

**Professor**  
Massaki

# **SISTEMAS OPERACIONAIS ABERTOS: Linux**

## **(Parte 2)**



# Tecclas de atalhos no Linux



# Tecclas de atalhos no Linux



## Ctrl + C

**Interrompe uma execução.** Usada para sair de um comando ou processo em um terminal. Isso interromperá imediatamente um programa em execução. Se você quiser parar de usar um programa em execução no primeiro plano, basta pressionar esta combinação de teclas.

## Ctrl + Z

**Passa a execução para 2º plano.** Este atalho irá enviar um programa em execução em segundo plano. Normalmente, você pode conseguir isso antes de executar o programa usando a opção &, mas se você esqueceu de fazer isso, use esta combinação de teclas.



# Tecclas de atalhos no Linux



## Ctrl + D

**Desconexão.** Este atalho do linux desconecta você do terminal atual. Se você estiver usando uma conexão SSH, ela será fechada. Se você estiver usando um terminal diretamente, o aplicativo será fechado imediatamente.

## Ctrl + L

**Limpa o terminal.** Além de poder escrever C-L-E-A-R, você pode simplesmente usar Ctrl + L para limpar o terminal.

# Tecclas de atalhos no Linux

## Ctrl + A

**Move para início da linha.** Este atalho moverá o cursor para o início da linha. Caso você tenha digitado um longo comando ou caminho no terminal e deseje ir para o início dele, usando a tecla de seta para mover o cursor levará bastante tempo. Observe que você não pode usar o mouse para mover o cursor para o início da linha.

## Ctrl + E

**Move p/ final da linha.** Além de poder escrever C-L-E-A-R, você pode simplesmente usar Ctrl + L para limpar o terminal.



# Tecclas de atalhos no Linux

## Ctrl + U

**Apaga a linha.** Ao digitar um comando errado ao vez de usar o backspace para descartar o comando atual, use o atalho Ctrl + U no terminal do Linux. Este atalho apaga tudo, desde a posição atual do cursor até o início da linha.

## Ctrl + K

**Apaga do cursor em diante.** Este atalho é semelhante ao atalho Ctrl + U. A única diferença é que, em vez do início da linha, ela apaga tudo, da posição atual do cursor até o final da linha.





# Tecclas de atalhos no Linux



## Ctrl + W

**Apaga uma palavra.** Mas e se você só precisa excluir uma única palavra? Use o atalho Ctrl + W. Usando o atalho Ctrl + W, você pode apagar a palavra anterior à posição do cursor. Se o cursor estiver em uma palavra em si, apagará todas as letras da posição do cursor até o início da palavra.

## Ctrl + Y

**Cola de volta.** Isto irá colar o texto apagado que você viu com os atalhos Ctrl + W, Ctrl + U e Ctrl + K. Vem à mão no caso de você ter apagado o texto errado ou se você precisar usar o texto apagado em outro lugar.

# Tecclas de atalhos no Linux

## Ctrl + P

**Comando anterior.** Você pode usar este atalho no linux para visualizar o comando anterior. Você pode pressioná-lo repetidamente para continuar retornando ao histórico de comandos. Em muitos terminais, o mesmo pode ser alcançado com a tecla PgUp.

## Ctrl + N

**Próximo comando.** Você pode usar este atalho em conjugação com Ctrl + P. Ctrl + N exibe o próximo comando. Se você estiver visualizando comandos anteriores com Ctrl + P, poderá usar Ctrl + N para navegar para frente e para trás. Muitos terminais têm esse atalho mapeado para a chave PgDn.





# Tecclas de atalhos no Linux

## Comando “alias”

Você digitou uma combinação de comandos legais que tem que repetir sempre? Use o comando alias para criar seu próprio atalho:

```
aluno@aluno-Virtual-Machine:~$ alias conexoes="netstat -plan|grep :80|awk '{print $5}'|cut -d: -f  
1|sort|uniq -c|sort -nk 1"
```



# Tecclas de atalhos no Linux

## Atalhos básicos

Os atalhos básicos do Linux são aqueles que precisam estar “grudados” na sua mente e na ponta dos seus dedos, literalmente. Afinal, é com eles que as ações básicas e corriqueiras serão executadas, sem você precisar pensar muito para fazê-las.



