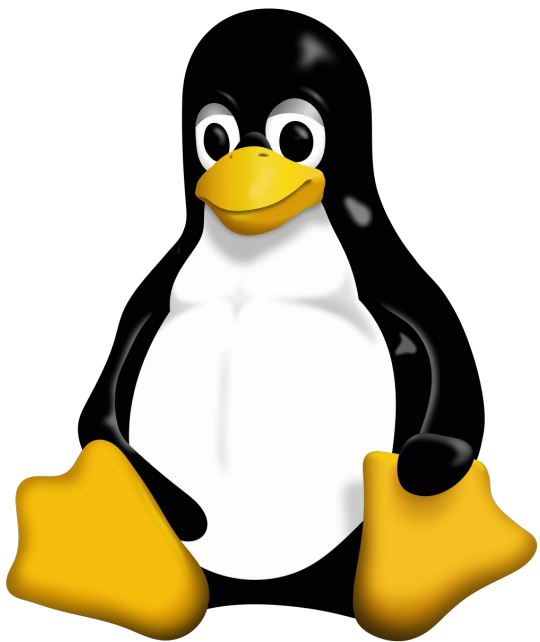


# Minicurso de Linux

Vamo começar esse role!

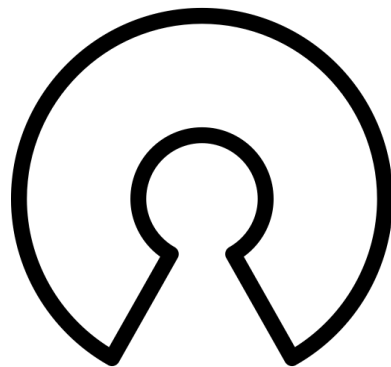


# Introdução

# O que é o Linux?

- Sistema Operacional baseado em Unix
- Gratuito
- OpenSource
- Diversas distribuições ou “sabores” diferentes
- Criado por Linus Torvalds

UNIX®



# História do Linux

- Em 1991, o então estudante Linus Torvalds inicia seu projeto pessoal de criação de um novo núcleo de sistema operacional
- Lançamento da primeira versão em setembro de 1991
- Primeira versão recebeu diversas críticas, incluindo comentários negativos vindos de Andrew S. Tanenbaum
- O desenvolvimento do Linux continuou, com um grande número de colaboradores, licenciado pela GPL (GNU Public License)



# Arquitetura do Linux

- Kernel é a camada responsável pela interface entre o Hardware e as aplicações
- A Biblioteca de Funções Padrão permite o acesso a recursos através da execução de chamadas feitas por processos a funções padrão suportadas pelo Kernel



# Arquitetura do Linux

- Shell é um processo que fornece ao usuário uma interface para acessar os serviços do SO (pode ser de linha de comando ou gráfico)
- Aplicações são programas com os quais o usuário interage (o Shell também é uma aplicação)

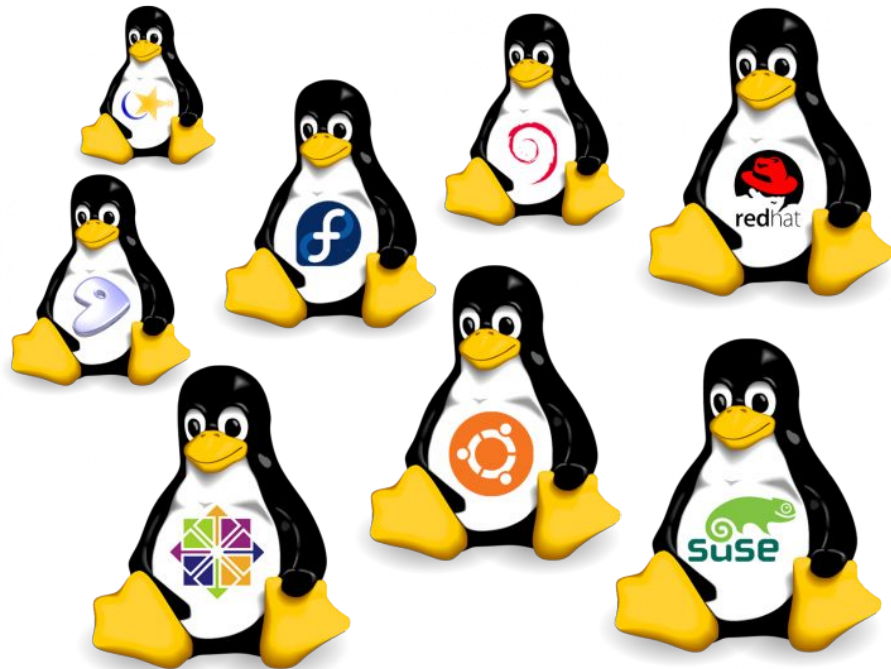


# Vantagens do Linux

- É gratuito e open source
- O usuário tem total controle sobre o sistema
- Fácil de utilizar
- É altamente personalizável
- É seguro e estável
- Roda bem em computadores “modestos”
- *E por aí vai...*



# Distribuições



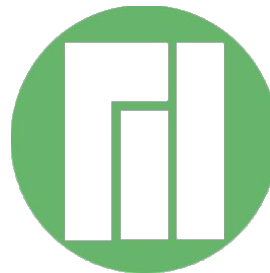
- Cada distribuição é um diferente “sabor” de Linux
- Trata-se de um conjunto de softwares elaborados em torno do kernel do Linux, tornando essas distribuições diferentes entre si



# Distribuições

## Ubuntu

- Uma das distribuições mais populares
- Muito amigável a novos usuários, o que faz dela uma boa primeira distribuição



## Manjaro

- É baseada no Arch Linux e possui 100% de compatibilidade com o seu repositório (AUR - Arch User Repository)
- É distribuído em três versões distintas (cada uma com um ambiente gráfico diferente)

# Distribuições

## Debian

- Possui um ótimo sistema de gerenciamento de pacotes
- Alta quantidade de softwares



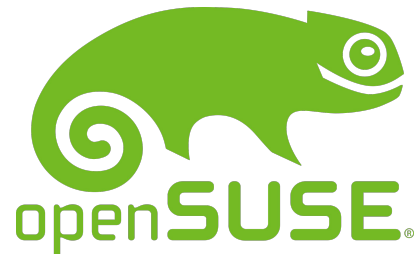
## Mint

- É baseada no Ubuntu e no Debian
- Possui uma interface fácil de utilizar
- Rápida, estável e prática

# Distribuições

## Fedora

- Possui atualizações frequentes
- Possui excelentes aplicativos de segurança
- É muito inovadora



## OpenSUSE

- Possui grande estabilidade e atualizações constantes
- É mais focada para empresas

# Por que usar o Ubuntu?

- Fácil de instalar e de configurar
- Possui vasto material com dicas e tutoriais na internet
- A cada 18 meses é lançada uma nova versão LTS (Long Term Support), que tem atualizações de segurança garantidas por 5 anos
- Possui diversos programas já pré-instalados
- Facilidade na busca e na instalação de programas



# Ambiente de Utilização

VirtualBox

# Formas de Utilização

- Instalação em uma máquina;
  - Instalação normal
  - Dual boot
- Live CD / Pendrive
  - em modo de teste
  - configurar ordem de boot
  - Criação de mídia de instalação
- **Máquina Virtual**
  - modo de teste

# Dual Boot

- Instalação de dois sistemas operacionais na mesma máquina;
- Quando se realiza a instalação de ambos os sistemas aparece uma tela de seleção durante o boot da máquina para selecionar qual sistema operacional quer ser utilizado;
- Tem melhor desempenho do que rodar em uma máquina virtual
- Melhor configurável para sua própria máquina;
- Processo que pode dar erros e pode interferir no funcionamento anterior da máquina;
- Semelhante ao processo de formatação;
- FAÇAM BACKUP se forem tentar.

# VirtualBox

- Software de virtualização
- Permite a execução de outros sistemas operacionais separado do que está funcionando na máquina
- Tem um desempenho pior do que no dual boot
- Para abranger todas as funcionalidades do sistema operacional, deve-se configurar vários detalhes







Downloads



# Security, cloud native & confidential computing

A deep dive into securing against the threats of cyber attacks and data breaches

Register for the webinar



Find out what's new in Ubuntu 20.10 in our webinar

## Ubuntu Desktop >

Download Ubuntu desktop and replace your current operating system whether it's Windows or Mac OS, or, run Ubuntu alongside it.

20.04 LTS

20.10

## Ubuntu Server >

The most popular server Linux in the cloud and data centre, you can rely on Ubuntu Server and its five years of guaranteed free upgrades.

Get Ubuntu Server

[Mac and Windows](#)

[ARM](#)

[IBM Power](#)

[s390x](#)

## Ubuntu for IoT >

Are you a developer who wants to try snappy Ubuntu Core or classic Ubuntu on an IoT board?

[Raspberry Pi 2, 3 or 4](#)

[Intel NUC](#)

[KVM](#)

[Qualcomm Dragonboard 410c](#)

[UP² IoT Grove](#)

[Intel IEI TANK 870](#)

## Ubuntu Cloud >

Use Ubuntu optimised and certified server images on most major clouds.

[Get started on Amazon AWS, Microsoft Azure, Google Cloud Platform and more...](#)

[Download cloud images for local development and testing](#)

### TUTORIALS

If you are already running Ubuntu - you can [upgrade](#) with the Software Updater

Burn a DVD on [Ubuntu](#), [macOS](#), or

### READ THE DOCS

Read the official docs for [Ubuntu Desktop](#), [Ubuntu Server](#), and [Ubuntu Core](#)

### OTHER WAYS TO DOWNLOAD

Ubuntu is available via [BitTorrents](#) and via a minimal [network installer](#) that allows you to customise what is installed, such as additional languages. You can also find [older releases](#).

### UBUNTU FLAVOURS

Find new ways to experience Ubuntu, each with their own choice of default applications and settings.

