

Início

Revisão da aula passada

Quais os benefícios de usar linux?

O que é uma distribuição?

Qual a primeira coisa que você faria pra instalar linux agora?

Qual distro instalamos

Abram o virtual box

Objetivo

A arvore de arquivos e como isso se relaciona com o terminal

O que são Usuário, grupo e permições

O que são processos, e como interrompelos (control+c e sig9)

O que é IPC e o que é um sinal

O que são Package managers e o que fazem

Importância da aula atual

Terminal

Servidores e outras instalações sem gráficos, package managers, e contato direto com a shell

Tem mais ferramentas pra command line pq são mais rápidas de desenvolver

É por onde administramos o sistema

Administração do sistema

Entender os conceitos de administração do sistema vai evitar problemas futuros como não conseguir editar ou executar arquivos importantes além de serem os fundamentos necessários pra administrar sistemas mais complexos como servidores e sistemas distribuídos

Package managers

Instalação mais simples e rápida de aplicações e ferramentas

Digitar é mais rápido do que clicar

Corpo

Teoria e prática

Shell

A shell é nossa camada de contato com o sistema operacional

Terminal

O que é um emulador de terminal

A árvore de arquivos

Lembre onde você está o tempo todo
pwd e convertendo uma path pra uma árvore

A diferença entre paths absolutas e relativas
Paths absolutas mostram o caminho completo até o arquivo ou diretório
Paths relativas mostram o caminho entre o ponto que você está e um arquivo ou diretório

ls + uma path (relativa ou absoluta) nos mostra todos os arquivos e diretórios que existem numa pasta, se nenhuma path for dada a padrão é .

Podemos criar um arquivo com touch e usar cat pra ler o arquivo
Echo seguido de >> pode ser usado pra adicionar texto ao final da linha

mkdir é usado pra criar diretórios

editando um arquivo com nano

Tipos de argumentos comuns
posicionais, flags, keywords (baseados em flags e com iguais)
Quem define que tipo de argumento é usado é o programa

ls -a
-h, --help e man

stdin, stdout e como isso nos permite usar o pipe

pipelining

analisando e mantando processos
ps, ps aux, kill
IPC

Usuários e permissões

O que é um usuário
root
O super user e o comando sudo
O que sudo faz

Adicionando um usuário

O que são grupos
O que é wheel
/etc/sudoers

Permissões em arquivo e diretórios

permissões em arquivos e diretórios (rwx)
chmod chown

Package managers

O que um package manager faz
O package manager é responsável por instalar um 'pacote' e suas dependências
Um pacote é código do software em conjunto com instruções de como instalá-lo (quais as bibliotecas necessárias para linkagem por exemplo)
A instalação pode ser feita pelo terminal com um único comando e o package manager lidará com as dependências necessárias

Por que esse modelo é melhor do que os instaladores?
Um único comando baixa e instala nosso programa e suas dependências, sem que você tenha que se preocupar em executar um instalador

Exercícios

Tentem fazer algo do seu dia a dia pelo terminal.
Pros programadores, desenvolva um programa
Pros analistas de dados, rodem python como uma shell
Pratiquem os comandos ensinados hoje
Procurem e instalem pacotes que vocês já usam (spotify, steam, python, node)

Pra se aprofundar na teoria

[30 comandos pra dominar o linux](#)

[Bash tutorial](#)

[Como usar apt](#)

[bash avançado](#)

[IPC](#)

Conclusão

como isso se conecta com o resto?

O terminal é uma ferramenta essencial pra qualquer programador, o domínio do terminal permite que o programador interaja com o sistema operacional de

forma mais direta e é a base pra capacidades mais avançadas como administração do sistema e uso de servidores e outras instalações sem interface gráfica. Já os package managers são ferramentas responsáveis por um dos maiores ganhos de produtividade do uso de sistemas linux

Revisão dos conteúdos

O que é shell, bash

pwd

cd

paths absolutas e relativas

ls

touch

cat

echo >

nano

nvim

grep

pipe

stdin, stdout

Argumentos e arquivos escondidos

ls -a

-h, --help e man

sudo

instalando programas com apt

bibliografia

[como apt funciona](#)

[como sinais funcionam](#)

[Programas não são processos](#)