TECLA(s)	FUNÇÃO
F2	Fazer uma pesquisa de texto rápida
F3	Avançar um resultado de busca dentro do atalho F2
Shift + F3	Voltar um resultado da pesquisa do trecho dentro de F2
F5	Atualizar uma página, assim como no Windows
F7	Abrir o corretor ortográfico do Linux (muito útil se você estiver escrevendo)
CTRL + A	Selecionar todo o texto de uma página
CTRL + B	Deixar o texto selecionado em negrito ( <b>Bold</b> )
CTRL + I	Deixar o texto selecionado em itálico
CTRL + U	Deixar o texto selecionado sublinhado
CTRL + S	Salvar o seu arquivo (útil principalmente em editores de texto)

# Tecclas de atalhos no Linux

## Trabalhar com processos

Para trabalhar com os **processos no Linux**, é muito simples. Basta saber qual atalho funciona para a ação que você deseja executar.

Se deseja interromper, desconectar, limpar ou executar processos em segundo plano, por exemplo, existem comandos específicos:



TECLA(s)	FUNÇÃO
CTRL + C	Interromper um programa
CTRL + Z	Segundo plano
CTRL + D	Desconexão
CTRL + L	Limpa o terminal
CTRL + A	Mover para o início da linha
CTRL + E	Mover para o final da linha
CTRL + U	Apagar a linha
CTRL + K	Apagar do cursor em diante
CTRL + W	Apagar uma palavra
CTRL + Y	Cola de volta



# Tecclas de atalhos no Linux

## Navegação no Terminal

É preciso realizar a navegação no terminal de modo simples e rápido. Afinal, agilidade e simplicidade precisam estar no cotidiano de trabalho das equipes de TI que executam seus processos em Linux. Um modo fácil de fazer isso é combinando atalhos específicos, que possibilitarão a navegação simples. Veja:



TECLA(s)	FUNÇÃO
Super+Tab ou o Alt+Tab	Alternar entre aplicativos
Super+L ou Ctrl+Alt+L	Bloquear a tela
Super+D ou Ctrl+Alt+D	Minimizar janelas abertas
Ctrl+Q ou Ctrl+W.	Fechar todas as janelas
Ctrl+Alt+Del	Encerrar sessão (desligar)
Super+Tab ou o Alt+Tab	Alternar entre aplicativos
Super+L ou Ctrl+Alt+L	Bloquear a tela
Super+D ou Ctrl+Alt+D	Minimizar janelas abertas
Ctrl+Q ou Ctrl+W.	Fechar todas as janelas

# Principais comandos Linux

## useradd

O comando **useradd** serve para criar diferentes tipos de contas (usuário, grupo, sistema...) aplicando algumas configurações (com ou sem diretório /home, diretórios alternativos, data de expiração...) e, ainda, fazer consultas a contas já existentes (visualizar ID, listar grupos e diretórios...).

```
aluno@aluno-Virtual-Machine:~$ sudo useradd -m UsuarioTeste
```

A opção **-m**, conforme a tabela acima, cria um diretório /home para o usuário recém-criado. Por isso, sempre que criamos usuários com esse método, um novo diretório é gerado pelo sistema.

várias configurações da conta podem ser definidas com as opções descritas a seguir:

**-c comentário**

Comentário (geralmente o nome completo do usuário).

**-d diretório**

Caminho para o diretório pessoal do usuário. O padrão é utilizar um diretório com o mesmo nome de login em /home.

**-g grupo**

Grupo principal (GID). Precisa existir previamente no sistema.

**-G grupo1,grupo2**

Grupos adicionais, separados por vírgula.

**-u UID**

UID (user ID) do usuário.

**-s shell**

Shell padrão para o usuário.

**-p senha**

Senha (entre aspas).

**-e data**

Data de validade da conta.

**-k /etc/skel**

Copia os modelos de arquivos e diretórios em /etc/skel para a nova conta.

**-m**

Cria o diretório pessoal em /home, se não existir.

# Principais comandos Linux

## useradd

O comando **useradd** serve para criar diferentes tipos de contas (usuário, grupo, sistema...) aplicando algumas configurações (com ou sem diretório /home, diretórios alternativos, data de expiração...) e, ainda, fazer consultas a contas já existentes (visualizar ID, listar grupos e diretórios...).

```
aluno@aluno-Virtual-Machine:~$ sudo useradd -m UsuarioTeste
```

A opção **-m**, conforme a tabela acima, cria um diretório /home para o usuário recém-criado. Por isso, sempre que criamos usuários com esse método, um novo diretório é gerado pelo sistema.