[Kubuntu](#)

[Lubuntu](#)

[Ubuntu MATE](#)

[Ubuntu Studio](#)



# VirtualBox

## Download VirtualBox

search...  
[Login](#) [Preferences](#)

[About](#)  
[Screenshots](#)  
[Downloads](#)  
[Documentation](#)  
    [End-user docs](#)  
    [Technical docs](#)  
[Contribute](#)  
[Community](#)

Here you will find links to VirtualBox binaries and its source code.

### VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see [VirtualBox 6.0 builds](#). Please also use version 6.0 if you need to run VMs with software virtualization, as this has been discontinued in 6.1. Version 6.0 will remain supported until July 2020.

If you're looking for the latest VirtualBox 5.2 packages, see [VirtualBox 5.2 builds](#). Please also use version 5.2 if you still need support for 32-bit hosts, as this has been discontinued in 6.0. Version 5.2 will remain supported until July 2020.

### VirtualBox 6.1.16 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- [SHA256 checksums](#), [MD5 checksums](#)

**Note:** After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

# Criação de uma Máquina Virtual





Ferramentas



Ubuntu (Snapshot 3)



Salva



Preferências



Importar



Exportar



Novo



Novo (Ctrl+N)

**Bem-Vindo ao VirtualBox!**

A parte esquerda desta janela contém ferramentas globais, e lista todas as máquinas virtuais disponíveis em seu computador. Você pode importar, acrescentar e criar novas VMs utilizando os botões correspondentes na barra de ferramentas.

Você pode pressionar **F1** para obter ajuda instantânea, ou visitar [www.virtualbox.org](http://www.virtualbox.org) para obter informações e notícias sobre o VirtualBox.



?×

← Criar Máquina Virtual


Nome e Sistema Operacional

Escolha um nome descritivo para a nova máquina virtual e selecione o tipo de sistema operacional que você pretende instalar nela. O nome que você escolher será utilizado pelo VirtualBox para identificar esta máquina.

Nome:

UbuntuMinicurso

Pasta da Máquina:

 C:\Users\Carrefuor\VirtualBox VMs

▼

Tipo:


Linux

▼

Versão:

Ubuntu (64-bit)

▼



Modo Expert

Próximo (N)

Cancelar


?×

← Criar Máquina Virtual

Tamanho da memória

Selecione a quantidade de memória (RAM) em megabytes que será alocado para a máquina virtual.

O tamanho recomendado para memória é de **1024MB**.



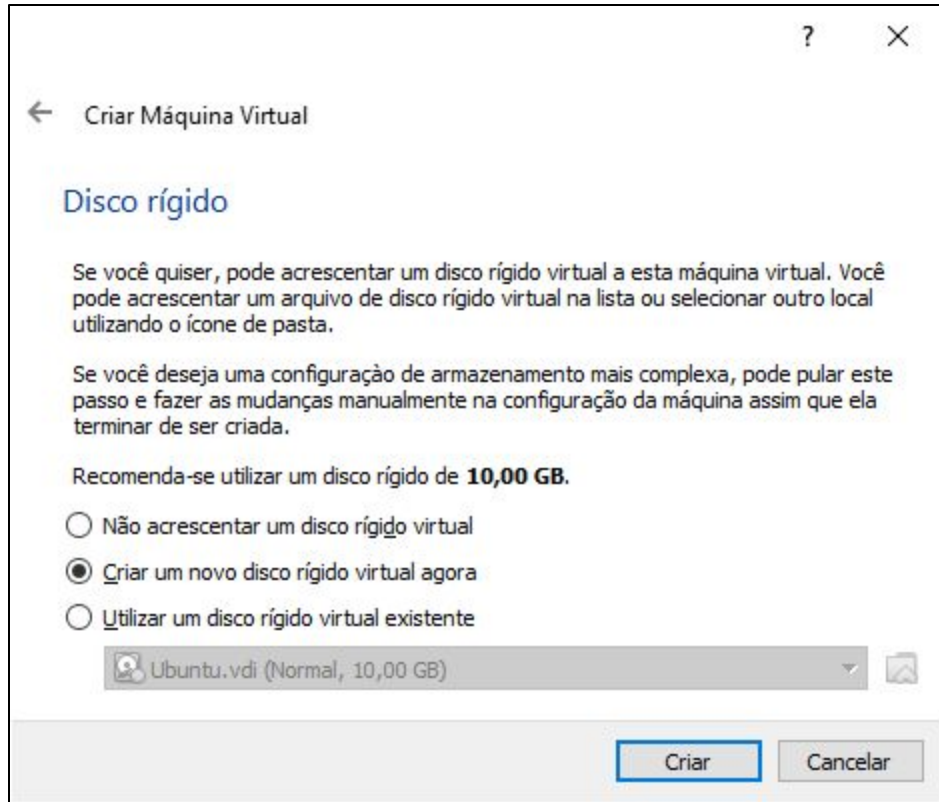
A horizontal slider bar for selecting memory size. The bar has a green segment on the left and a red segment on the right. A blue arrow points to the slider. Below the bar, '4 MB' is on the left and '8192 MB' is on the right. To the right of the slider is a numeric input field containing '2048' and a 'MB' unit label.

4 MB 8192 MB 2048 MB

Próximo (N)

Cancelar







## ← Criar Disco Rígido Virtual

### Tipo de arquivo de disco rígido

Escolha o tipo de arquivo que você gostaria de utilizar para o novo disco rígido virtual. Caso não necessite utilizá-lo com outros softwares de virtualização, pode deixar esta opção como está.

- ☒ VDI (VirtualBox Disk Image)
- ☐ VHD (Virtual Hard Disk)
- ☐ VMDK (Virtual Machine Disk)

Modo Expert

Próximo (N)

Cancelar



## Criar Disco Rígido Virtual

### Armazenamento em disco rígido físico

Escolha se o arquivo contendo o disco rígido virtual deve crescer à medida em que é utilizado (dinamicamente alocado) ou se ele deve ser criado já com o tamanho máximo (tamanho fixo).

Um arquivo de disco rígido virtual **dinamicamente alocado** irá utilizar espaço em seu disco rígido físico à medida em que for sendo utilizado (até um **tamanho máximo pré-definido**), mas não irá encolher caso seja liberado espaço nele.

Um arquivo de disco rígido virtual de **tamanho fixo** pode levar mais tempo para ser criado em alguns sistemas, mas geralmente possui acesso mais rápido.

☒ Dinamicamente alocado

☐ Tamanho Fixo

Próximo (N)

Cancelar



## ← Criar Disco Rígido Virtual

### Localização e tamanho do arquivo

Informe o nome do arquivo em disco que conterá o disco virtual no campo abaixo ou clique no ícone da pasta para selecionar uma localização diferente para o arquivo.

C:\Users\Carrefuor\VirtualBox VMs\UbuntuMinicurso\UbuntuMinicurso.vdi



Selecione o tamanho da imagem de disco virtual em megabytes. Este tamanho é o limite máximo de dados que uma máquina virtual poderá armazenar neste disco rígido.



Criar

Cancelar



Ferramentas



Ubuntu (Snapshot 3)

Salva



UbuntuMinicurso

Desligada



Criar



Apagar (D)



Restaurar



Propriedades



Clonar



Configurações



Descartar



Iniciar (T)

Nome

Criado



Estado Atual

Atributos

Informação

Nome:

Entre com o nome para o novo snapshot...

Descrição:

Criar

Desfazer

# Configurações VirtualBox





Geral



Sistema



Monitor



Armazenamento



Áudio



Rede



Portas Seriais



USB



Pastas Compartilhadas



Interface do Usuário

**Geral**

Básico

Avançado

Descrição

Criptografia de Disco

Nome: UbuntuMinicurso

Tipo: Linux



Versão: Ubuntu (64-bit)

OK

Cancelar



Geral



Sistema



Monitor



Armazenamento



Áudio



Rede



Portas Seriais



USB



Pastas Compartilhadas



Interface do Usuário

## Armazenamento

### Dispositivos de Armazenamento



Controladora: IDE



Vazio



Controladora: SATA



UbuntuMinicurso.vdi



### Atributos

Nome: IDE

Tipo: PIIX4

☒ Utilizar cache de I/O do hospedeiro

OK

Cancelar





Ferramentas



Criar



Apagar (D)



Restaurar



Propriedades



Clonar



Configurações



Descartar



Iniciar (T)



Ubuntu (Snapshot 3)

