



## **ROTERIO DE AULA PRÁTICA**

CAMPINAS-SP

2025

NOME:TASSIANA MILKA FONTANA SOARES

## **ROTERIO DE AULA PRÁTICA**

Relatório da aula prática de realizar exercícios proposto de comandos Linux e VirtualBox.

CAMPINAS-SP

2025

# SUMÁRIO

PROCESSOS E THREADS .....	6
1.Introdução .....	6
2.Objetivos .....	6
3.Metódos .....	7
3.1. Thread .....	7
3.1.1. Tipos de Thread .....	7
3.1.1.1. Nível de kernel.....	7
3.1.1.1.1. Tipos de Kernels .....	7
3.1.1.2. Nível de usuário.....	8
3.1.1.3. Multiplexação.....	8
3.2. Comunicação entre Processos.....	8
3.3. Comunicação entre Processos com Pipes.....	11
3.3.1. Pipes .....	11
3.4. Estados de Threads .....	11
3.5. Comparação de Arquiteturas .....	13
4.Resolução .....	13
5.Conclusão .....	14
6. Referências Bibliográficas.....	14
GERENCIAMENTO DE ARQUIVOS E SISTEMAS DE DIRETÓRIOS .....	16
1.Introdução .....	16
2.Objetivos .....	14
3.Metódos .....	14
3.1. Gerenciamento de Arquivos e Diretórios .....	17
3.1.1.ls.....	14
3.1.2.pwd.....	15
3.1.3.cd .....	15

2.1.4.mkdir.....	18
3.1.5.touch .....	18
3.1.6.cp .....	18
3.1.7.mv.....	18
3.1.8.rm .....	19
3.2.Gerenciamento de Pacotes.....	19
3.3. Controle de Permissões .....	21
3.4. Diferenças entre Arquiteturas.....	21
3.4.1. Debian.....	21
3.4.2. Red Hat.....	21
4.Resolução .....	22
5.Conclusão .....	22
6. Referências Bibliográficas.....	22
<b>GERENCIAMENTO DE MEMÓRIA:ALOCAÇÃO E MEMÓRIA VIRTUAL .....</b>	<b>22</b>
1.Introdução .....	23
2.Objetivos .....	23
3.Metódos .....	23
3. 1. HTOP .....	25
3.2. Comandos de Consulta de Memória .....	26
3.3. Divisão de Memória .....	26
3.4. Comparação de Arquiteturas de Sistemas.....	27
4.Resolução .....	28
5.Conclusão .....	28
6. Referências Bibliográficas.....	28
<b>VIRTUALIZAÇÃO E CONTAINERIZAÇÃO.....</b>	<b>30</b>
1.Introdução .....	30
2.Objetivos .....	31
3.Metódos .....	31
3.1. Docker.....	31
3.2. Aplicação simples de contêiner .....	32
3.4. Comparação das Arquiteturas.....	33
3.4.1. Virtualização .....	33

3.4.1.1. Vantagens.....	33
3.4.1.2. Desvantagens.....	34
3.4.2. Contêineres .....	34
3.4.2.1. Vantagens.....	34
3.4.2.2. Desvantagens .....	35
4.Resolução .....	35
5.Conclusão .....	35
6. Referências Bibliográficas.....	36

# PROCESSOS E THREADS

## 1. Introdução

Os sistemas operacionais é uma utilidade muito positiva na utilização de conectar com o usuário para facilitar o acesso ao ambiente virtual pelos seus meios de administração dos gerenciamentos e processos das arquiteturas.

Nas suas diferenciações e semelhanças em descrição dos comandos no terminal para visualizar os itens importantes de dados armazenados, funcionamento e realizar abertura de informações.

## 2. Objetivos

- O objetivo é descrever os comandos do Linux e pesquisa sobre o que foi proposto.

## **3.Metódos**

- A seguir contém todas partes desde o começo para finalização.

### **3.1. Thread**

- Thread são entidade de um processo para compartilhar o seu espaço de endereço virtual e recursos do sistema que mantém manipulação de exceção, uma prioridade de agendamento, armazenado em um local que identifica exclusivamente em um conjunto de estruturas.

#### **3.1.1. Tipos de Thread**

- Divido em seus diferentes tipos de implementação que são considerados conforme a seguir.

