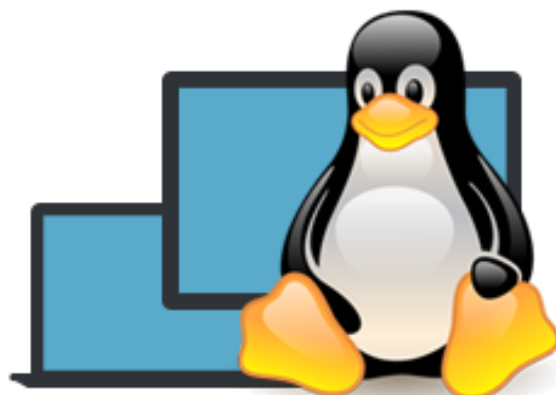


Curso DevOps - Automação e Integração Contínua

Sistemas Operativos
Linux



1. INTRODUÇÃO

O que é um Sistema Operativo?

Um sistema operativo (SO) é o software essencial que gere os recursos de hardware de um computador e fornece uma interface para que os utilizadores e outros programas possam interagir com o dispositivo. Ele atua como um intermediário entre o hardware (como o processador, a memória e os discos) e as aplicações, garantindo que tudo funcione de forma coordenada.

Funções principais de um sistema operativo:

- ❖ **Gestão de Processos:** Controla a execução de programas, decidindo quais processos são executados e quando são executados.
- ❖ **Gestão de memória:** Aloca e liberta memória para aplicações e processos.
- ❖ **Gestão de ficheiros:** Organiza e controla o acesso a ficheiros e diretórios no disco.
- ❖ **Gestão de dispositivos:** Comunica com hardware, como impressoras, discos e redes.
- ❖ **Interface com o utilizador:** Oferece uma forma de interação, seja por interface gráfica (GUI) ou pelo terminal/linha de comandos (CLI).

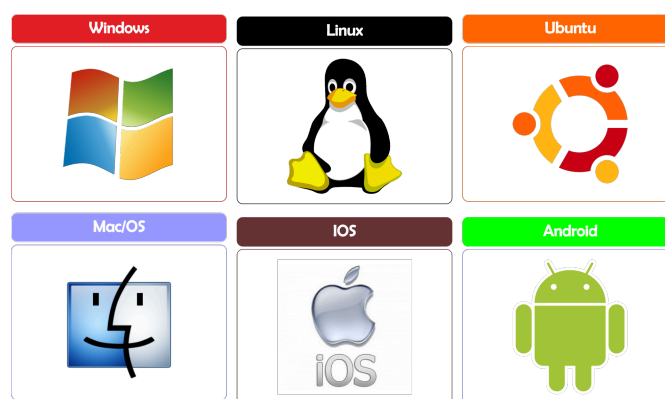
Arquiteturas de sistemas operativos

A arquitetura de um sistema operativo pode variar bastante, refletindo diferentes objetivos de desempenho, segurança, modularidade e portabilidade. Algumas abordagens notáveis incluem:

- **Monolítica:** O kernel (núcleo do sistema) implementa todas as funções básicas, como gestão de memória, de processos e de dispositivos, em um único bloco. Exemplos incluem sistemas tipo UNIX clássicos e o Linux, ainda que o Linux seja, na prática, modular, mas mantém um kernel monolítico.
- **Microkernel:** A abordagem microkernel tenta manter no núcleo apenas as funções essenciais (gestão de processos e IPC - comunicação entre processos) e delega serviços adicionais a processos em espaço de utilizador. Essa abordagem pode trazer maior confiabilidade e facilidade de manutenção, mas potencialmente a um custo de desempenho. O Minix é um exemplo clássico.

- **Híbrida:** Muitos sistemas operativos comerciais modernos, como o Windows e o macOS, adotam uma abordagem híbrida, mesclando características de kernel monolítico e microkernel, tentando equilibrar desempenho com modularidade e segurança.
- **Exokernel e outros:** Modelos mais experimentais, como o exokernel, oferecem apenas serviços mínimos para a gestão de recursos, deixando a maior parte do trabalho para bibliotecas e servidores rodando em nível de utilizador. Essa arquitetura pode atingir altíssimo desempenho, embora seja mais complexa de programar.

Exemplos de sistemas operativos



Windows: Um dos sistemas operativos mais populares em computadores pessoais e ambientes corporativos. Oferece interface gráfica intuitiva e ampla gama de software e jogos.

macOS: Um sistema operativo desenvolvido pela Apple, é o sistema operativo utilizado em dispositivos Apple, focado em design e integração.