### Registrar senhas

Para completar o processo é necessário registrar uma senha para cada usuário. Nesse caso, utiliza-se o comando **passwd**, conforme o exemplo a seguir.

```
sudo passwd manoj
```

Assim que executado o comando, o terminal retornará um campo no qual deve ser digitada a senha.



# Principais comandos Linux

## userdel

Para deletar usuários, o comando utilizado é o `userdel`, acompanhado do nome do usuário. Com isso, tanto o usuário como todos os itens atrelados à conta dele são removidos, também.

```
aluno@aluno-Virtual-Machine:~$ sudo useradd -m UsuarioTeste
```

A opção `-r` é usada para remover a conta declarada no fim do comando.

# Principais comandos Linux

## passwd

Para completar o processo é necessário registrar uma senha para cada usuário. Nesse caso, utiliza-se o comando passwd, conforme o exemplo a seguir.

```
aluno@aluno-Virtual-Machine:~$ sudo passwd Maria
```

Assim que executado o comando, o terminal retornará um campo no qual deve ser digitada a senha.

# Principais comandos Linux

## addgroup

Além de criar usuários, o administrador da rede também pode criar grupos para facilitar o gerenciamento quando há grande quantidade de usuário ou alguma segmentação se faz necessária (departamentos, times, divisões etc.), deixando o acesso a arquivos e recursos mais bem organizado. Neste exemplo, vamos criar dois tipos de grupos: o primeiro será composto por usuários com permissões para editar arquivos (denominado “diretores”) e o segundo, por usuários que podem fazer a leitura somente (denominado “operadores”).

Criando os dois grupos:

```
aluno@aluno-Virtual-Machine:~$ addgroup diretores
```

```
aluno@aluno-Virtual-Machine:~$ addgroup operadores
```

Como saber se os grupos foram indexados corretamente pelo sistema? Basta digitar o comando `less /etc/group` e verificar se ambos constam na lista.

Agora, vamos supor que Carlos Silva e Manoel sejam, respectivamente, diretor e assistente do departamento financeiro da empresa. Para adicioná-los em seus devidos grupos, digite:

```
aluno@aluno-Virtual-Machine:~$  
sudo usermod -a -G  
diretores carlos_silva  
sudo usermod -a -G
```



# Principais comandos Linux

## ls

O comando 'ls' no Linux é uma ferramenta poderosa usada para exibir nomes de usuários e grupos junto com seus IDs numéricos (ID do usuário – UID ou ID do grupo – GID) do usuário atual ou de qualquer usuário especificado no sistema. Este comando é particularmente útil para administradores de sistema e usuários que precisam verificar identidades de usuários e permissões associadas.

### 1. Opções Básicas

**ls** → Lista arquivos e diretórios no diretório atual.

**ls -l** → Exibe detalhes como permissões, proprietário, grupo, tamanho e data de modificação.

**ls -a** → Mostra tudo, incluindo os ocultos (arquivos que começam com .).

**ls -h** → Exibe tamanhos de arquivos em formato legível (KB, MB, GB).

### 2. Opções de Ordenação e Formatação

**ls -t** → Ordena os arquivos por data de modificação (mais recente primeiro).

**ls -r** → Inverte a ordem da listagem.

**ls -S** → Ordena os arquivos por tamanho (do maior para o menor).

**ls -X** → Ordena os arquivos por extensão.

**ls -1** → Lista os arquivos em uma única coluna.

### 3. Opções Avançadas

**ls -lh** → Combina -l e -h para exibir detalhes e tamanhos legíveis.

**ls -la** → Combina -l e -a, mostrando todos os arquivos com detalhes.

**ls -d \*/** → Lista apenas diretórios.

**ls -i** → Mostra o número do inode de cada arquivo.

**ls -R** → Lista arquivos e subdiretórios recursivamente.

# **MANIPULAÇÃO DE ARQUIVOS E PASTAS**

# Principais comandos Linux

## cd

O comando **'cd'** permite que os usuários alterem seu diretório de trabalho atual dentro do sistema de arquivos. A sintaxe básica do comando **'cd'** é a seguinte:

```
aluno@aluno-Virtual-Machine:~$ cd [directory]
```

Aqui, substitua **[directory]** pelo caminho do diretório de destino para o qual você deseja navegar. Se nenhum diretório for especificado, **'cd'** redirecionará para seu diretório inicial por padrão. Vamos explorar a funcionalidade do comando por meio de exemplos.