##### **3.1.1.1. Nível de kernel**

- Um nível de operação do sistema operacional que administra o sistema operacional.CPU,memória e os dispositivos de entrada e saída do acesso ao total ao hardware e pode executar instruções críticas das suas funções.

###### **3.1.1.1.1. Tipos de Kernels**

- Monolítico são um único bloco de código que gerencia todos os serviços do sistema operacional;
- Microkernel é modular de funções básicas no kernel e serviços adicionais da execução em espaço usuário para impactar na performance e na segurança do sistema.

### **3.1.1.2. Nível de usuário**

- Contém permissão com o papel ou função do usuário, garantindo para o acesso de funcionalidades do sistema das configurações e recursos dos usuários comuns para visualização de informações que depende do nível da complexidade para realizar funcionalidades.

### **3.1.1.3. Multiplexação**

- Multiplexação threads é uma técnica que permite ao processador de várias tarefas para utilizar a abordagem da possibilidade com tempo do processador para distribuir forma correta na execução das variações múltiplas tarefas simultaneamente em um único núcleo.

## **3.2. Comunicação entre Processos**

Conforme solicitado foi realizado dentro da máquina virtual box na utilidade de comando na descrição próxima:

- Ping de uma página web onde foi informado o endereço web como na FIGURA 1.

FIGURA 1:Ping

```
vboxuser@Ubuntu:~$ ping github.com
PING github.com (4.228.31.150) 56(84) bytes of data.
64 bytes from 4.228.31.150: icmp_seq=1 ttl=255 time=8.04 ms
64 bytes from 4.228.31.150: icmp_seq=2 ttl=255 time=11.2 ms
64 bytes from 4.228.31.150: icmp_seq=3 ttl=255 time=10.5 ms
64 bytes from 4.228.31.150: icmp_seq=4 ttl=255 time=9.17 ms
64 bytes from 4.228.31.150: icmp_seq=5 ttl=255 time=8.11 ms
64 bytes from 4.228.31.150: icmp_seq=6 ttl=255 time=10.5 ms
64 bytes from 4.228.31.150: icmp_seq=7 ttl=255 time=9.35 ms
64 bytes from 4.228.31.150: icmp_seq=8 ttl=255 time=7.56 ms
64 bytes from 4.228.31.150: icmp_seq=9 ttl=255 time=10.0 ms
64 bytes from 4.228.31.150: icmp_seq=10 ttl=255 time=9.08 ms
64 bytes from 4.228.31.150: icmp_seq=11 ttl=255 time=8.06 ms
64 bytes from 4.228.31.150: icmp_seq=12 ttl=255 time=10.5 ms
64 bytes from 4.228.31.150: icmp_seq=13 ttl=255 time=9.21 ms
64 bytes from 4.228.31.150: icmp_seq=14 ttl=255 time=7.85 ms
64 bytes from 4.228.31.150: icmp_seq=15 ttl=255 time=11.0 ms
64 bytes from 4.228.31.150: icmp_seq=16 ttl=255 time=10.3 ms
64 bytes from 4.228.31.150: icmp_seq=17 ttl=255 time=9.47 ms
64 bytes from 4.228.31.150: icmp_seq=18 ttl=255 time=10.6 ms
64 bytes from 4.228.31.150: icmp_seq=19 ttl=255 time=9.19 ms
64 bytes from 4.228.31.150: icmp_seq=20 ttl=255 time=11.9 ms
64 bytes from 4.228.31.150: icmp_seq=21 ttl=255 time=11.1 ms
64 bytes from 4.228.31.150: icmp_seq=22 ttl=255 time=9.78 ms
64 bytes from 4.228.31.150: icmp_seq=23 ttl=255 time=8.94 ms
64 bytes from 4.228.31.150: icmp_seq=24 ttl=255 time=7.91 ms
64 bytes from 4.228.31.150: icmp_seq=25 ttl=255 time=11.1 ms
64 bytes from 4.228.31.150: icmp_seq=26 ttl=255 time=10.0 ms
64 bytes from 4.228.31.150: icmp_seq=27 ttl=255 time=10.1 ms
64 bytes from 4.228.31.150: icmp_seq=28 ttl=255 time=7.74 ms
64 bytes from 4.228.31.150: icmp_seq=29 ttl=255 time=10.6 ms
64 bytes from 4.228.31.150: icmp_seq=30 ttl=255 time=8.63 ms
64 bytes from 4.228.31.150: icmp_seq=31 ttl=255 time=11.1 ms
```

Fonte: autoria própria

- Já na FIGURA 2 demonstra sobre ps encontrado.