Linux: Kernel de código aberto, suporta uma variedade de distribuições (Ubuntu, Fedora, Debian, RedHat, etc.). Amplamente usado em servidores, dispositivos embarcados e supercomputadores, o Linux é reconhecido pela sua estabilidade, segurança e flexibilidade.

Android e iOS: Dominam o mercado de dispositivos móveis. O Android, baseado no Kernel Linux, é mantido pelo Google e utilizado em uma ampla gama de smartphones e

tablets de diversos fabricantes. O iOS, da Apple, é exclusivo para iPhones e iPads, oferecendo um ecossistema fechado e alto nível de controle de qualidade dos aplicativos.

2. LINUX

O Linux é um sistema operativo de código aberto, o que significa que o seu código-fonte está disponível para consulta, modificação e distribuição. Criado por Linus Torvalds em 1991, o Linux é conhecido pela sua estabilidade, segurança e flexibilidade. Diferentemente de outros sistemas operativos, o Linux é desenvolvido por uma comunidade global de programadores e empresas, o que o torna altamente personalizável.

Características do Linux

- **Código aberto:** Permite adaptações para necessidades específicas.
- **Multitarefa e multiutilizador;** Suporta vários utilizadores e processos em simultâneo.
- **Modularidade:** Pode ser configurado para servidores, desktop ou dispositivos embarcados
- **Segurança:** Oferece controlos robustos de permissões e atualizações frequentes.
- **Ampla suporte:** Usado em servidores, supercomputadores, Clouds e ferramentas DevOps.

No contexto de DevOps, o Linux é essencial devido à sua compatibilidade com ferramentas de automação (como Ansible, Docker e Kubernetes), eficiência em servidores e suporte a scripts e pipelines.

Uma distribuição (distro) Linux é um sistema operativo que é construído em cima do Kernel do Linux. Ela inclui uma ampla gama de ofertas de software, como o conjunto de ferramentas GNU, bibliotecas de sistema, uma interface gráfica de utilizador (GUI) e aplicativos e utilitários adicionais para formar um sistema completo.

Existem várias distribuições Linux, entre as mais usadas podemos encontrar o Ubuntu, Linux Mint, Fedora, Debian, Arch Linux, openSUSE, CentOS, entre outros.

2.1. Sistemas de Arquivos

Partições

Podemos dividir um disco rígido em várias partes ou partições, onde cada partição é independente das outras, ou seja, cada partição pode ter o próprio sistema de arquivo e um diferente sistema operativo. Isto quer dizer que uma partição do disco não interfere nas outras partições. Podemos, por exemplo, instalar o Linux numa partição e o Windows noutra partição.

Atualmente existem dois padrões que determinam como os dados são armazenados do disco. Estes padrões são: MBR e GPT.

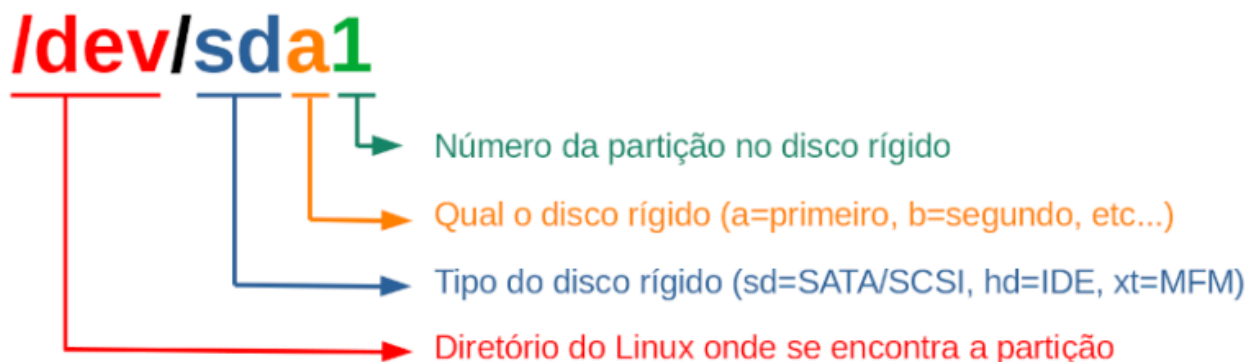
MBR (Master Boot Record)

É um padrão antigo que só permite 4 partições (chamadas primárias) no mesmo disco. Por isso, costuma-se usar a quarta partição como partição estendida para criar várias partições lógicas (em outras áreas do disco).

GPT (GUID Partition Table)

Pode criar até 128 ou mais partições (depende do sistema operativo). Neste caso, não há necessidade de criar partição estendida, embora seja possível.