# Principais comandos Linux

## mkdir

O comando 'mkdir' é como uma varinha mágica para criar pastas super facilmente. 'mkdir' significa "criar diretório" e ajuda você a organizar as coisas do seu computador criando pastas com apenas um comando. Não importa se você está criando uma pasta ou várias delas em sequência, 'mkdir' está lá para ajudar você a manter as coisas limpas e organizadas no seu computador. Neste guia, falaremos sobre como usar 'mkdir', quais palavras digitar e alguns truques legais para deixar suas pastas do jeito que você quer no Linux.

Este comando pode criar vários diretórios de uma vez, bem como definir as permissões para os diretórios. É importante notar que o usuário que executa este comando deve ter permissão suficiente para criar um diretório no diretório pai, ou ele/ela pode receber um erro de 'permissão negada'.

```
aluno@aluno-Virtual-Machine:~$ mkdir [opções...] [nome_do_diretório]
```

OPÇÕES	DESCRIÇÃO
-help	Exibe opções de ajuda sobre determinado comando
-version	Exibe o número da versão e informações adicionais sobre a licença para mkdir
-v or verbose	Habilita o modo verbose, exibindo uma mensagem para cada diretório criado. Quando usado com o argumento [directories], ele mostra os nomes dos diretórios que estão sendo criados.
-p	Um sinalizador que permite a criação de diretórios pais conforme necessário.
-m	Define modos de arquivo ou permissões para os diretórios criados. A sintaxe segue a do comando chmod.

SOP Abertos: LINUX

# Principais comandos Linux

## rm

rm significa **remover** here. rm comando é usado para remover objetos como arquivos, diretórios, links simbólicos e assim por diante do sistema de arquivos como o UNIX. Para ser mais preciso, rm remove referências a objetos do sistema de arquivos, onde esses objetos podem ter várias referências (por exemplo, um arquivo com dois nomes diferentes). **Por padrão, ele não remove diretórios.** Este comando normalmente funciona silenciosamente e você deve ter muito cuidado ao executar o comando **rm**, porque depois de excluir os arquivos, você não poderá recuperar o conteúdo dos arquivos e diretórios. **Sintaxe:**

```
aluno@aluno-Virtual-Machine:~$ rm [OPTION]... FILE...
```

Já quando preciso excluir um diretório digito:

```
aluno@aluno-Virtual-Machine:~$ rm -r [DIRETORIO]
```

# Sistema de Arquivos

## Arquivos e Diretórios Escondidos

Para esconder os arquivos e diretórios no Linux, basta renomear o seu nome, iniciando-o com um ponto “.”.

Se renomearmos o diretório musicas, iniciando com um “.”, ele não irá aparecer mais na lista de arquivos e diretórios:

```
$ mv musicas .musicas
```

O comando “mv” serve para mover, ou renomear arquivos e diretórios.  
Ao listar a pasta, o diretório “musicas” não irá aparecer:

```
$ ls  
Arquivo
```

# Principais comandos Linux

## mv

O comando mv no Linux move ou renomeia arquivos e diretórios. Ele não altera os atributos dos arquivos ou diretórios movidos se a transferência for o mesmo sistema de arquivos. Se o destino para onde os arquivos ou diretórios forem movidos não existir, o comando renomeia a origem, senão os dados são gravados por cima. As opções mais frequentes são:

- f: Força a movimentação dos arquivos suprimindo mensagens de confirmação para gravar por cima.
- i: Pergunta ao usuário antes de mover cada arquivo.

### Sintaxe:

```
aluno@aluno-Virtual-Machine:~$ mv [opções] origem destino
```

### Exemplos de uso:

#### 1.Renomear um arquivo:

```
aluno@aluno-Virtual-Machine:~$ mv documento.txt /home/usuário/Documents
```

Renomeia antigo.txt para novo.txt.

#### 2.Mover um arquivo para outra pasta:

```
aluno@aluno-Virtual-Machine:~$ mv antigo.txt novo.txt
```

Move documento.txt para a pasta Documents.



# apt-get

O APT ( *Advanced Packaging Tool* ) é uma interface de usuário de software livre e de código aberto que, em conjunto com bibliotecas de sistema, instala, configura ou remove automaticamente pacotes de software para Debian e distribuições Linux relacionadas. O APT simplifica o processo de gerenciamento de software em sistemas de computador do tipo Unix, desde a compra de pacotes de software até a configuração deles para instalação automatizada, seja de arquivos pré-compilados ou compilando e compilando código-fonte.