Salva



UbuntuMinicurso

Desligada

## UbuntuMinicurso - Configurações



Geral



Sistema



Monitor



Armazenamento



Áudio



Rede



Portas Seriais



USB



Pastas Compartilhadas



Interface do Usuário

## Armazenamento

Dispositivos de Armazenamento



Controladora: IDE



Vazio



Controladora: SATA



UbuntuMinicurso.vdi

Atributos

Nome: IDE

Acrescenta Disco Óptico

☒ Utilizar cache de I/O do hospedeiro

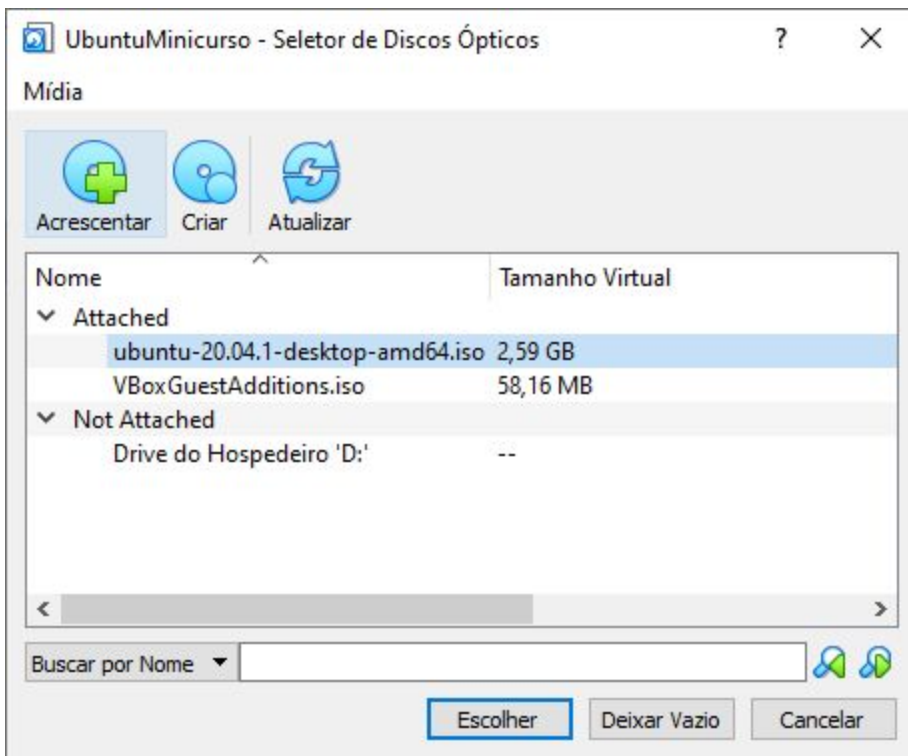
OK

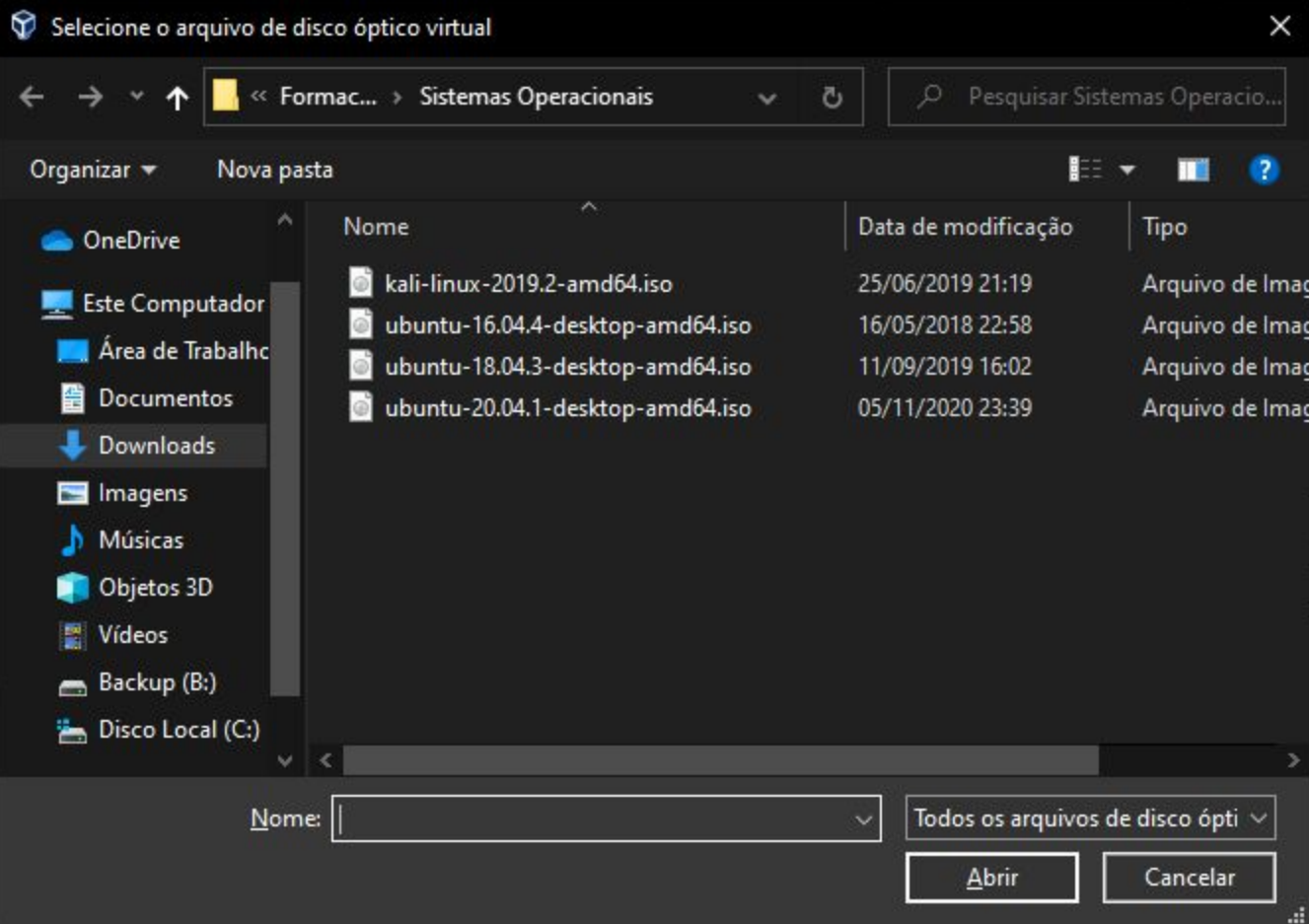
Cancelar

Criar

Desfazer

POR  
PTB215:39  
10/11/2020







Geral



Sistema



Monitor



Armazenamento



Áudio



Rede



Portas Seriais



USB



Pastas Compartilhadas



Interface do Usuário

## Armazenamento

### Dispositivos de Armazenamento



Controladora: IDE



ubuntu-20.04.1-desktop-amd64.iso



Vazio



Controladora: SATA



UbuntuMinicurso.vdi



### Atributos

Nome: IDE

Tipo: PIIX4

☒ Utilizar cache de I/O do hospedeiro

OK

Cancelar

# Na Máquina Virtual



## Instalação



## Bem-vindo

Galego  
Hrvatski  
Íslenska  
Italiano  
Kurdî  
Latviski  
Lietuviškai  
Magyar  
Nederlands  
No localization (UTF-8)  
Norsk bokmål  
Norsk nynorsk  
Polski  
Português  
**Português do Brasil**  
Română  
Sámegiellii  
Shqip

**Experimentar o Ubuntu**

Você pode experimentar o Ubuntu sem fazer quaisquer alterações em seu computador, diretamente a partir deste CD.

Ou se você estiver pronto, pode instalar o Ubuntu ao lado (ou no lugar) do seu sistema operacional atual. Isso não levará muito tempo.

Você talvez queira ler as [notas da versão](#).

**Instalar o Ubuntu**



ubuntu



Trash



 Instalar o  
Ubuntu 20.04.1  
LTS





## Configurações



idioma



## Região &amp; idioma

Selecione o seu idioma de exibição, formatos, fontes de entradas e disposições de teclado

## Rede



## Cabeada



Conectado - 1000 Mb/s



## VPN



Não configurado

Proxy de rede

Desativada







Configurações



Região &amp; idioma



Rede



Bluetooth



Plano de fundo



Aparência



Notificações



Pesquisa



Aplicativos &gt;



Privacidade &gt;



Contas online



Compartilhar



Som



Energia

Idioma

English (United States)

Formatos

United States (English)

**Fontes de entrada**

Escolha layouts de teclado ou métodos de entrada.



Inglês (EUA)



Gerir idiomas instalados



Configurações



Região &amp; idioma



Rede



Bluetooth



Plano de fundo



Aparência



Notificações



Pesquisa



Aplicativos &gt;



Privacidade &gt;



Contas online



Compartilhar



Som



Energia

Idioma

English (United States)

Formatos

Brasil

**Fontes de entrada**

Escolha layouts de teclado ou métodos de entrada.



Inglês (EUA)



Português (Brasil)



+

Gerir idiomas instalados



## Configurações



Rede



Bluetooth



Plano de fundo



Aparência



Notificações



Pesquisa



Aplicativos



Privacidade



Contas online



Compartilhar



Som



Energia



Reconfigurar



Inglês (EUA)

en

• Português (Brasil)

pt

Exibir disposição de teclado

Idioma

Formatos

Brasil

## Fontes de entrada



Escolha layouts de teclado ou métodos de entrada.



Inglês (EUA)

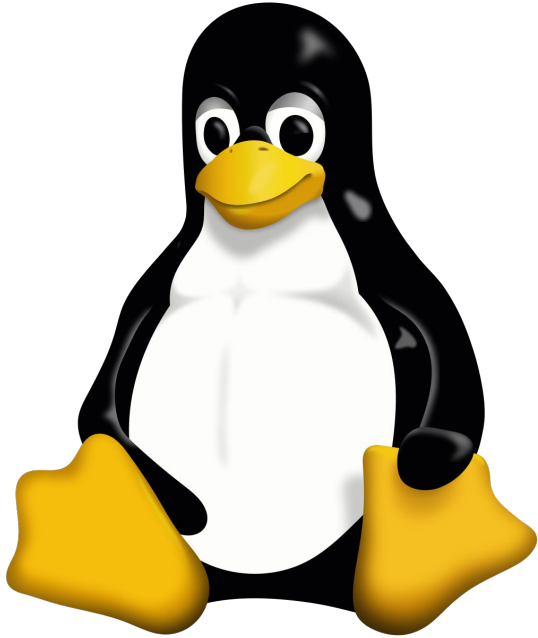


Português (Brasil)



+

Gerir idiomas instalados



# Diretórios e Arquivos

# Sistemas de Arquivos

- Sistema de arquivos é um conjunto de estruturas lógicas que permite ao sistema operacional ter acesso e controlar os dados gravados no disco;
- O sistema operacional Linux possui suporte a diversos sistemas de arquivos, como:
  - ext3
  - ext4
  - xfs
  - jfs

# Diretório Raiz

- É o diretório principal do sistema;
- Dentro dele estão todos os outros diretórios do sistema, chamados de sub-diretórios;
- Representação: /;

# Estrutura básica de diretórios

## **/bin**

- Contém arquivos de programas do sistema que são usados com frequência pelos usuários.

## **/boot**

- Contém arquivos necessários para a inicialização do sistema.

## **/cdrom**

- Ponto de montagem da unidade de CD-ROM.

# Estrutura básica de diretórios

## **/media**

- Ponto de montagem de dispositivos diversos do sistema (rede, pen-drives, CD-ROM em distribuições mais novas).