O Linux utiliza a seguinte nomenclatura para identificar discos e partições.



Um programa muito utilizado no Linux para particionar discos é o fdisk (<https://pt.wikipedia.org/wiki/Fdisk>). O problema com este utilitário é que ele destrói os dados armazenados ao particionar o disco.

Sistema de Arquivos no Linux

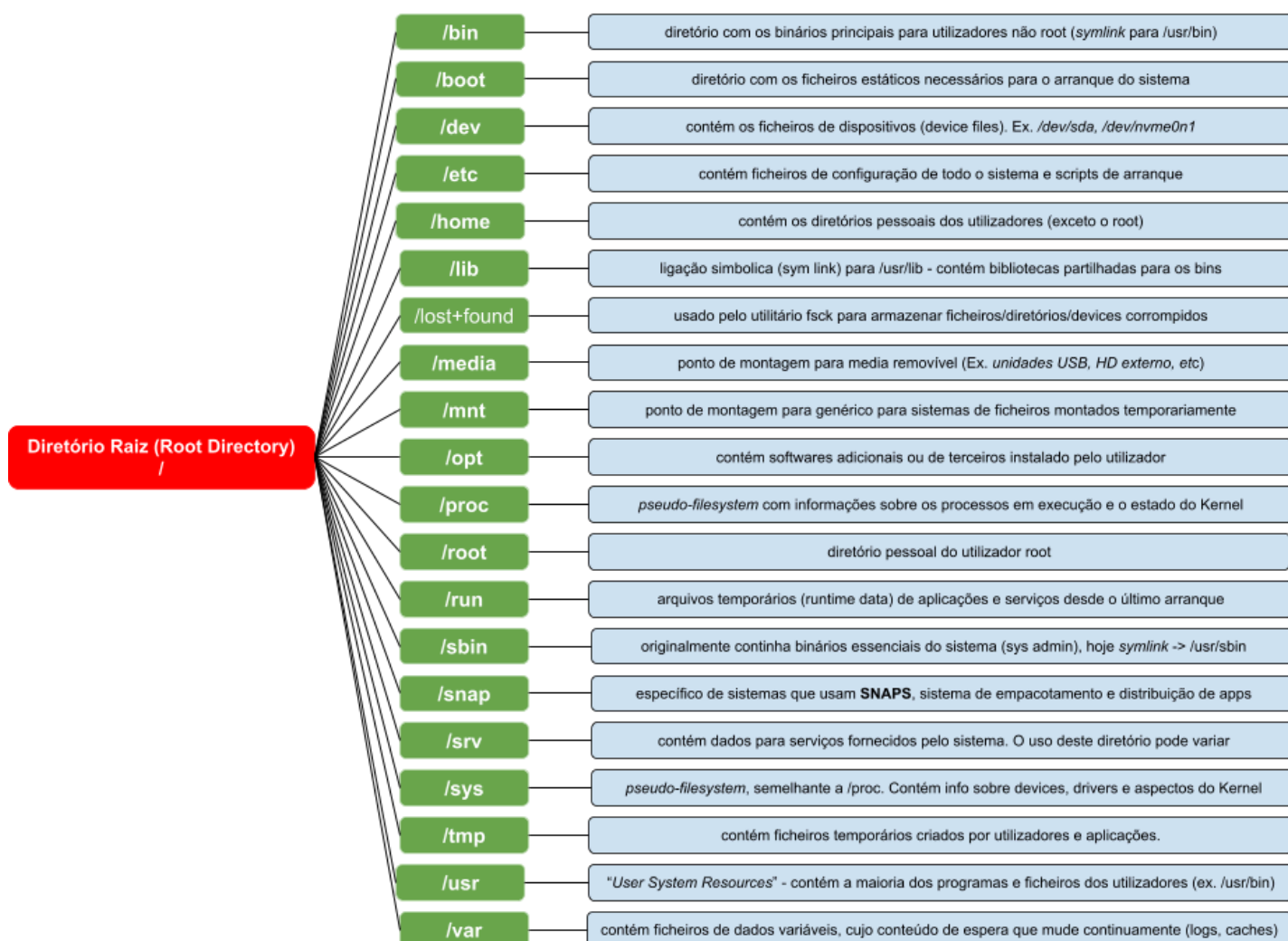
O sistema de arquivos é um conjunto de estruturas lógicas que permite o sistema operativo controlar o acesso a um dispositivo de armazenamento como disco rígido, pen-drive, CD-ROM, etc. Diferentes sistemas operativos podem usar diferentes sistemas de arquivos.