# Apt-get

APT é uma coleção de ferramentas de software incluídas em um pacote chamado apt. Grande parte do apt é uma biblioteca C++ e até inclui programas de linha de comando para gerenciar pacotes de software, como: **apt** , **apt-get** , ***apt-cache*** . O pacote apt é incluído em todas as distribuições Debian por padrão. O apt, assim como o **dpkg**, pode ser visto como um front-end como *dselect* versão mais amigável. Enquanto o dpkg lida com operações com pacotes personalizados, as ferramentas apt lidam com relacionamentos pacote a pacote, especialmente dependências, bem como decisões de controle de versão de nível superior (rastreamento de lançamento e captura de pacote).

```
aluno@aluno-Virtual-Machine:~$ apt [options] command [packages]
```

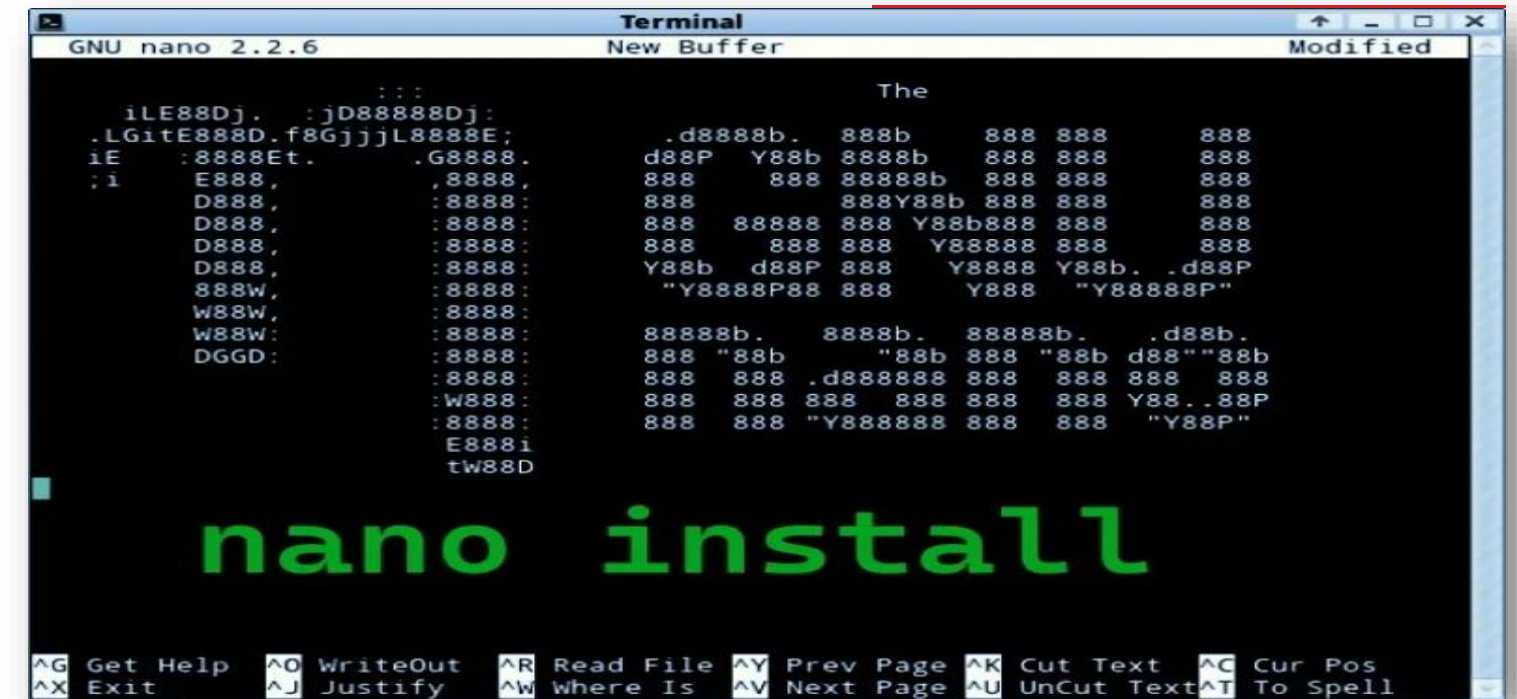
O APT é frequentemente considerado um dos melhores recursos do Debian, sujeito a um rigoroso controle de qualidade pelos desenvolvedores do Debian.