## **/dev**

- Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.

## **/etc**

- Arquivos de configuração de seu computador local.



# Estrutura básica de diretórios

## **/home**

- Diretórios contendo os arquivos dos usuários.

## **/lib**

- Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.

## **/lost+found**

- Local para a gravação de arquivos/diretórios recuperados pelo utilitário fsck.ext2. Cada partição possui seu próprio diretório lost+found.

# Estrutura básica de diretórios

## **/mnt**

- Ponto de montagem temporário.

## **/proc**

- Sistema de arquivos do kernel. Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam configurações do sistema ou modificar o funcionamento de dispositivos do sistema através da alteração em seus arquivos.

# Estrutura básica de diretórios

## **/sbin**

- Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.

## **/tmp**

- Diretório para armazenamento de arquivos temporários criados por programas.

# Estrutura básica de diretórios

## **/usr**

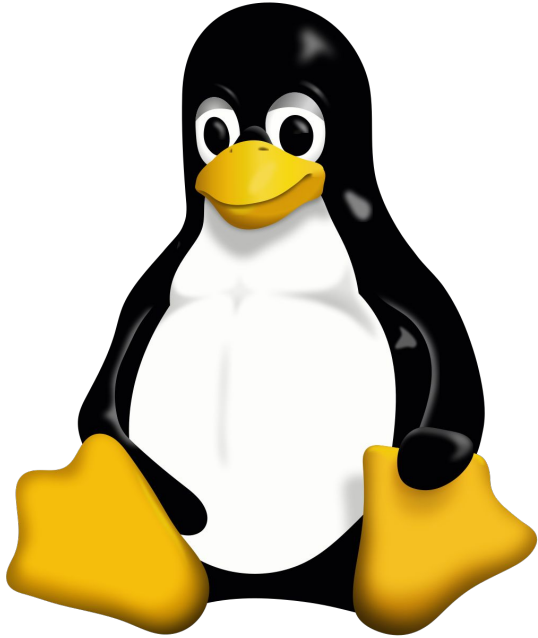
- Contém maior parte de seus programas. Normalmente acessível somente como leitura.

## **/var**

- Contém maior parte dos arquivos que são gravados com frequência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

## **/root**

- Diretório do usuário root.



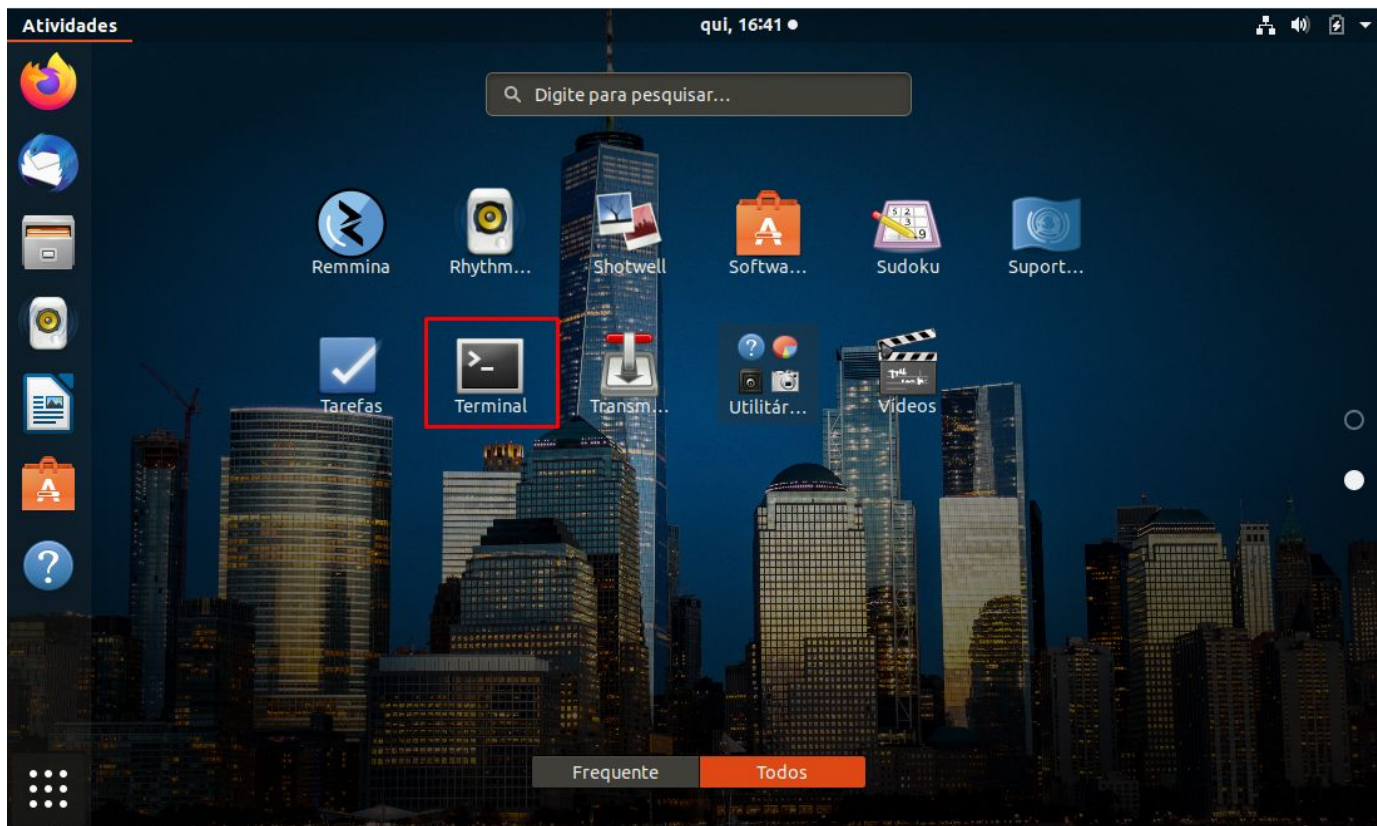
Terminal

# Terminal

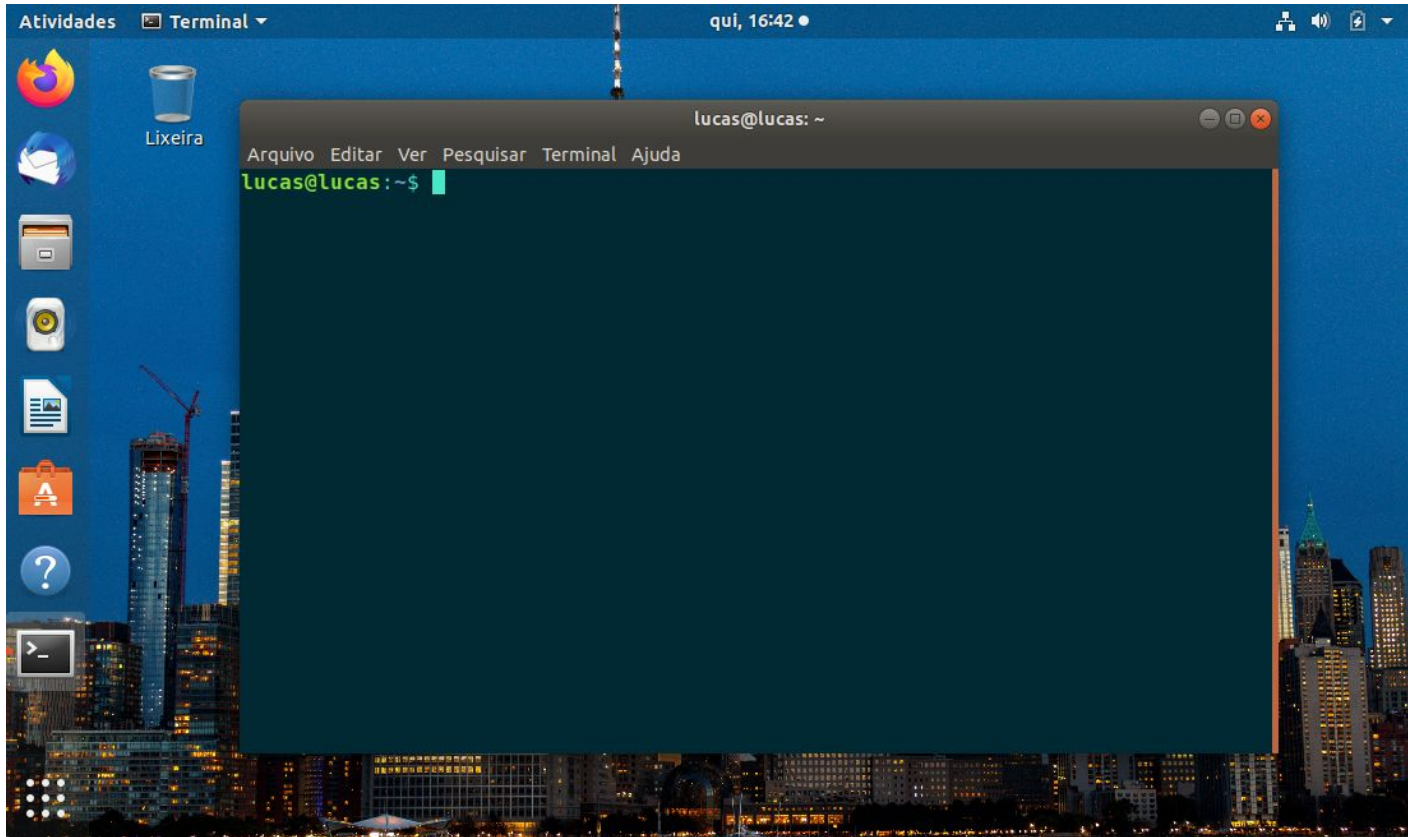
- Através do terminal o usuário pode ter acesso a todo o sistema
  - Diretórios;
  - Arquivos;
  - Aplicações.
- Como acessar o terminal:
  - Ctrl + Alt + t
  - Menu de aplicações



# Terminal



# Terminal





# Terminal

```
lucas@lucas:~$
```

Usuário @ Host:

