



# Sistemas Linux

Fernando Demarchi Natividade Luiz



# Roteiro

1. Apresentação
2. História
3. Sistema Operacional
4. Arquitetura de um Sistema Linux
5. Laboratório (hands-on)



# 1

**Quem somos?**



## O que é o Celtab?

O CELTAB executa projetos de pesquisa aplicada nas diversas áreas e disciplinas inerentes ao tema tecnologias livres, promovendo a transferência de tecnologia e difusão do conhecimento por meio de parcerias com instituições públicas e privadas, acadêmicas e de pesquisa, de fomento e de produção, que contribuam para o desenvolvimento de soluções inovadoras que atendam à ITAIPU, as partes interessadas, e reforcem o desenvolvimento socioeconômico e tecnológico da região.



# 2

## HISTÓRIA



# 1991

- Escrito por Linus Torvals em 1991;
- Departamento de Ciência da Computação da Universidade de Helsinki, Finlândia;
- Inspirado no Minix, desenvolvido por Andrew S. Tanenbaum;
- Atende às normas POSIX (Portable Operating System Interface).



# Linus Torvals





# Minix

```
Executing in 32-bit protected mode.
```

```
Building process table: pm fs rs ds tty mem log init.
```

```
Physical memory: total 203060 KB, system 5700 KB, free 197360 KB.
```

```
PCI: video memory for device at 0.15.0: 134217728 bytes
```

```
Root device name is /dev/c0d0p0s0
```

```
AT-D0: multiword DMA modes supported: 0 1 2
```

```
AT-D0: Ultra DMA modes supported: 0 1 2
```

```
AT-D0: Ultra DMA mode selected: 2
```

```
Replacing root
```

```
Multiuser startup in progress ...: is cm0s.
```

```
/dev/c0d0p0s2 is read-write mounted on /usr
```

```
/dev/c0d0p0s1 is read-write mounted on /home
```

```
Starting services: random lance inet printer.
```

```
Starting daemons: update cron syslogd.
```

```
Starting networking: dhc0pd nonamed.
```

```
Alarm call
```

```
Unable to obtain an IP address.
```

```
Local packages (start): done.
```

```
/dev/rescue is read-write mounted on /boot/rescue
```

```
Minix Release 3 Version 1.2a (console)
```

```
145-116-229-112.uilenstede.casema.nl login: _
```

O Minix é um sistema operacional Unix-LIKE, criado por Andrew Tanenbaum, em 1987.

O principal objetivo do sistema era auxiliar no ensino da computação.





# Unix

Multics  
1960

Unix  
1969

**Bell  
Labs**



Ken Thompson e Dennis Ritchie

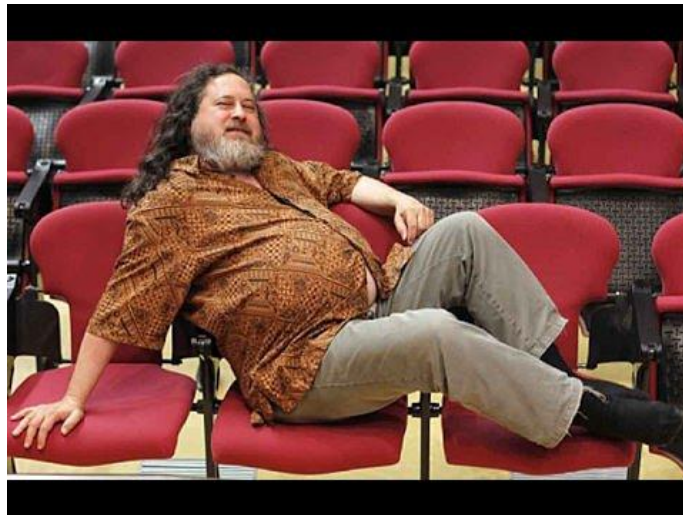


# 1991

- Linus Torvalds continua no processo de desenvolvimento do Linux;
- Disponibilização do código-fonte para a comunidade;
- Desenvolvimento de suporte à outras plataformas.