# Exemplo de uso apt-get

## Instalação do editor nano

Diferente de editores mais complexos, como vim ou emacs, o nano apresenta comandos básicos exibidos na parte inferior da tela, como **CTRL + O** para **salvar**, **CTRL + X** para **sair** e **CTRL + K** para **cortar uma linha**. Ele é útil para editar arquivos de configuração do sistema, scripts e outros documentos de texto de maneira rápida e eficiente.

```
aluno@aluno-Virtual-Machine:~$ sudo apt-get install nano
```



O nano é um editor de texto simples e intuitivo para o terminal do Linux, amplamente utilizado por sua facilidade de uso, especialmente por iniciantes. Ele permite editar arquivos de texto diretamente no terminal sem a necessidade de interfaces gráficas.

Antes de instalar você pode verificar se eventualmente ele já veio pré-instalado digitando:

**nano --version**

**Um pouquinho de Cybersegurança**



# Alguns comandos Linux de Cybersegurança

## find

Localiza arquivos suspeitos.

```
aluno@aluno-Virtual-Machine:~$ find / -type f -perm 777
```

*O exemplo acima Localiza arquivos com permissões perigosas.*

# Alguns comandos Linux de Cybersegurança

## journalctl

Análise de Logs do Sistema – Útil para verificar logs de serviços críticos.

```
aluno@aluno-Virtual-Machine:~$ journalctl -u sshd | grep "Failed password"
```

*O exemplo acima Busca falhas de autenticação no serviço SSH.*

# Alguns comandos Linux de Cybersegurança

## grep

Filtragem de Dados – Busca padrões em **logs** ou arquivos.

Logs são registros de eventos do sistema operacional, como alterações no sistema, mensagens da inicialização, erros, alertas e desligamentos inesperados.

```
aluno@aluno-Virtual-Machine:~$ grep -i "error" /var/log/syslog
```

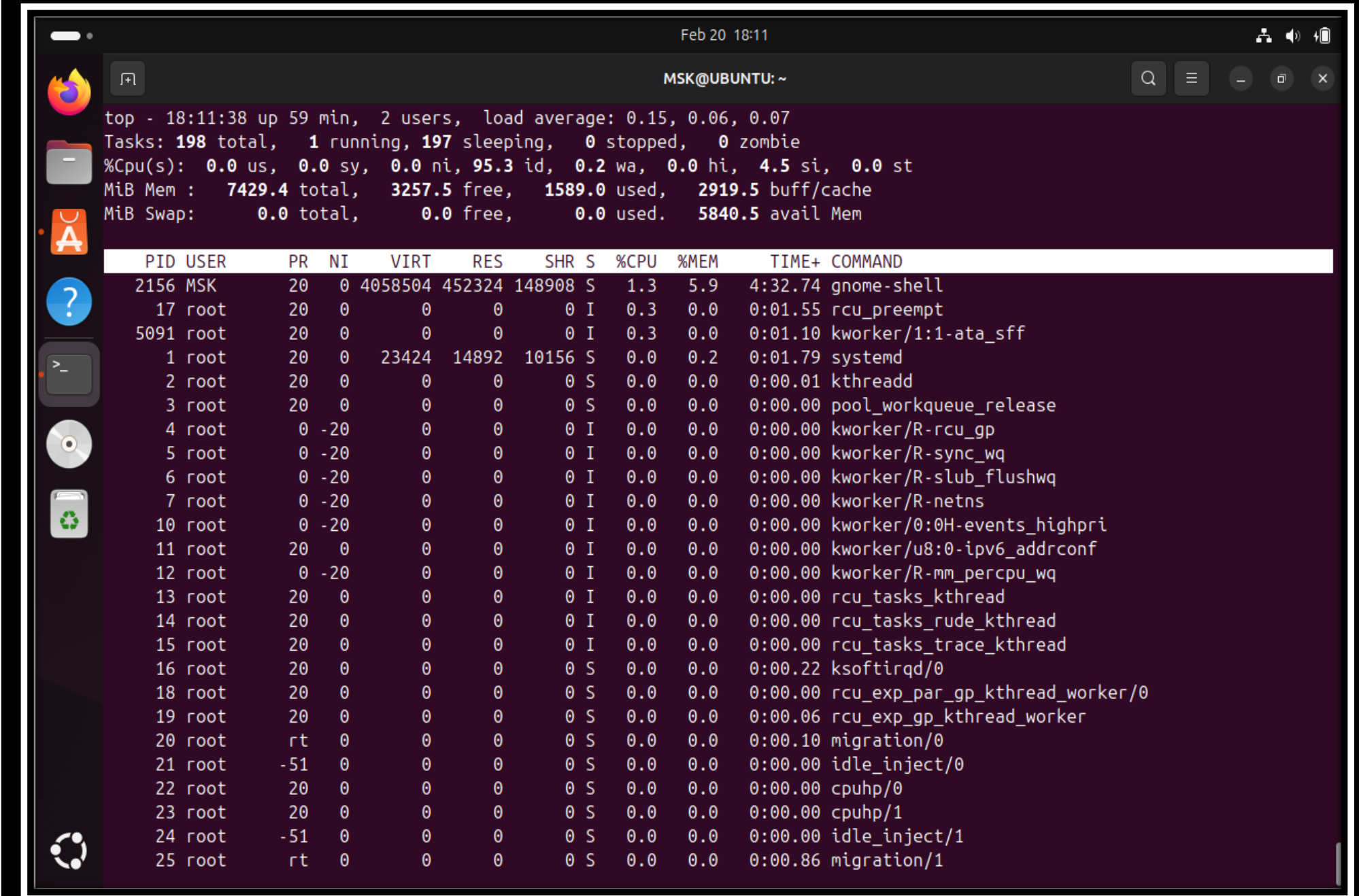
*O exemplo acima Busca Procura por ocorrências de "error" no log do sistema*