Diretório

Tipo de usuário

- \$: Usuário comum
- #: Super Usuário

# Terminal

## **Usuário:**

Identifica com qual usuário estamos conectados no terminal;

## **Host:**

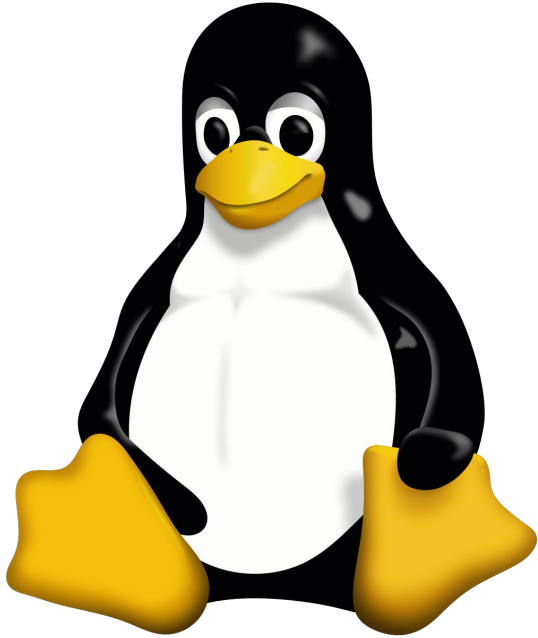
Identifica a qual host (computador) o terminal está conectado;

## **Diretório:**

Identifica em qual diretório (pasta) o terminal está navegando no momento;

## **Tipo de usuário:**

Identifica o nível de privilégio do usuário atual;



# Comandos Essenciais

# Diretórios

- Para navegar entre os diretórios utiliza-se o comando `cd` (change directory);
- Representações:
  - Diretório atual: `.`

```
lucas@lucas:~/Documentos$ cd .  
lucas@lucas:~/Documentos$
```

- Diretório superior: `..`

```
lucas@lucas:~/Documentos$ cd ..  
lucas@lucas:~$
```

# Diretórios

- Representações:
  - Diretório do usuário: ~

```
lucas@lucas:~/Documentos$ cd ~  
lucas@lucas:~$
```

- Diretório raiz: /

```
lucas@lucas:~/Documentos$ cd /  
lucas@lucas:/$
```

- Diretório anteriormente acessado: -

```
lucas@lucas:~/Documentos$ cd -  
/home/lucas  
lucas@lucas:~$
```

# Caracteres Reservados

Caractere	Finalidade/Significado	Exemplo
~	Home do usuário atual  Escape do caractere seguinte:	cd ~ ls -la ~  touch my\*file myfile mylfile (cria um arquivo my*file)
/	Separação de Diretórios	cd /etc/apache2 ls /var/log
\$	Variável	echo \$PATH
?	Existe um caractere qualquer	ls my?file
'	Cota absoluta	echo 'não substitui: \$PATH'
`	Executa comando/Substituição	NOW=`date` ; echo \$NOW
"	Cota dupla	echo "Este é o PATH: \$PATH"
*	Qualquer caractere, nenhuma ou várias vezes	ls m*file
&	Envia processo para o background	xeyes&

# Caracteres Reservados

Caractere	Finalidade/Significado	Exemplo
&&	Operador AND curto-circuito: se cmd1 executar com sucesso executa o segundo, senão encerra	echo 1 && echo 2 fake && echo 2
	Pipe: Redireciona a saída de um comando para outro comando	ls ~   xargs wc
	Operador OR curto-circuito: Se primeiro comando com sucesso não executa o segundo comando	echo 1    echo 2 fake    echo 2
;	Executa os comandos em sequência	cmd1; cmd2; cmd3... echo "O path é: "; echo \$PATH echo "O home é: "; echo \$HOME
[]	Range de caracteres	\$ touch file1 file2 file3 file4 \$ ls file[1-4]
>	Redireciona saída	Cmd > out_file.txt
<	Redireciona entrada	Cmd < in_file.txt

# Comandos Básicos

Para obter informações detalhadas dos comandos pode-se digitar:

- `<comando> --help`
- `man <comando>`
- `whatis <comando>`

## Exemplos:

```
lucas@lucas:~$ ls --help
```

```
lucas@lucas:~$ man ls
```

```
lucas@lucas:~$ whatis ls
ls (1)                - list directory contents
```



# Comandos Básicos

- Listar diretórios e arquivos:
  - ls

```
lucas@lucas:~$ ls
'Área de Trabalho'  Downloads  Imagens  Música  Vídeos
Documentos          examples.desktop  Modelos  Público
```

- Mudar do diretório atual para o especificado:
  - cd <diretório>

```
lucas@lucas:~$ cd Downloads
lucas@lucas:~/Downloads$
```

# Comandos Básicos

- Exibir caminho do diretório atual:
  - pwd

```
lucas@lucas:~/Downloads$ pwd  
/home/lucas/Downloads
```

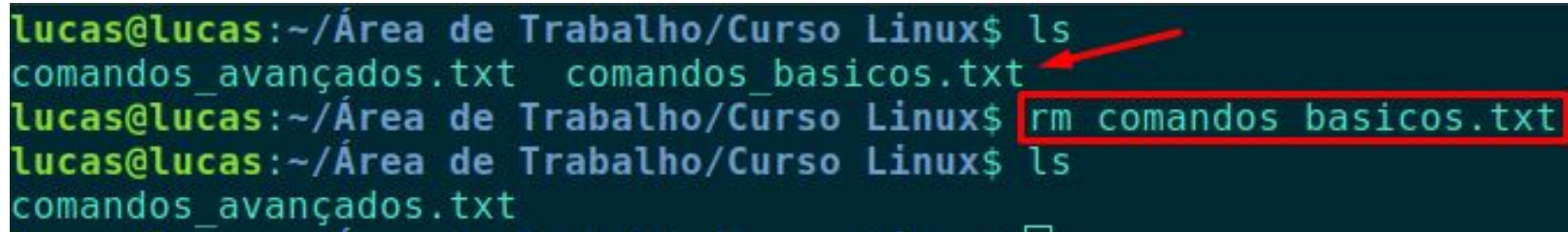
- Criar um novo diretório:
  - mkdir <diretório>

```
lucas@lucas:~/Área de Trabalho$ mkdir 'Curso Linux'  
lucas@lucas:~/Área de Trabalho$ ls  
'Curso Linux'
```

# Comandos Básicos

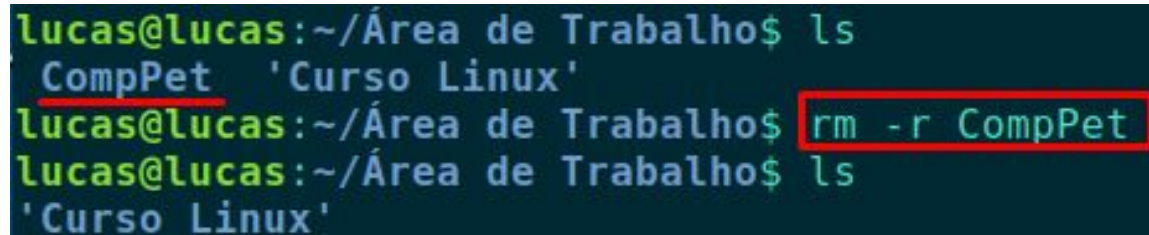
- Apagar arquivo especificado:
  - `rm <arquivo>`

```
lucas@lucas:~/Área de Trabalho/Curso Linux$ ls
comandos_avançados.txt  comandos_basicos.txt
lucas@lucas:~/Área de Trabalho/Curso Linux$ rm comandos_basicos.txt
lucas@lucas:~/Área de Trabalho/Curso Linux$ ls
comandos_avançados.txt
```



- Apagar diretório especificado:
  - `rm -r <diretório>`

```
lucas@lucas:~/Área de Trabalho$ ls
CompPet  'Curso Linux'
lucas@lucas:~/Área de Trabalho$ rm -r CompPet
lucas@lucas:~/Área de Trabalho$ ls
'Curso Linux'
```



# Comandos Básicos

- Copiar <arquivo 1> para o <arquivo 2>:
  - cp <arq1> <arq2>

```
lucas@lucas:~/Área de Trabalho$ cat comandos_avançados.txt
Lista de comandos avançados para Linux!
lucas@lucas:~/Área de Trabalho$ cat comandos_atualizados.txt
Atualização de comandos para o Linux.
```

```
lucas@lucas:~/Área de Trabalho$ cp comandos_atualizados.txt comandos_avançados.txt

lucas@lucas:~/Área de Trabalho$ cat comandos_avançados.txt
Atualização de comandos para o Linux.
lucas@lucas:~/Área de Trabalho$
```

# Comandos Básicos

- Copiar <diretório 1> para <diretório 2>:
  - `cp -r <dir1> <dir2>`
    - Obs: Caso o <diretório 2> não exista, ele é criado!

```
lucas@lucas:~/Área de Trabalho$ ls
CompPet  'Curso Linux'
```

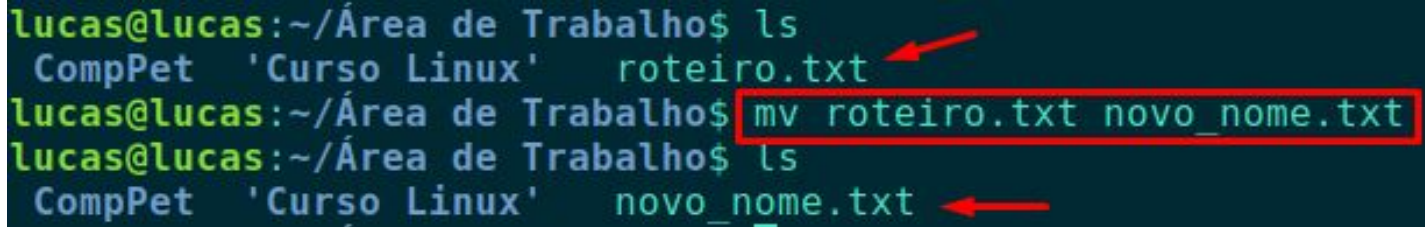
```
lucas@lucas:~/Área de Trabalho$ cd CompPet
lucas@lucas:~/Área de Trabalho/CompPet$ ls
Planejamento
```

```
lucas@lucas:~/Área de Trabalho$ cp -r "Curso Linux" CompPet
lucas@lucas:~/Área de Trabalho$ cd CompPet
lucas@lucas:~/Área de Trabalho/CompPet$ ls
'Curso Linux'  Planejamento
```