# GNU - GNU is Not Unix

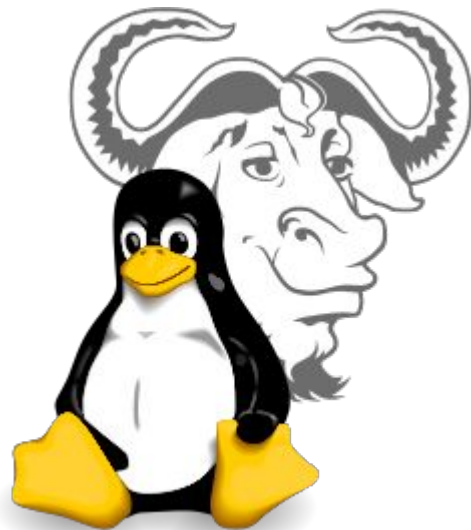


Richard Stallman



# GNU/Linux

- Linus Torvalds desenvolveu o kernel do Linux enquanto estudava na Universidade de Helsinki em 1991;
- No ano passado, 75% do código criado para o Linux foi desenvolvido por programadores que trabalham em empresas privadas;
- Em dezembro de 2009 a IBM anunciou um novo sistema Mainframe desenhado para trabalhar com Linux;
- Os sistemas baseados em Linux encontram-se em 446 dos 500 supercomputadores mais potentes do mundo;
- 95% dos servidores que se utilizam nos estúdios de Hollywood para os filmes de animação rodam Linux.





## Principais Distribuições



redhat®



CentOS



fedora™



debian



ubuntu



SUSE®



openSUSE™

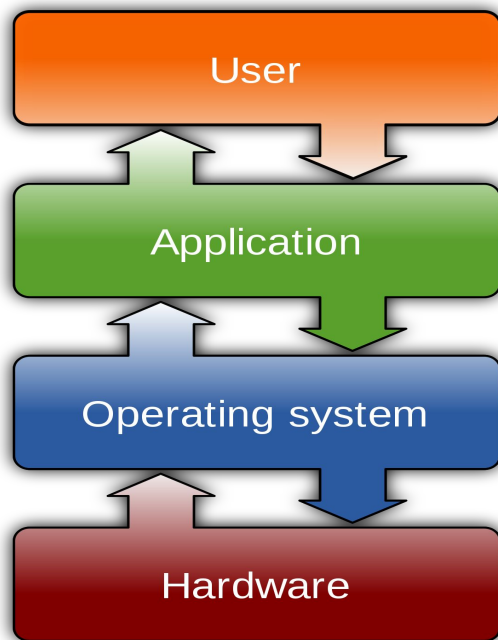


# 3

## O que é um Sistema Operacional



“ *Um sistema operacional (SO) é um software que gerencia os recursos de hardware e software do computador, fornecendo uma interface entre o computador e o usuário.*







# 4

## Arquitetura de um sistema Linux



# Componentes





“O kernel é o núcleo do sistema operacional e se encarrega de executar todas as funções básicas e necessárias para o funcionamento correto do sistema.



# Diagrama do Kernel Linux

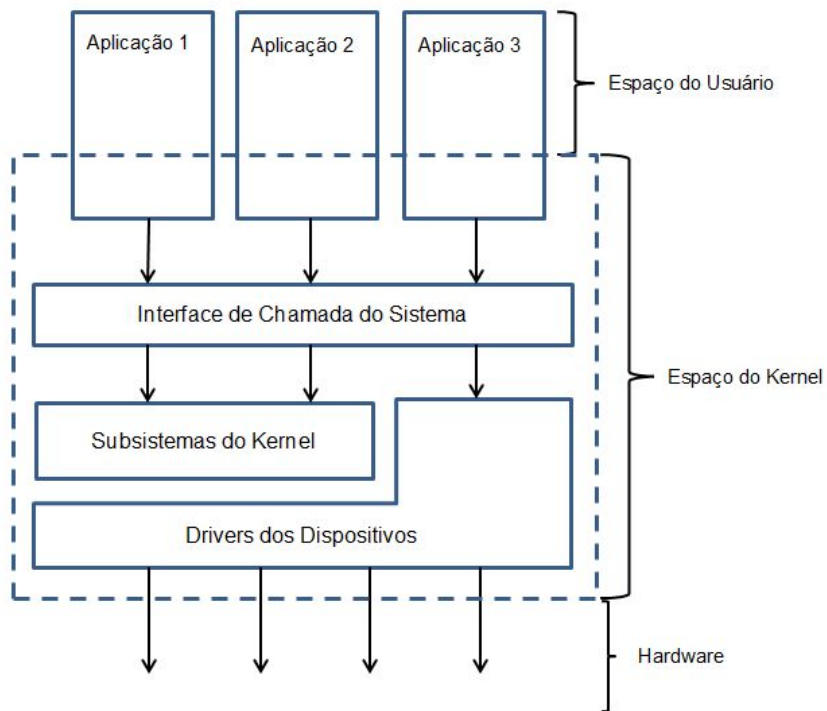


Figura 1 – Relação entre as aplicações, o kernel e o hardware.