A partição Linux nativa é conhecida por directório raiz do Linux e é representada por uma barra (/).

Atualmente, o NTFS (New Technology File System) é o sistema de arquivos padrão do Windows, enquanto o ext4 é o do Linux. Embora o ext4 seja o padrão no Linux, outros sistemas de arquivos vêm ganhando popularidade no Linux, como o: Btrfs (B-tree File System), XFS, ZFS e o F2FS. Para verificar quais os sistemas de arquivos o seu Linux suporta, basta verificar o conteúdo do arquivo /proc/filesystems.

```
ubuntu@devops01:~$ cat /proc/filesystems
nodev    sysfs
nodev    tmpfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cgroup2
nodev    cpuset
nodev    devtmpfs
nodev    configfs
nodev    debugfs
nodev    tracefs
nodev    securityfs
nodev    sockfs
nodev    bpf
nodev    pipefs
nodev    ramfs
nodev    hugetlbfs
nodev    devpts
        ext3
        ext2
        ext4
        squashfs
```

No Linux, um diretório (corresponde ao conceito de pasta do Windows) pode ter outros diretórios ou arquivos. Dizemos que um diretório é filho de outro diretório quando ele está logo abaixo do diretório em questão. O diretório que está um nível acima é chamado de diretório pai.



O diretório raiz do Linux (o diretório */*) é o diretório com maior hierarquia entre todos os diretórios do sistema. Isto significa que todos os diretórios do Linux ficam abaixo deste diretório.

Para ver os diretórios e arquivos que ficam abaixo do diretório */*, basta usar o comando *ls: ls /*.

3. Ubuntu

Ubuntu é uma distribuição Linux livre e open-source baseada no Kernel Linux, desenvolvido pela Canonical, lançado em 2004. É uma das distribuições mais populares devido a sua facilidade de uso, vasta documentação e suporte de uma grande comunidade. O Ubuntu é ideal tanto para iniciantes como para profissionais, sendo amplamente adotado em ambientes de produção, como servidores web, Clouds e pipelines de DevOps.

3.1. Vantagens do Ubuntu

Ubuntu oferece várias vantagens que fizeram dele uma escolha preferida entre os utilizadores e profissionais de TI, dentre eles, os profissionais DevOps:

Segurança

O Ubuntu é conhecido pelas suas fortes funcionalidades de segurança, incluindo firewalls incorporadas, correções de segurança regulares e uma equipa de segurança dedicada que monitoriza e resolve constantemente potenciais vulnerabilidades. Adicionalmente, o Ubuntu suporta a encriptação total do disco, o que protege os seus dados mesmo em caso de roubo ou perda do dispositivo.

Open source

Como uma plataforma open source, o Ubuntu permite que os utilizadores acessem, modifiquem ou partilhem o seu código fonte. Isto encoraja a colaboração e a inovação entre os programadores de todo o mundo. A natureza open source do Ubuntu também garante a transparência, dando aos utilizadores controlo sobre os seus softwares e dados.

Desempenho

Ubuntu é otimizado para desempenho, garantindo uma experiência de utilizador rápida e reativa, mesmo em hardware mais antigo. Utiliza componentes leves que consomem menos recursos, o que significa que pode ser executado com eficácia em dispositivos com capacidade de processamento ou memória limitada.

Facilidade de utilização

Ao contrário de outras distribuições Linux, Ubuntu foi concebido a pensar na ergonomia. Inclui uma interface gráfica do utilizador (GUI) simples e intuitiva, bem como aplicações pré-instaladas como um browser da Web, pacote de escritório e ferramentas multimédia. Isto faz dele uma excelente escolha para os utilizadores novos no Linux.

3.2. Ubuntu para Servidores

Ubuntu é amplamente utilizado para ambientes de servidor devido à sua estabilidade, segurança e suporte para várias aplicações de servidor. É uma excelente escolha para alojar servidores web, servidores de aplicações e bases de dados. Ubuntu Server foi otimizado para lidar com cargas de trabalho pesadas e oferece funcionalidades como:

Suporte para Containers

Ubuntu suporta Docker e outras tecnologias de containers, permitindo que você crie, implemente e gerencie aplicativos em ambientes isolados.

Atualizações automáticas

Mantém o seu servidor seguro com atualizações regulares e correções.

Ubuntu nas plataformas Cloud

Ubuntu é o sistema operativo mais utilizado pelas plataformas Cloud. Foi otimizado a nível de escalabilidade, segurança e compatibilidade com os principais fornecedores de cloud, como a AWS, Microsoft Azure e Google Cloud. As dimensões reduzidas do Ubuntu e o Kernel otimizado para Cloud fazem dele uma excelente escolha para implementar aplicações, máquinas virtuais e containers nativos da cloud.