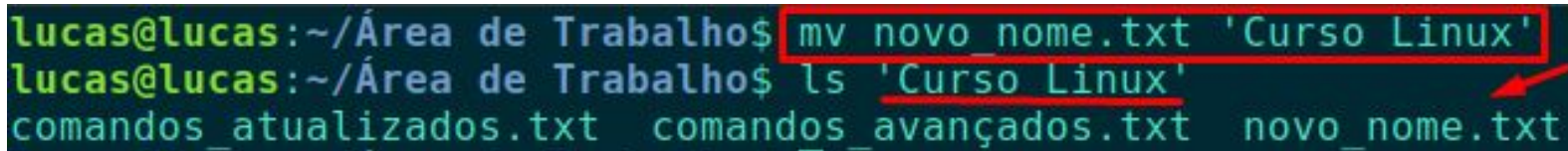
# Comandos Básicos

- Mover <arquivo 1> para <arquivo 2>:
  - `mv <arq1> <arq2>`
    - Se o <arq2> for um diretório, o <arq1> será movido para dentro dele;
    - Também é utilizado para **renomear** o arquivo;

```
lucas@lucas:~/Área de Trabalho$ ls
CompPet  'Curso Linux'  roteiro.txt
lucas@lucas:~/Área de Trabalho$ mv roteiro.txt novo_nome.txt
lucas@lucas:~/Área de Trabalho$ ls
CompPet  'Curso Linux'  novo_nome.txt
```



```
lucas@lucas:~/Área de Trabalho$ mv novo_nome.txt 'Curso Linux'
lucas@lucas:~/Área de Trabalho$ ls 'Curso Linux'
comandos_atualizados.txt  comandos_avançados.txt  novo_nome.txt
```





# Comandos Básicos

- Criar arquivo:
  - `touch <nome_arquivo>`

```
lucas@lucas:~/Documentos$ touch lista_chamada.txt
lucas@lucas:~/Documentos$ ls
lista_chamada.txt
```

- Mostrar o conteúdo de um arquivo binário ou texto:
  - `cat <arquivo>` ou `more <arquivo>`

```
lucas@lucas:~/Documentos$ cat lista_chamada.txt
Higor Sousa
Lucas Guimarães
Marcelo Mendonça
```

# Comandos Básicos

- Limpar a tela do terminal:
  - `clear`
  - `Crtl + I`
- Ver/Modificar data e hora do sistema:
  - `date`
- Mostrar o espaço livre/ocupado de cada partição:
  - `df <opções>`
    - Ex: `df -h`



# Comandos Básicos

- Exibir todos os processos rodando em tempo real:
  - `top`
- Matar processo específico:
  - `kill <pid_processo>`
  - `killall <nome_processo>`
- Procurar pasta ou arquivo:
  - `find <diretório> <opções>`

```
lucas@lucas:~$ find ./Documentos -name lista chamada.txt  
./Documentos/lista_chamada.txt ←
```

# Comandos Básicos

- Logar com superusuário (root):
  - `sudo su`

```
lucas@lucas:~$ sudo su  
[sudo] senha para lucas:  
root@lucas:/home/lucas#
```

- Trocar usuário:
  - `su <usuário>`

```
root@lucas:/home/lucas# su lucas  
lucas@lucas:~$
```

# Comandos Básicos

- Desligar o computador imediatamente ou após tempo determinado:
  - shutdown <opções>
    - Ex.: Desligar imediatamente: shutdown -h now;
    - Ex.: Desligar após 5 minutos: shutdown -h 5.
- Reiniciar o computador imediatamente ou após tempo determinado:
  - reboot
  - shutdown -r
    - Ex.: Reiniciar imediatamente: shutdown -r now;
    - Ex.: Reiniciar após 5 minutos: shutdown -r 5.

# Instalação de Programas

Os programas no Linux podem ser instalados basicamente de duas formas:

- Através do Código Fonte;
- Através de Pacotes;
- Através de Interface Gráfica;
  - Ex: Software Ubuntu

# Instalação de Programas

## Código Fonte

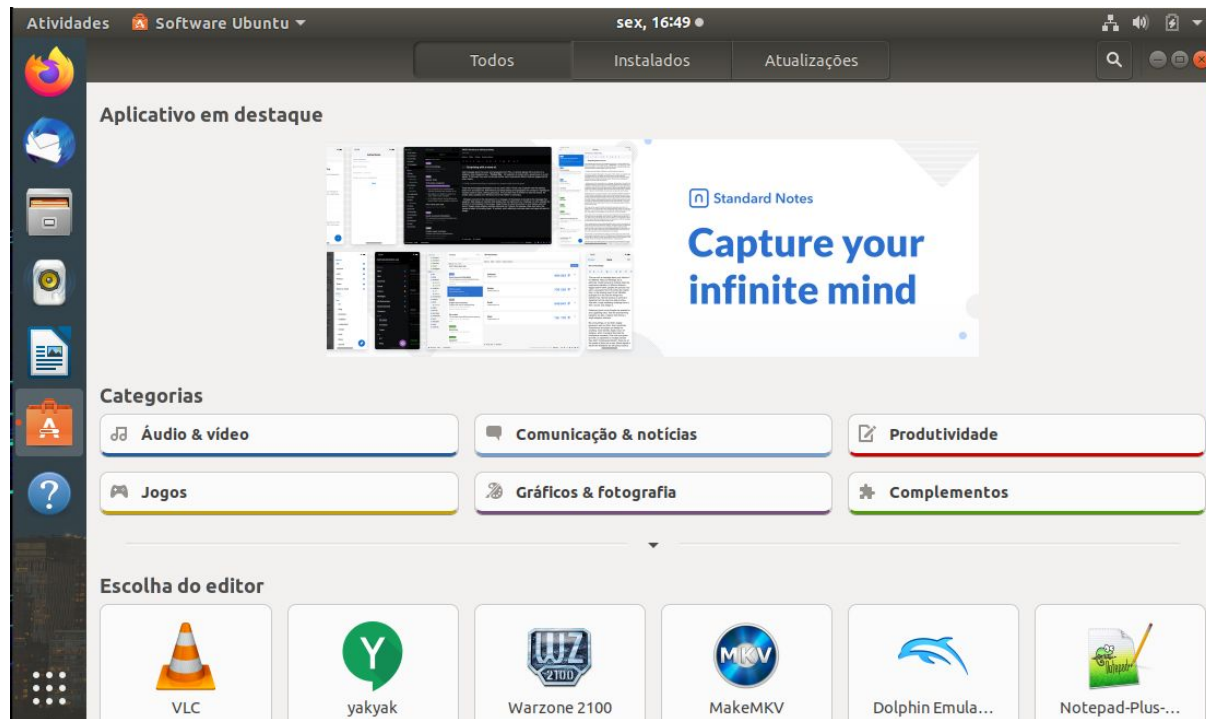
- A instalação consiste na compilação do código fonte do programa, gerando assim o código executável;
- Geralmente acompanha um *script*.

## Pacotes

- O programa já foi compilado e está “empacotado” juntamente com os arquivos necessários para a instalação e devido funcionamento;
- Possível erro: Dependency Hell
  - Acontece quando, na configuração são exigidas atualizações de outros pacotes;

# Instalação de Programas

## Interface Gráfica



# Instalação de Programas

## Tipo de pacote padrão:

- .deb
  - Utilizado pelas distribuições Debian, Ubuntu, entre outras;
- .rpm
  - Utilizado pelas distribuições RedHat, Fedora, CentOS, entre outras;

# Instalação de Programas

## **Gerenciador de Pacotes:**

Software com o objetivo de fornecer uma interface mais amigável para o usuário nas operações de instalação e remoção de pacotes.

- Para pacotes .deb:
  - dpkg
  - apt
- Para pacotes .rpm:
  - yum
  - rpm
  - zypper



# Instalação de Programas

## Comandos:

- Comando para instalação de programas:

```
lucas@lucas:~$ sudo apt-get install nome_programa
```

- Comando para remoção de programas:

```
lucas@lucas:~$ sudo apt-get remove nome_programa
```

- Comando para remover arquivos de programas:

```
lucas@lucas:~$ sudo apt-get purge nome_programa
```

# Instalação de Programas

- O comando apt-get serve também para:
  - Atualizar a lista de pacotes do sistema:

```
lucas@lucas:~$ sudo apt-get update
```

- Atualizar os softwares instalados:

```
lucas@lucas:~$ sudo apt-get upgrade
```

# Instalação de Programas

- Exemplo:
  - Editor de texto por linha de comando:

```
lucas@lucas:~$ sudo apt-get nano
```

- Editor de texto por interface gráfica:

```
lucas@lucas:~$ sudo apt-get gedit
```

# Instalação de Programas

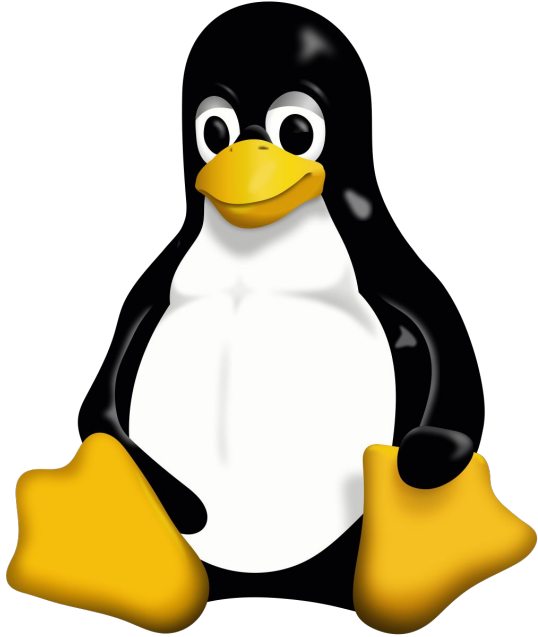
- Caso necessite de um programa que só funcione no Windows, o Linux oferece suporte através do programa chamado Wine;
- O Wine pode ser instalado através do Ubuntu Software Center ou pelo comando:

```
lucas@lucas:~$ sudo apt-get install wine
```



# Exercícios

- 1. Digite o comando para realizar as seguintes tarefas:**
  - a. Entrar, listar e criar/apagar diretórios.
  - b. Criar e Copiar arquivos.
  - c. Visualizar conteúdo de arquivos de texto.
  - d. Mostrar a função de um determinado comando.
  - e. Listar processos e finalizar processos.
  - f. Mostrar uso de recursos do sistema como memória e processamento.
  - g. Mostrar o uso dos discos rígidos.
  - h. Instalar novos softwares/pacotes (Ubuntu).