## Kernel - Principais Funções

- Detecção de Hardware;
- Gerenciamento de entrada e saída;
- Manutenção do sistema de arquivos;
- Gerenciamento de memória e swapping;
- Controle da fila de processos.



## Bibliotecas de funções padrão

- Funções responsáveis por realizar a comunicação entre as aplicações e o núcleo do Sistema Operacional;
- Uso de funções padrões, tais como:
  - ▷ open;
  - ▷ close;
  - ▷ read;
  - ▷ write.

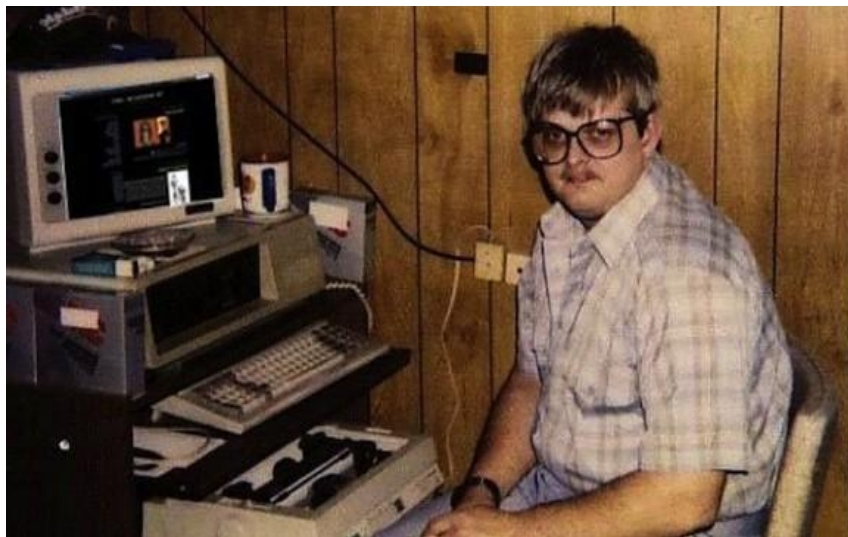


- Processo responsável por ler os comandos de entrada de um terminal;
- Cria novos processos à medida que são requisitados;
- Permite ao usuário trocar de interpretador (Shell) durante a sessão, tais como:
  - ▷ Bash; Sh; Dash; Fish; tcsh.



# Aplicações e Usuários

- Programas com os quais os usuários interagem:
  - ▷ Editores de texto;
  - ▷ Planilhas;
  - ▷ Compiladores.





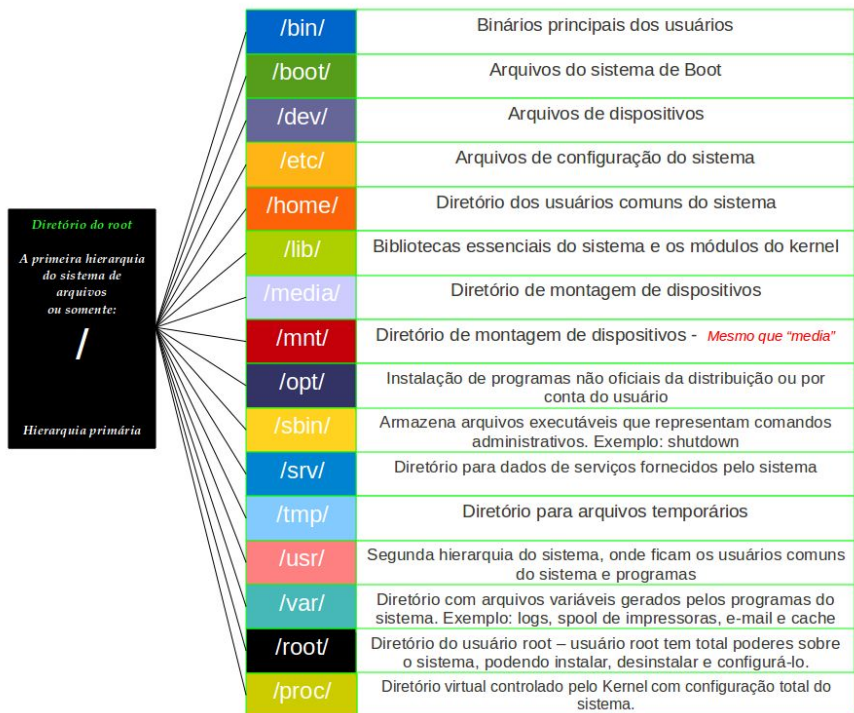


“

*O Fllesystem Hierarchy Standard define os principais diretórios, e o seu conteúdo, em um sistema operacional Linux ou do tipo Unix.*