# Alguns comandos Linux de Cybersegurança

## top

Analisa processos ativos  
e recursos usados.

`aluno@aluno-Virtual-Machine:~$ top`



```
Feb 20 18:11
MSK@UBUNTU: ~
top - 18:11:38 up 59 min,  2 users,  load average: 0.15, 0.06, 0.07
Tasks: 198 total,  1 running, 197 sleeping,  0 stopped,  0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 95.3 id,  0.2 wa,  0.0 hi,  4.5 si,  0.0 st
MiB Mem :  7429.4 total,  3257.5 free,  1589.0 used,  2919.5 buff/cache
MiB Swap:   0.0 total,   0.0 free,   0.0 used.  5840.5 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 2156 MSK       20   0 4058504 452324 148908 S   1.3   5.9   4:32.74 gnome-shell
    17 root        20   0        0        0        0 I   0.3   0.0   0:01.55 rcu_preempt
 5091 root        20   0        0        0        0 I   0.3   0.0   0:01.10 kworker/1:1-ata_sff
     1 root        20   0   23424  14892  10156 S   0.0   0.2   0:01.79 systemd
     2 root        20   0        0        0        0 S   0.0   0.0   0:00.01 kthreadd
     3 root        20   0        0        0        0 S   0.0   0.0   0:00.00 pool_workqueue_release
     4 root         0 -20        0        0        0 I   0.0   0.0   0:00.00 kworker/R-rcu_gp
     5 root         0 -20        0        0        0 I   0.0   0.0   0:00.00 kworker/R-sync_wq
     6 root         0 -20        0        0        0 I   0.0   0.0   0:00.00 kworker/R-slab_flushwq
     7 root         0 -20        0        0        0 I   0.0   0.0   0:00.00 kworker/R-netns
    10 root         0 -20        0        0        0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
    11 root        20   0        0        0        0 I   0.0   0.0   0:00.00 kworker/u8:0-ipv6_addrconf
    12 root         0 -20        0        0        0 I   0.0   0.0   0:00.00 kworker/R-mm_percpu_wq
    13 root        20   0        0        0        0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
    14 root        20   0        0        0        0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
    15 root        20   0        0        0        0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
    16 root        20   0        0        0        0 S   0.0   0.0   0:00.22 ksoftirqd/0
    18 root        20   0        0        0        0 S   0.0   0.0   0:00.00 rcu_exp_par_gp_kthread_worker/0
    19 root        20   0        0        0        0 S   0.0   0.0   0:00.06 rcu_exp_gp_kthread_worker
    20 root        rt    0        0        0        0 S   0.0   0.0   0:00.10 migration/0
    21 root       -51   0        0        0        0 S   0.0   0.0   0:00.00 idle_inject/0
    22 root        20   0        0        0        0 S   0.0   0.0   0:00.00 cpuhp/0
    23 root        20   0        0        0        0 S   0.0   0.0   0:00.00 cpuhp/1
    24 root       -51   0        0        0        0 S   0.0   0.0   0:00.00 idle_inject/1
    25 root        rt    0        0        0        0 S   0.0   0.0   0:00.86 migration/1
```