3.3. Como aceder ao Ubuntu

Existem várias formas de aceder ao Ubuntu, dependendo do seu ambiente e objetivos:

Máquina virtual

Utilize ferramentas como VirtualBox ou VMware para instalar o Ubuntu. Faça o download da imagem ISO em <https://ubuntu.com/download/server> e configure a máquina virtual.

WSL (Windows Subsystem for Linux)

O WSL (Windows Subsystem for Linux) é uma funcionalidade do Windows que permite executar um ambiente Linux diretamente no Windows, sem necessidade de máquina virtual tradicional. Possibilita usar ferramentas, utilitários e aplicações Linux nativamente no sistema Windows, mantendo acesso aos ficheiros do sistema anfitrião. É particularmente útil para programadores que precisam de ambientes de desenvolvimento Linux enquanto trabalham no Windows.

No Windows, instale o Ubuntu via WSL. Execute no terminal o seguinte comando:

```
wsl --install -d ubuntu
```

Servidor Remoto (Acesso Remoto via SSH)

O acesso remoto SSH (Secure SHELL) é um protocolo de rede seguro que permite controlar computadores à distância através de uma ligação encriptada. Permite iniciar sessão num servidor remoto, executar comandos, transferir ficheiros e gerir sistemas, tudo com comunicações protegidas por métodos criptográficos robustos.

Aceda a um servidor Linux via SSH usando o seguinte comando:

```
ssh nome_de_utilizador@endereço_IP_ou_hostname
```

Instalação nativa

Instale o Ubuntu diretamente no computador, usando um dispositivo de arranque (USB) ou CD-ROM com a imagem ISO do Ubuntu.

<https://ubuntu.com/tutorials/install-ubuntu-server#1-overview>

Multipass

O Multipass é uma ferramenta leve desenvolvida pela Canonical (empresa por trás do Ubuntu) que permite criar e gerir máquinas virtuais Linux de forma rápida e simples. Funciona em sistemas Windows, macOS e Linux, oferecendo uma maneira fácil de criar instâncias Ubuntu isoladas para desenvolvimento, teste ou experimentação. Com comandos simples, permite criar máquinas virtuais pré-configuradas, gerir recursos e aceder a elas via SHELL ou SSH. É particularmente útil para programadores que precisam de ambientes Linux consistentes e descartáveis sem a complexidade das soluções de virtualização tradicionais.

Instalação do Multipass

A instalação do Multipass pode ser feita usando as instruções descritas no site oficial <https://multipass.run> (<https://canonical.com/multipass/install>)

Criar uma nova instância:

```
multipass launch -name devops01
```

Aceder a instância criada:

```
multipass shell devops01
```

Parar e Eliminar uma instância

```
multipass stop devops01  
multipass delete devops01 --purge
```

A vantagem de usarmos o Multipass é que ele é rápido, leve e perfeito para ambientes DevOps temporários. Por isso, será o nosso companheiro no decorrer da nossa formação DevOps.

Mais comandos e instruções como usar o Multipass, pode ser encontradas no endereço a seguir:

<https://documentation.ubuntu.com/multipass/en/latest/how-to-guides/>

4. DICAS E ATALHOS

Dicas

- a) Digite qualquer comando seguindo de “--help” (dois traços e a palavra *help*) para ver a documentação ou descrição detalhada do comando.

```
wget --help
```

- b) Outra forma de conseguir documentação oficial dos comandos Linux é o comando *man* (manual). Digite *man* seguido do nome do comando que você precisa de informação

```
man wget
```

- c) Se por algum motivo você preferir guardar as informações do *man* (manual) de algum comando num disco removível ou no disco em PDF para uso posterior, use o seguinte comando:

```
man -t wget | ps2pdf - wget.pdf
```

Atalhos

CTRL + C : Cancela o comando atual em funcionamento.

CTRL + Z : Pausa o comando atual, retorna com “fg” em primeiro plano Linux ou “bg” em segundo plano.