# Filesystem Hierarchy Standard





# 5

## Laboratório: Comandos básicos



# Cheat sheet

Comando	Ação
cd	Entra em um novo diretório
pwd	Lista o diretório atual do usuário
ls	Lista arquivos e diretórios
date	Mostra a data atual
mkdir	Cria um diretório
clear	Limpa a tela
touch	Cria um arquivo vazio
cp	Copia um arquivo ou diretório
mv	Move um arquivo ou diretório

Comando	Ação
shutdown -h now (sudo)	Desliga o computador
reboot (sudo)	Reinicia o computador
cat	Exibe o conteúdo de um arquivo
more / less	Exibe o conteúdo de um arquivo de forma navegável
sudo	Executa um comando como root (super usuário)
man	Exibe a função de um determinado comando



# Buscando informações sobre o Sistema

Comando	Ação
<code>cat /etc/*-release</code>	Informações gerais sobre a distribuição utilizada
<code>lscpu</code>	Lista informações sobre o processador
<code>lspci</code>	Lista dispositivos PCI
<code>lsusb</code>	Lista dispositivos USB
<code>free - (b k m g)</code>	Lista informações sobre a memória
<code>uname -a / uname -r</code>	Lista informações sobre o kernel e arquitetura do sistema



## Laboratório: Desafio 01

[https://github.com/nativanando/linux-course/blob/master/desafio\\_01.md](https://github.com/nativanando/linux-course/blob/master/desafio_01.md)



# Sistemas Linux - Parte 2

Fernando Demarchi Natividade Luiz



# Roteiro

1. Gerenciadores de pacotes;
2. Sistema de arquivos:
  - a. Tipos de arquivos;
  - b. Permissões de arquivos.
3. Laboratório (hands-on).





## O que é um pacote?

É um arquivo, compactado, que contém todos os arquivos (binários, shell scripts, documentação, arquivos de configuração...) necessários para a instalação da aplicação.

No mundo Linux, os principais formatos de empacotamento são:

- .deb;
- .rpm.



## O que é um gerenciador de pacotes?

Um gerenciador de pacotes é um software, existente em uma distribuição Linux, cuja função é facilitar a instalação, remoção, configuração e manutenção dos pacotes.

No mundo Linux existem vários gerenciadores de pacotes, sendo que os principais são o DPKG (.deb) e o RPM (.rpm).



# APT e YUM

Utilitários desenvolvidos para facilitar a instalação, remoção e atualização de softwares em suas respectivas distribuições.

## YUM

Utilizado para administrar pacotes .rpm

## APT

Utilizado para administrar pacotes .deb



# Principais comandos: APT e YUM

Ação	APT	YUM
Instalar um novo pacote	apt-get install (-y) "pacote"	yum install (-y) "pacote"
"Reinstalar" um pacote	apt-get install --reinstall "pacote"	yum reinstall "pacote"
Remover um pacote	apt-get remove "pacote"	yum remove "pacote"
Atualizar a lista de repositórios	apt-get update	yum update
Atualiza a lista de repositórios e pacotes já instalados no sistema	apt-get upgrade	yum upgrade
Informações de um pacote	apt-cache show "pacote"	yum info "pacote"
Busca um determinado pacote na lista de repositórios	apt-cache search "pacote"	yum search "pacote"



## Estudo de caso: VIM

- Verificar se o pacote está disponível para instalação;
- Verificar informações sobre o pacote;
- Realizar a atualização da lista de repositórios;
- Realizar a instalação do pacote;
- Remover o pacote.



## Estudo de caso: git

- Verificar se o pacote está disponível para instalação;
- Verificar informações sobre o pacote;
- Realizar a atualização da lista de repositórios;
- Realizar a instalação do pacote;
- Remover o pacote.



“Em qualquer SO, é necessário armazenar dados em arquivos e organizá-los em diretórios. Essas ações são responsabilidades de um componente chamado Sistema de arquivos.



# Sistema de arquivos Linux

Bloco de boot

Super bloco

Tabela de blocos

Inodes

Bloco de dados

Instruções para carregar o SO.