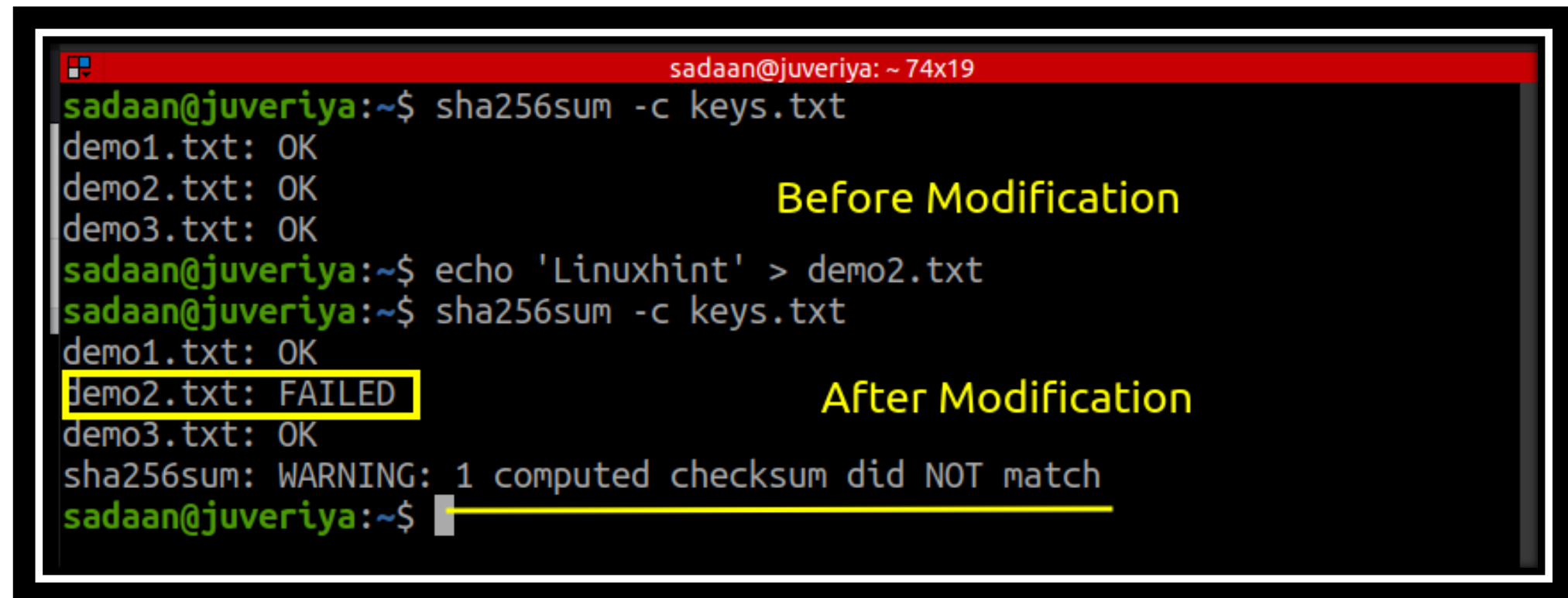


# Alguns comandos Linux de Cybersegurança

## sha256sum

### Verificação de Integridade –

Para validar a integridade de arquivos críticos.



```
sadaan@juveriya: ~ 74x19
sadaan@juveriya:~$ sha256sum -c keys.txt
demo1.txt: OK
demo2.txt: OK
demo3.txt: OK
sadaan@juveriya:~$ echo 'Linuxhint' > demo2.txt
sadaan@juveriya:~$ sha256sum -c keys.txt
demo1.txt: OK
demo2.txt: FAILED
demo3.txt: OK
sha256sum: WARNING: 1 computed checksum did NOT match
sadaan@juveriya:~$
```

Before Modification

After Modification

**Código:** `sha256sum arquivo_suspeito` (Compara o hash do arquivo com um valor de referência).

# Alguns comandos Linux de Cybersegurança

## SS

**Monitoramento de Conexões de Rede** – Substituto moderno do netstat, usado para monitorar conexões ativas e portas abertas.

```
aluno@aluno-Virtual-Machine:~$ ss -tunap
```

*Exibe conexões TCP/UDP ativas com processos associados).*