FIGURA 2:Ps

```
vboxuser@Ubuntu:~$ ps
  PID TTY          TIME CMD
 6565 pts/0    00:00:00 bash
 6576 pts/0    00:00:00 ps
```

Fonte: autoria própria

- O próximo comando descrito como na FIGURA 3 o top que em tempo real sobre o funcionamento.

FIGURA 3:Top

```
vboxuser@Ubuntu:~$ top
top - 17:37:24 up 3:23, 1 user, load average: 0.17, 0.16, 0.11
Tasks: 180 total, 1 running, 179 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.6 id, 0.0 wa, 0.0 hi, 0.4 si, 0.0 st
MiB Mem : 1967.8 total, 325.8 free, 1040.6 used, 789.4 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 927.1 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 3331 vboxuser  20   0 3502708 333696 97752 S  0.3 16.6  1:22.46 gnome-shell
 6558 vboxuser  20   0 563788 53664 42948 S  0.3  2.7  0:00.57 gnome-terminal-
  1 root    20   0 23196 12860 7996 S  0.0  0.6  0:08.42 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
 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  0:05.38 ksoftirqd/0
 17 root    20   0    0    0    0 I  0.0  0.0  0:01.19 rCU_prempt
 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  0:00.00 rCU_exp_gp_kthread_worker
 20 root    rt   0    0    0    0 S  0.0  0.0  0:00.26 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 kdevtmpfs
 24 root    0 -20   0    0    0 I  0.0  0.0  0:00.00 kworker/R-inet_frag_wq
 26 root    20   0    0    0    0 S  0.0  0.0  0:00.00 kauditd
```

Fonte: autoria própria

### 3.3. Comunicação entre Processos com Pipes

- O comando `ls -l | less` são utilizados para os processos de entrada e estabelecer um resultado de saída no terminal para os pipes.

#### 3.3.1. Pipes

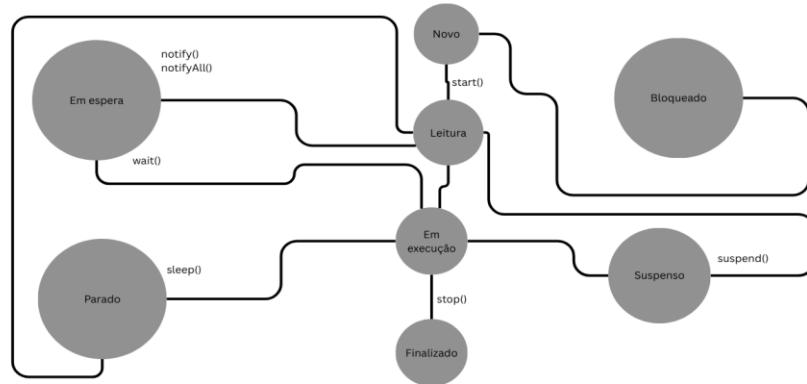
Os pipes é um canal de comunicação entre os processos conhecidos pelas suas diferenciações em tópicos próximos a seguir:

- Pipes Unidirecionais é o redirecionamento da saída padrão de programa da entrada padrão e de outra do recurso utilizado em sistemas Unix/Linux sobre saídas de dados do recurso versátil para permitir redirecionado a entrada ou saída padrão do processo para outra sequência;
- Pipes Nomeados (FIFO) contém a principal função de extensão do sistema Unix/Linux e de métodos da comunicação dos processos para realizar com outro processo que pode durar somente no tempo da execução ou ser excluído quando não mais usado pelos comandos como `mknod` ou `mkfifo`.

### 3.4. Estados de Threads

- Os estados são compostos em várias etapas conforme na FIGURA 4 deste o começo para finalização.