Parâmetros do File System.

Gerencia o espaço livre no file system.

Estrutura de dados dos arquivos

Armazena o conteúdo dos arquivos





# Inode

```
shum@sol:~$ ls -l
total 20
drwx----- 2 shum  staff  4096 Jan 16 22:04 Mail
drwx----- 3 shum  staff  4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum  staff  4096 Jan 13 16:42 public
drwxr-xr-x  2 shum  staff  4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum  staff   628 Jan 15 20:04 verse
```

Diagram illustrating the components of the `ls -l` output:

- file type**: Indicated by the first character of the permissions (e.g., `d` for directory, `-` for regular file).
- permissions**: Indicated by the next nine characters (e.g., `rw-r--r--`).
- number of hard links**: Indicated by the number following the permissions (e.g., `1`).
- user (owner) name**: Indicated by the user name (e.g., `shum`).
- group name**: Indicated by the group name (e.g., `staff`).
- size**: Indicated by the file size in bytes (e.g., `628`).
- date/time last modified**: Indicated by the date and time (e.g., `Jan 15 20:04`).
- filename**: Indicated by the file name (e.g., `verse`).

Legend for permissions:

- rwx**: permissions for the user (owner)
- group permissions**: permissions for the group
- other (everyone) permissions**: permissions for others
- executable**: permission to execute the file
- writable**: permission to write to the file
- readable**: permission to read the file

Todo sistema de arquivos possui uma tabela de inodes e cada arquivo possui um inode associado a ele.



# Tipos de arquivos

Tipo do arquivo	Simbolo	Definição e exemplos
<b>Arquivo regular</b>	-	Executável, texto, imagem.
<b>Diretório</b>	d	diretórios -
<b>Arquivos de dispositivos</b>	c	I/O, portas seriais, discos rígidos.
<b>Socket</b>	s	Comunicação bilateral entre processos, ex: datagram socket (UDP), stream socket (TCP).
<b>Named pipe</b>	p	Permite a comunicação entre dois processos em uma mesma máquina.
<b>Symbolic link</b>	l	Ponteiro para um arquivo existente



# Permissões de arquivos

- Nove bits controlam quem poder ler, escrever e executar um arquivo.
- O linux define permissões em três camadas:
  - ▷ Dono do arquivo;
  - ▷ Grupo ao qual o arquivo pertence;
  - ▷ Outros usuários do sistema.



# Permissões de arquivos

```
shum@sol:~$ ls -l
total 20
drwx----- 2 shum staff 4096 Jan 16 22:04 Mail
drwx----- 3 shum staff 4096 Jan 16 14:15 csc128
drwxr-xr-x 2 shum staff 4096 Jan 13 16:42 public
drwxr-xr-x 2 shum staff 4096 Jan 16 14:07 public_html
-rw-r--r-- 1 shum staff 628 Jan 15 20:04 verse
```

Diagram illustrating the components of the `ls -l` output:

- file type**: Indicated by the first character of the permission string (e.g., `d` for directory, `-` for regular file).
- user (owner) permissions**: Indicated by the next three characters (e.g., `rw` for read and write).
- group permissions**: Indicated by the next three characters (e.g., `x` for execute).
- other (everyone) permissions**: Indicated by the last three characters (e.g., `-` for no permissions).
- number of hard links**: The number before the owner name (e.g., `2`).
- user (owner) name**: The name after the number of hard links (e.g., `shum`).
- group name**: The name after the user name (e.g., `staff`).
- size**: The file size in bytes (e.g., `4096`).
- date/time last modified**: The date and time the file was last modified (e.g., `Jan 16 22:04`).
- filename**: The name of the file (e.g., `Mail`).

Legend for permissions:

- r**: readable
- w**: writeable
- x**: executable

Permissão	Binário	Decimal
---	000	0
--X	001	1
-W-	010	2
-WX	011	3
r--	100	4
r-X	101	5
rW-	110	6
rWX	111	7



# Comandos importantes

Comando	Ação
<b>chown &lt;novodono&gt;:&lt;novogrup&gt; &lt;arquivo&gt;</b>	Altera o dono e o grupo de um determinado arquivo
<b>chmod "permissões" arquivo</b>	Altera a permissão de um determinado arquivo



## Laboratório: Desafio 02

[https://github.com/nativanando/linux-course/blob/master/desafio\\_02.md](https://github.com/nativanando/linux-course/blob/master/desafio_02.md)



# Grupos e usuários



## Grupos

A criação de grupos de usuários geralmente é feita para controlar o acesso a arquivos ou serviços. Cada grupo possui um nome e um identificador numérico único.