# Comandos Intermediários

# Redirecionadores de E/S

Operador	Significado
> ou 1>	Redireciona para saída.
>>	Redireciona para fim de arquivo.
<	Redireciona para a entrada.
<<	Redireciona para a entrada e mantém a entrada aberta até que seja digitado algum caractere de EOF (fim de arquivo), como, por exemplo, CTRL + D.
>& ou 2>	Redireciona a saída de erros.
	Redireciona a saída de um comando para a entrada de outro comando.
tee	Redireciona o resultado para a saída padrão e para um arquivo, deve ser usado em conjunto com o " ".

# Redirecionadores de E/S

## Exemplo:

- Abaixo temos três ações: primeiro, o comando cat exibe o conteúdo do arquivo roteiro.txt; segundo, o resultado do comando cat é usado como entrada do comando wc que conta o número de linhas do arquivo; e terceiro, o número de linhas do arquivo roteiro.txt é gravado no arquivo resultado.txt.

```
lucas@lucas:~/Área de Trabalho$ cat roteiro.txt | wc -l > resultado.txt
lucas@lucas:~/Área de Trabalho$ cat resultado.txt
6
```



# Comandos diversos (find)

- Procurar por arquivos/diretórios no disco:
  - find
    - **find [diretório para início da busca] [opções/expressões];**
    - Caso não insira o diretório inicial, começará pelo diretório atual;
    - Exemplo:
      - Procurar o arquivo resultado.txt no diretório atual e em todos os subdiretórios abaixo do diretório atual:

```
lucas@lucas:~$ find -name resultado.txt  
./Área de Trabalho/resultado.txt
```

# Comandos diversos (grep)

- O comando grep procura por um texto dentro de um arquivo ou no dispositivo de entrada padrão:
  - **grep [expressão] [arquivo] [opções]**
- O grep interpreta a [expressão] como uma expressão regular básica;
  - Uma expressão regular é um padrão que descreve um conjunto de caracteres. Uma expressão regular cria uma regra para ser interpretada na busca.

# Comandos diversos (grep)

- **Exemplo:**

- Achar todas as linhas do arquivo roteiro.txt que possuem o caracter 'a';

```
lucas@lucas:~/Área de Trabalho$ grep a roteiro.txt
Roteiro para curso Linux!
História
Terminal
Comandos Básicos
Comandos Intermediários
Tarefas
```

# Comandos diversos (adduser / addgroup)

- Adicionar um usuário no sistema:
  - adduser
    - Por padrão, quando um novo usuário é adicionado, é criado um grupo com o mesmo nome do usuário. Opcionalmente o adduser também pode ser usado para adicionar um usuário a um grupo.

```
lucas@lucas:~$ sudo adduser marcelo
```

- Adicionar um novo grupo de usuários no sistema:
  - addgroup

```
lucas@lucas:~$ sudo addgroup compnet
```

# Comandos diversos (chmod)

- O comando chmod muda a permissão de acesso a um arquivo ou diretório:
  - **chmod [opções] [permissões] [diretório/arquivo]**
    - [opções]:
      - u: para usuário atual
      - g: para o grupo
      - o: para “outros”
      - a: para todos
    - [permissões]:
      - r: leitura
      - w: gravação
      - x: execução

# Comandos diversos (wget)

- O comando wget permite baixar arquivos usando os protocolos HTTP, HTTPS e FTP
  - **wget [opções] [url]**
- Exemplo
  - Baixar o html da página inicial do site do CompPet:

```
lucas@lucas:~$ wget https://www.comppet.ufu.br/
```

# Compactar e Descompactar Arquivos

- Comando tar:
- Compactação
  - tar [opções] [nome do arquivo compactado] [arquivo ou diretório a ser compactado]
  - Exemplo:
    - tar -czf arquivo.tar.gz codes/\*
- Descompactação:
  - tar [opções] [arquivo a ser descompactado]
  - Exemplo:
    - tar -xzf arquivo.tar.gz

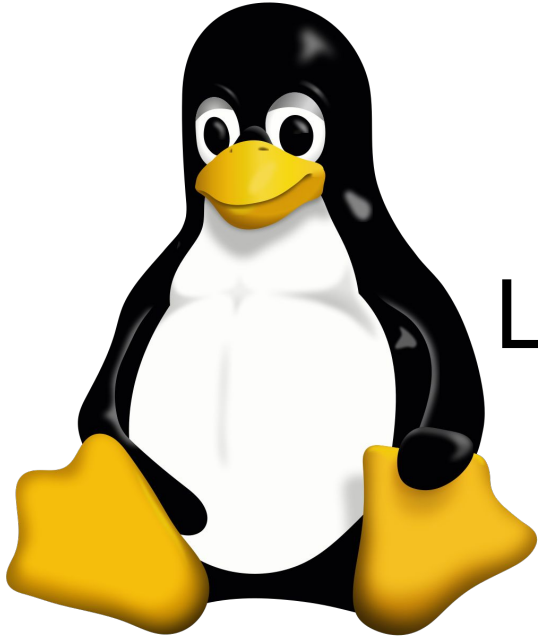
# Conexão Remota

- Comando ssh:
  - `ssh user@ip -p[numero]`
  - Exemplo:
    - `ssh bandit0@bandit.labs.overthewire.org -p2220`
    - password: bandit0
- <https://overthewire.org/wargames/>



# Criptografia e Descriptografia

- Comando openssl
- Criptografia
  - openssl aes-256-cbc -in [arquivo\_entrada] -out [arquivo\_saida]
  - Exemplo:
    - openssl aes-256-cbc -in teste.py -out encrypt.enc
- Descriptografia
  - openssl aes-256-cbc -in [arquivo\_entrada] -out [arquivo\_saida] -d
  - Exemplo:
    - openssl aes-256-cbc -in encrypt.enc -out decrypt.txt -d



# Linguagens de Programação

# Compilação e Execução pelo Terminal

## Linguagem C

- Compilar programa em C:
  - `gcc programa.c -o programa`
- Executar programa em C:
  - `./programa`
- Normalmente compilador GCC já vem instalado
- É preciso alterar as permissões para executar o programa

# Compilação e Execução pelo Terminal

## Linguagem Java

- Compilação em Java:
  - `javac Classe.java`
- Execução em Java:
  - `java Classe.class`
- Necessário instalar Java JDK e JRE
  - `sudo apt install default-jdk`
  - `sudo apt install default-jre`

# Execução pelo Terminal

## Linguagem Python

- Normalmente não se compila python
- Ubuntu já vem com uma versão instalada do python
- Execução de um programa em python
  - `python3 programa.py`

# Exercícios

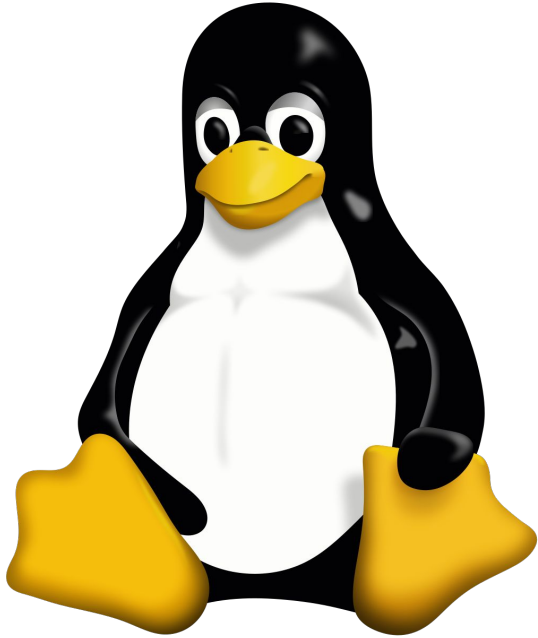
## OverTheWire

- Conjunto de desafios Bandit
- <https://overthewire.org/wargames/bandit/>
- Níveis de 1 a 6
- Lembrem-se de usar comandos man, --help, e o famoso Google para acharem as senhas

# Exercícios

## OverTheWire - Solução

- Level 0-1: ls, cat
- Level 1-2: cat ./-
- Level 2-3: cat “spaces in a filename”
- Level 3-4: ls -a, cat ./hidden
- Level 4-5: for x in {0..9}; do file ./-file0\$x; done
- Level 5-6: find -type f -size 1033c ! -executable



# Shell Script



# Criar um arquivo Shell Script e Interpretador

- Arquivo com extensão .sh
  - `touch script.sh`
- Mudar permissões para poder executá-lo
  - `chmod +rwx script.sh`
- Shell Script possui vários interpretadores, para descobrir o da sua máquina digite
  - `printenv SHELL`
- Definimos o interpretador do script na primeira linha
  - `#!/bin/bash`