CTRL + D : Termina a sessão atual (similar ao comando “exit”)

CTRL + W : Apaga uma palavra na linha atual

CTRL + U : Apaga a linha inteira

CTRL + R : Busque por um comando recentemente digitado

!! : Repete o último comando

5. COMANDOS ESSENCIAIS LINUX

Gestão de Ficheiros e Diretórios

Lista o conteúdo do diretório atual, incluindo ficheiros ocultos, em formato longo:

```
ls -al
```

Muda para o diretório especificado:

```
cd nome_do_diretorio
```

Muda para o diretório pessoal do utilizador atual:

```
cd
```

Mostra o caminho completo do diretório atual:

```
pwd
```

Cria um novo diretório:

```
mkdir nome_do_diretorio
```

Apaga o ficheiro especificado:

```
rm nome_do_ficheiro
```

Apaga o diretório especificado e todo o seu conteúdo recursivamente:

```
rm -r nome_do_diretorio
```

Força a remoção do ficheiro especificado, sem pedir confirmação:

```
rm -f nome_do_ficheiro
```

Força a remoção do diretório e todo o seu conteúdo recursivamente. **Utilize com extrema precaução!**

```
sudo rm -rf nome_do_diretorio
```

Copia o ficheiro1 para ficheiro2:

```
cp nome_do_ficheiro_1 nome_do_ficheiro_2
```

Copia o diretório1 (e o seu conteúdo) para o diretório2:

```
cp -r nome_do_diretorio_1 nome_do_diretorio_2
```

Move ou renomeia a origem (ficheiro ou diretório) para o destino:

```
Mv nome_ficheiro_diretorio_orig nome_ficheiro_diretorio_dest
```

Cria uma ligação simbólica (atalho) para o alvo:

```
ln -s alvo nome_da_ligacao
```

Cria um ficheiro vazio se não existir, ou atualiza as suas datas de acesso e modificação:

```
touch nome_do_ficheiro
```

Mostra o conteúdo de um ficheiro:

```
cat nome_do_ficheiro
```

Mostra o conteúdo de um ficheiro página a página, permitindo navegação:

```
less nome_do_ficheiro
```

Mostra as primeiras N linhas de um ficheiro (por defeito, 10):

```
head -n NUMERO_DE_LINHAS nome_do_ficheiro
```

Mostra as últimas N linhas de um ficheiro (por defeito, 10):

```
tail -n NUMERO_DE_LINHAS nome_do_ficheiro
```

Mostra as últimas linhas de um ficheiro e continua a exibir o conteúdo à medida que é adicionado.

```
tail -f nome_do_ficheiro
```


Processos e Trabalhos

Mostra os processos atualmente em execução pelo utilizador e outros (formato BSD):

```
ps aux
```

Mostra os processos do sistema em tempo real, ordenados pelo uso de CPU:

```
top
```

Alternativa interativa ao **top**, mais fácil de usar e com mais funcionalidades. (Pode precisar instalar primeiro: **`sudo apt install htop`**):

```
htop
```

Termina o processo com o ID (PID) especificado:

```
kill PID
```

Força o término do processo com o ID (PID) especificado (sinal KILL):

```
kill -9 PID
```

Termina todos os processos com o nome especificado:

```
killall nome_do_processo
```

Coloca um trabalho suspenso em segundo plano:

```
bg
```

Traz o trabalho mais recente (ou especificado) para primeiro plano:

```
fg %JOB_ID
```

Permissões

Altera as permissões de acesso de um ficheiro ou diretório usando notação octal (ex: 755, 644):

```
chmod OCTAL nome_do_ficheiro_or_diretorio
```

Altera o proprietário de um ficheiro ou diretório:

```
sudo chown nome_proprietario nome_ficheiro_ou_diretorio
```

Altera o grupo de um ficheiro ou diretório:

```
sudo chgrp nome_grupo nome_ficheiro_ou_diretorio
```

Conectividade e Rede

Conecta-se a um servidor remoto via SSH:

```
ssh utilizador@ip_ou_hostname
```

Copia a chave pública SSH para um anfitrião remoto, permitindo login sem password:

```
ssh-copy-id utilizador@ip_ou_hostname
```

Busca e filtros

Procura um padrão de texto em ficheiros:

```
grep PATTERN ficheiro_ou_lista_de_ficheiros
```

Procura recursivamente um padrão de texto num diretório.

```
grep -r PATTERN nome_do_directorio
```

Filtra a saída de um comando, mostrando apenas linhas que contêm o padrão:

```
comando | grep PATTERN
```

Localiza rapidamente ficheiros (requer base de dados atualizada com *sudo updatedb*):

```
locate nome_do_ficheiro
```

Mostra a data e hora atuais do sistema:

```
date
```

Mostra o calendário do mês atual:

```
cal
```

Mostra há quanto tempo o sistema está ligado e a carga média:

```
uptime
```

Mostra quem está ligado ao sistema e o que estão a fazer:

```
w
```

Mostra o nome do utilizador atual:

```
whoami
```

Mostra todas as informações do sistema (kernel, nome do host, etc.):

```
uname -a
```