Essas informações podem ser encontradas nos arquivos `/etc/group` - `/etc/gshadow`





# Arquivo /etc/group

```
oracle:x:1000:dba,oinstall,grid
```



Campo	Propósito
1 - Nome	Contém o nome do grupo
2 - Senha (x)	Contém a senha do grupo
3 - GID	Identificador numérico do grupo
4 - Lista de usuários	Lista de usuários pertencentes ao grupo, separados por “,”



# Adicionando e removendo grupos

Comando	Ação
<b>addgroup "grupo"</b>	Adicionar um novo grupo ao sistema
<b>groupdel "grupo"</b>	Exclui um grupo do sistema
<b>groups "usuario"</b>	Grupos que o usuário pertence
<b>addgroup "usuario" "grupo"</b>	Adicionar um usuário a um grupo
<b>deluser "usuario" "grupo"</b>	Excluir um usuário de um grupo



# Usuários

No linux, apenas os usuários cadastrados podem acessar o sistema. Eles são identificados por:

- Um nome e uma senha;
- Um diretório de trabalho;
- Um interpretador de comandos (shell);
- Um identificador único.



# /etc/passwd

```
testuser:x:1481:1482:This is a test user:/home/testuser:/bin/bash
```

Diagram illustrating the fields of the `/etc/passwd` entry for `testuser`:

- [Username] points to `testuser`
- [Password] points to `x`
- [Userid] points to `1481`
- [Groupid] points to `1482`
- [User Information] points to `This is a test user`
- [User home path] points to `/home/testuser`
- [User shell] points to `/bin/bash`



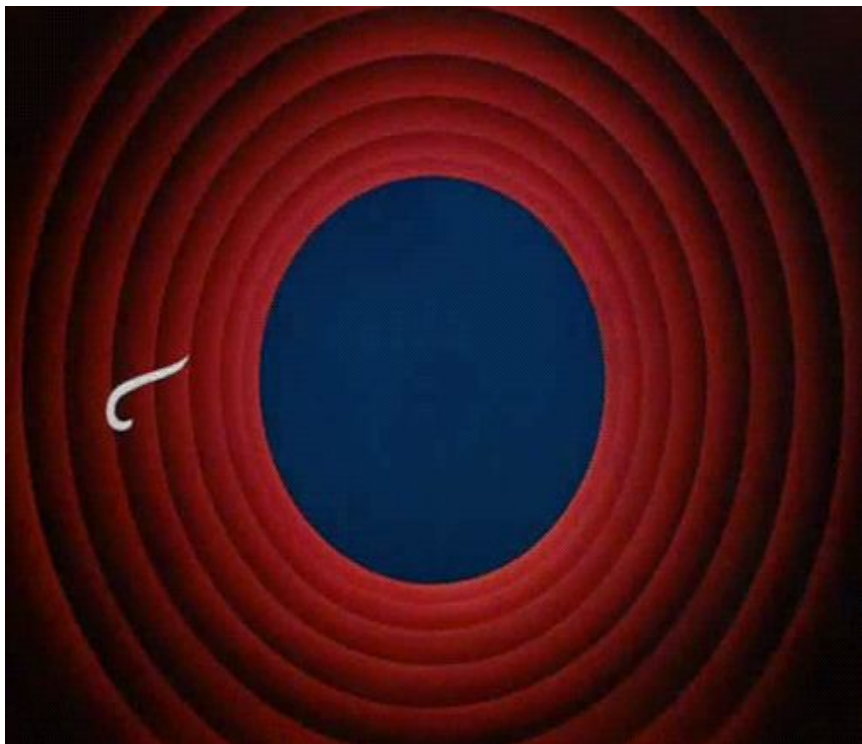
# Comandos para adicionar e remover usuários

Comando	Ação
<b>adduser --home /home/diretorio usuario</b>	Adicionar um novo usuário ao sistema
<b>deluser --remove-home usuario</b>	Remover um usuário do sistema



## Laboratório: Desafio 03

[https://github.com/nativanando/linux-course/blob/master/desafio\\_03.md](https://github.com/nativanando/linux-course/blob/master/desafio_03.md)



[fernando.luiz@pti.org.br](mailto:fernando.luiz@pti.org.br)



[Fernando Natividade](#)



[Material](#)