# Impressão na Tela e Comandos

- Podemos imprimir no terminal uma mensagem
  - echo “Mensagem”
- Podemos realizar comandos diversos do terminal utilizando o script
  - echo “Username: ”
  - whoami
  - echo “Diretorio do Script: “
  - pwd
- Para sair do programa usamos o comando exit

# Exemplo

- Numerar os arquivos em um diretório:

```
1  #!/bin/bash
2
3  echo "Quantidade de Arquivos"
4
5  x=$(ls |wc -l)
6
7  echo "o numero de arquivos nesse diretorio eh $x"
8
9  sleep 6
10
11  exit
12
13
```

# Comentários e Variáveis

- Comentários podem ser feitos utilizando a marcação #
  - #Isto eh um comentario
- Shell Script não é tipado então pode-se utilizar qualquer valor para uma variável
- A sintaxe é:
  - nome\_da\_variavel=valor
  - nome\_da\_variavel é uma sequência de caracteres, com maiúsculas e minúsculas e \_
  - valor é o qualquer dado que será associado à variavel

# Referenciar variáveis no Shell Script

- Deve-se utilizar o caractere \$ para fazer uma referência a uma variável declarada
  - nome="Marcelo"
  - echo "Meu nome eh \$nome"
- Pode-se ignorar um caractere utilizando \
  - nome="Marcelo"
  - echo "Meu nome eh \ \$nome"
  - Vai mostrar o nome da variável em vez de seu conteúdo

# Saída de um comando

- Podemos associar a saída de um comando a uma variável
- Utilizamos umas das seguintes sintaxes:
  - `variavel=$(comando)`
  - `variavel=`comando``
- Exemplo:
  - `system_info=$(df -h)`
  - `system_info=`df -h``
  - `echo "$system_info"`

# Pegando a Entrada do Usuário

- Para receber uma entrada do usuário usamos o comando read, junto com uma variável:
  - echo “Digite sua idade: “
  - read idade;
  - echo “\$idade”

# Expressões Condicionais

- Para o if temos 2 formas:
- Primeira forma (mais utilizada)  
if [ condição ];  
then  
    ação  
fi
- Segunda forma (menos utilizada)

```
if test condição;  
then  
    ação  
fi
```



# Argumentos para condições

- **n** string1: o comprimento de string1 é diferente de 0;
- **z** string1: o comprimento de string1 é zero;
- string1 **=** string2: string1 e string2 são idênticas;
- string1 **!=** string2: string1 e string2 são diferentes;
- inteiro1 **-eq** inteiro2: inteiro1 possui o mesmo valor que inteiro2;
- inteiro1 **-ne** inteiro2: inteiro1 não possui o mesmo valor que inteiro2;
- inteiro1 **-gt** inteiro2: inteiro1 é maior que inteiro2;
- inteiro1 **-ge** inteiro2: inteiro1 é maior ou igual a inteiro2;
- inteiro1 **-lt** inteiro2: inteiro1 é menor que inteiro2;
- inteiro1 **-le** inteiro2: inteiro1 é menor ou igual a inteiro2;

# Else e Elif

- Para o comando Else

```
if [ condição ];  
then  
    ação  
else  
    ação  
fi
```

- Para o comando Elif

```
if [ condição ];  
then  
    ação  
elif [ condição ];  
then  
    ação  
fi
```

# Comando For

- Utilizando o for com um intervalo de valores entre chaves e separados por ..

```
for VARIABEL in {VALOR1..VALORN}  
do  
    ações  
done
```

- Utilizando for usando o comando do linux seq para geração de sequências
  - Sintaxe: seq 1 100, para sequência de 1 a 100
  - Sintaxe: seq 1 2 100, para sequência de 1 a 100, pulando de 2 em 2

```
for VARIABEL in $(seq 1 100)  
do  
    ações  
done
```

# Comando While e Atribuição

- Sintaxe para o while

```
while [ condição ]
```

```
do
```

```
ações
```

```
done
```

- Atribuição é feita da seguinte forma ((var=valor)), por exemplo:

```
i = 0
```

```
while [ i -lt 10]
```

```
do
```

```
    echo $i
```

```
    ((i=$i+1))
```

```
done
```

# Criação de Funções

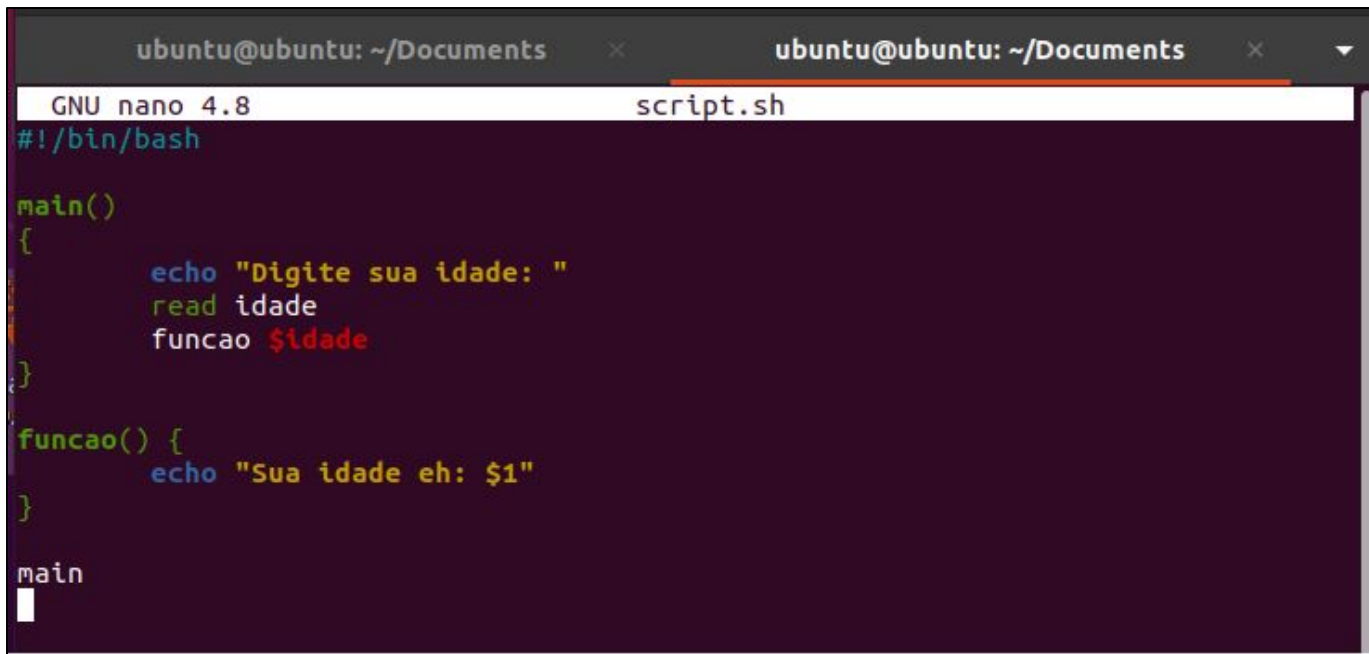
- Sintaxe de uma função

```
nome_funcao()  
{  
    corpo da função  
}
```
- Chamada de uma função é feita igual qualquer outro comando no shell script somente escrevendo o nome da função
- Funções serão executadas somente quando chamadas no shell script
- Podemos criar uma função main, caso queiramos que fique semelhante à linguagem C por exemplo

# Passagem de Parâmetros

- A sintaxe para passagem de parâmetros é:
  - comando PARAMETRO1 PARAMETRO2
- Acesso aos parâmetros passados são feitos, dentro do corpo da função utilizando \$# onde # representa o índice dos argumentos.
- O índice vai de \$0 que é o nome do script sendo executado, \$1 ... \$n, para n argumentos.
- \$# pode ser utilizado para recuperar o número de parâmetros recebidos, ignorando o \$0
- \$\* retorna todos os argumentos de uma vez

# Exemplo



```
ubuntu@ubuntu: ~/Documents  x  ubuntu@ubuntu: ~/Documents  x  ▾
GNU nano 4.8                script.sh
#!/bin/bash

main()
{
    echo "Digite sua idade: "
    read idade
    funcao $idade
}

funcao() {
    echo "Sua idade eh: $1"
}

main
█
```

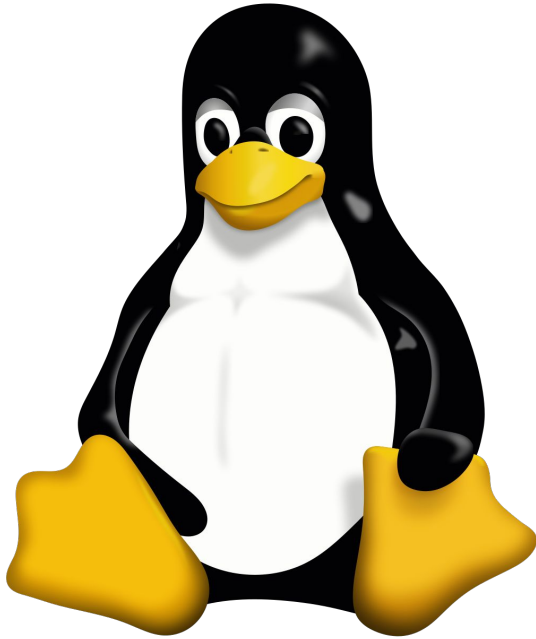
```
ubuntu@ubuntu:~/Documents$ ./script.sh
Digite sua idade:
12
Sua idade eh: 12
```

Deu certo agr hsauhsuahushaa

# Exercícios

- Criar uma pasta chamada Arquivos e entrar nela
- Criar 10 arquivos txt na pasta com nome aleatório
- renomear #\_teste.txt
- Compactar os arquivos
- Criar uma nova pasta fora da pasta Arquivos, chamada Arquivos 2
- Move o arquivo compactado para dentro de Arquivos 2
- Descompactar o arquivo compactado





OBRIGADO PELA  
PARTICIPAÇÃO!!!

Referência para shell script:

<https://www.devmedia.com.br/introducao-ao-shell-script-no-linux/25778>