Mostra informações detalhadas sobre a(s) CPU(s):

```
cat /proc/cpuinfo
```

Mostra informações detalhadas sobre a utilização da memória:

```
cat /proc/meminfo
```

Mostra a página de manual (ajuda) para um comando:

```
man comando
```

Mostra o espaço utilizado e livre nos sistemas de ficheiros (formato legível):

```
df -h
```

Mostra o espaço em disco utilizado por ficheiros/diretórios (formato legível e sumário para um diretório):

```
du -sh diretório_ou_ficheiro
```

Mostra a quantidade de memória RAM e swap utilizada e livre (formato legível):

```
free -h
```

Localiza o binário, código fonte e página de manual de uma aplicação:

```
whereis nome_da_aplicacao
```

Mostra o caminho completo do executável da aplicação que seria executada:

```
which nome_da_aplicacao
```

Empacotamento e Compressão

Cria um arquivo TAR não comprimido:

```
tar cf pacote.tar ficheiros_ou_diretórios
```

Extrair ficheiros de um arquivo TAR:

```
tar xf pacote.tar
```

Cria um arquivo TAR comprimido com GZip:

```
tar czf pacote.tar.gz ficheiros_ou_diretórios
```

Extrai um arquivo TAR comprimido com GZip:

```
tar xzf pacote.tar.gz
```

Cria um arquivo TAR comprimido com BZip2:

```
tar cjf pacote.tar.bz2 ficheiros_ou_diretórios
```

Extrai um arquivo TAR comprimido com BZip2:

```
tar xjf pacote.tar.bz2
```

Comprime um ficheiro (substituindo-o por ficheiro.gz):

```
gzip ficheiro
```

Descomprime um ficheiro .gz (substituindo-o pelo original):

```
gunzip ficheiro.gz
```

Testa a conectividade de rede com um anfitrião:

```
ping hostname_or_IP
```

Obtém informações de DNS para um anfitrião:

```
dig hostname
```

Faz uma pesquisa DNS reversa para um endereço IP:

```
dig -x endereco_ip
```

Descarregar um ficheiro a partir de um URL:

```
wget URL_do_ficheiro
```

Continua o descarregamento interrompido de um ficheiro:

```
wget -c URL_do_ficheiro
```

Instalação a Partir do Código Fonte (Compilação Típica)

Prepara o código fonte para compilação no sistema atual (executar no diretório do código fonte):

```
./configure
```

Compila o código fonte (executar após o ./configure):

```
make
```

Instala o programa compilado no sistema (geralmente requer privilégios de root):

```
sudo make install
```

Gestão de Pacotes (Debian/Ubuntu)

Atualiza a lista de pacotes disponíveis dos repositórios:

```
sudo apt update
```

Instala o pacote especificado e as suas dependências:

```
sudo apt install nome_do_pacote
```

Atualiza todos os pacotes instalados para as suas versões mais recentes.

```
sudo apt upgrade
```

Remover o pacote especificado:

```
sudo apt remove nome_do_pacote
```

Remover o pacote e os seus ficheiros de configuração:

```
sudo apt purge nome_do_pacote
```

Remove pacotes que foram instalados automaticamente como dependências e já não são necessários:

```
sudo apt autoremove
```

Procura pacotes disponíveis nos repositórios:

```
apt search termo_a_pesquisar
```

Mostra informação detalhada sobre um pacote:

```
apt show nome_do_pacote
```

Instalar um ficheiro de pacote .deb local:

```
sudo dpkg -i ficheiro_pacote.deb
```

Lista todos os pacotes DEB instalados e filtre os que contêm o texto:

```
dpkg -l | grep "texto"
```

Informações do Sistema

Mostra a arquitetura da máquina (ex: x86_64):

```
uname -m
```

Mostra a versão do kernel em utilização:

```
uname -r
```

Mostra estatísticas das interfaces de rede:

```
cat /proc/net/dev
```

Mostra os sistemas de ficheiros atualmente montados:

```
cat /proc/mounts
```

Mostra os dispositivos PCI numa vista em árvore:

```
lspci -tv
```

Mostra os dispositivos USB numa vista em árvore:

```
lsusb -tv
```

Define a data e hora do sistema (MMDDhhmmYYYY.ss):

```
sudo date MMDDhhmmAAAA.ss
```

