### Explicação

Executa backup diariamente às 2h.

---

# CAPÍTULO 11 — PROJETO FINAL

## Objetivo

Montar ambiente Linux contendo:

- usuários
  - permissões
  - monitoramento
  - backup
  - serviços
- 

## Etapas

### 1. Atualizar sistema

```
apt update && apt upgrade -y
```

---

### 2. Criar usuários

```
adduser aluno1
```

---

### 3. Instalar ferramentas

```
apt install htop nginx -y
```

---

### 4. Verificar serviços

```
systemctl status nginx
```

---

### 5. Criar backup

```
tar -czvf sistema.tar.gz /etc
```

---

# Resultado esperado

Os alunos terão experiência prática com:

- administração Linux
  - gerenciamento de processos
  - memória
  - permissões
  - backup
  - monitoramento
  - serviços Linux
- 

## CONCLUSÃO

Ao final da disciplina o aluno será capaz de:

- compreender a estrutura dos sistemas operacionais
- administrar Linux e Windows
- gerenciar processos e memória
- controlar usuários e permissões
- administrar serviços
- trabalhar com sistemas de arquivos
- realizar backup e monitoramento
- utilizar terminal Linux profissionalmente