Guarda a hora do sistema no relógio de hardware:

```
sudo hwclock --systohc
```

Desligar e Reiniciar

Desliga o sistema imediatamente.

```
sudo poweroff
```

Agenda o desligamento do sistema para daqui a N minutos

```
sudo shutdown -h +N_minutos "mensagem_opcional"
```

Reinicia o sistema imediatamente.

```
sudo reboot
```

Agenda o reinício do sistema para daqui a N minutos.

```
sudo shutdown -r +N_minutos "mensagem_opcional"
```

Cancela um desligamento ou reinício agendado com **shutdown**.

```
sudo shutdown -c
```

Termina a sessão atual do shell.

```
exit
```

Utilitários de Disco e Sistema de Ficheiros

Monta uma partição num ponto de montagem.

```
sudo mount /dev/particao /mnt/ponto_montagem
```

Desmonta um sistema de ficheiros.

```
sudo umount /mnt/ponto_montagem_ou_dispositivo
```

Monta um ficheiro de imagem ISO.

```
sudo mount -o loop ficheiro.iso /mnt/ponto_montagem_iso
```

Verifica e repara (interativamente) a integridade de um sistema de ficheiros (numa partição desmontada!).

```
sudo fsck /dev/particao_desmontada
```

Formata uma partição com o sistema de ficheiros ext4. **Apaga todos os dados!**

```
sudo mkfs.ext4 /dev/particao
```

Configura uma partição para ser usada como área de swap.

```
sudo mkswap /dev/particao_swap
```


Ativa uma área de swap.

```
sudo swapon /dev/particao_swap
```

Redes (Configuração e Diagnóstico)

Mostra a configuração das interfaces de rede.

```
ip addr show
```

Ativa uma interface de rede (ex: eth0, wlan0).

```
sudo ip link set interface up
```

Desativa uma interface de rede.

```
sudo ip link set interface down
```

Configura um endereço IP e máscara de rede numa interface (ex: 192.168.1.100/24).

```
sudo ip addr add endereco_ip/mascara_cidr dev interface
```

Obtém uma configuração de IP via DHCP para uma interface.

```
sudo dhclient interface
```

Mostra a tabela de encaminhamento (rotas) do sistema.

```
ip route show
```

Configura o gateway padrão.

```
sudo ip route add default via IP_do_Gateway
```

Mostra todas as ligações de rede ativas e a escutar (TCP e UDP) com os PIDs dos processos.

```
sudo ss -tulnp
```

Captura e mostra tráfego de rede (ex: na interface eth0, para a porta 80).

```
sudo tcpdump -i interface PORT PORT
```

Monitorização e Depuração

Mostra mensagens do buffer do kernel (útil para problemas de hardware/arranque).

```
dmesg
```

Mostra mensagens de log do sistema em tempo real (para sistemas com systemd).

```
journalctl -f
```

Lista os módulos do kernel carregados.

```
lsmod
```

Mostra informação geral e estado SMART de um disco (ex: /dev/sda). (Pode precisar de: *sudo apt install smartmontools*)

```
sudo smartctl -i /dev/sda
```

Mostra todos os atributos SMART de um disco.

```
sudo smartctl -A /dev/sda
```

Dicas e Utilitários Adicionais

Procura comandos relacionados com uma palavra-chave nas páginas de manual.

```
apropos palavra_chave
```

Mostra uma breve descrição de um comando.

```
whatis comando
```

Encripta um ficheiro usando uma palavra-passe simétrica. (Pode precisar de: *sudo apt install gnupg*)

```
gpg -c ficheiro
```

Desencripta um ficheiro .gpg.

```
gpg ficheiro.gpg
```

Mostra as bibliotecas partilhadas requeridas por um programa.

```
ldd /caminho/para/programa
```

Define um alias temporário para um comando. (Para persistência, adicione a `~/.bashrc`)

```
Alias atalho='comando_completo'
```

Limpa o ecrã do terminal.

```
clear
```

Executa um comando e redireciona STDOUT e STDERR para um ficheiro.

```
comando ficheiro_saida.txt 2>&1
```

Executa um comando, mostra a saída no ecrã e guarda-a num ficheiro.

```
comando | tee ficheiro_saida.txt
```

REFERÊNCIAS

<https://www.sutori.com/es/historia/a-historia-dos-sistemas-operativos--R7u2kMxRUYicDfVVm3Xu8Myn>

<https://a3aengenharia.com.br/conteudo/artigos-tecnicos/sistemas-operacionais/>

<https://pt.wikipedia.org/wiki/Fdisk>

<https://guialinux.uniriotec.br/sistemas-de-arquivos/>

<https://www.ovhcloud.com/pt/learn/what-is-ubuntu/>