FIGURA 4:Diagrama dos Estados de Threads



Fonte: autoria própria

Diante a descrição vai conter a seguir como cada uma delas funciona durante a realização:

- Novo é o método de start() que entra no estado que executa o run() para quando criamos um thread e chamamos seu método start() ela entra no estado que vai depender do gerenciador de threads quando criada para execução;
- Executando é o método run() de uma thread para finalizar o chamado para entrar no estado que pode voltar ao começo conforme o tempo do CPU ou método yield() chamado;

- Bloqueado é um estado na finalidade de torna inativa que pode ser chamado os métodos `join()`, `sleep()`, `wait()` or `lock()` e que está aguardando alguma operação de entrada/saída (I/O) ser finalizada;
- Finalizada são todas as instruções da thread concluído e não consegue criar novamente chamar o método `start()` que contém a necessidade de criar uma nova instância.

### **3.5. Comparação de Arquiteturas**

- Arquiteturas conhecidas como monolítica interligada com Kernel de apenas funções básicas, microkernel de kernel na execução essenciais, camadas definidas em níveis, máquina virtual são o acesso de multiplas formas de como Hyper -V e híbrida contendo monolítico com microkernel e semelhanças entre todas para funcionar ao usuário possuir o seu acesso na conexão do hardware de utilização.

## **4.Resolução**

O desenvolvimento obteve a visualização dos comandos digitados em uma máquina virtual para visualizar sobre dados importantes e compreender a finalidade da interligação das thread do seu funcionamento, comunicação de processos e arquiteturas.

## 5.Conclusão

Durante a realização foi pesquisado os principais pontos de thread e seus estados na interligação com os processos que caracteriza seus tipos dentro do sistema operacional e o hardware das informações de comunicação administrada.

## 6. Referências Bibliográficas

MICROSOFT.Sobre Processos e Threads. Disponível em:<https://learn.microsoft.com/pt-br/windows/win32/procthread/about-processes-and-threads> .Acesso em:28/09/2025.

CAETANO INFORMATICA.O que é Kernel Level. Disponível em: <https://wcaetanoinformatica.com.br/glossario/o-que-e-kernel-level-entenda-sua-importancia/#:~:text=O%20termo%20%E2%80%9Ckernel%20level%E2%80%9D%20refere-se%20ao%20n%C3%ADvel%20de,mem%C3%B3ria%20e%20os%20dispositivos%20de%20entrada%20e%20sa%C3%ADda.>.Acesso em:28/09/2025.

VSP SOLUTION.Terminos e Condições.Disponível em: [https://vpssolution.com.br/glossario/o-que-e-nivel-de-acesso/](https://vpsolution.com.br/glossario/o-que-e-nivel-de-acesso/).Acesso em:28/09/2025.

INCC.BR.Processos e Threads.Disponível em: <https://www.lncc.br/~borges/ist/SO1/cap02.pdf> .Acesso em:28/09/2025.

MENEGUITE.Pipes-Comunicação entre Processos.Disponível em: <https://meneguite.com/2010/01/15/pipes-comunicacao-entre-processos/> .Acesso em:28/09/2025.

ARQUIVODECÓDIGOS.Entenda o Ciclo de Vida de uma Thread. Disponível em:

<https://www.arquivodecodigos.com.br/dicas/2767-java-entenda-o-ciclo-de-vida-de-uma-thread.html>.Acesso em:28/09/2025.

WIKI.Sistema Operacionais:Conceitos e Mecanismo. Disponível em:

<https://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-03.pdf> Acesso em:28/09/2025.

# **GERENCIAMENTO DE ARQUIVOS E SISTEMAS DE DIRETÓRIOS**

## **1.Introdução**

A evolução dos sistemas operacionais fez que os usuários armazenam vários arquivos de informações estabelecendo a necessidade de gerenciamento dos arquivos dentro de pastas centralizadas pelos diretórios conhecidas das funcionalidades rápidas e simples ao acesso.

Na sequência contemos outras formas de visualizar pelos terminais de comandos dos sistemas para criação, recuperar, modificar ou excluir e podemos compartilhar os dados da outra máquina de diretório diferente.

## **2.Objetivos**

- O objetivo é descrever os casos proposto de linux os comandos e pesquisar sobre o assunto destacado.

## **3.Metódos**

- A seguir contém cada etapas mais importante definida para realizar.

### **3.1. Gerenciamento de Arquivos e Diretórios**

- A seguir vai conter todas etapas dos comandos no terminal sobre gerenciamento de arquivos e diretórios.

#### **3.1.1.ls**

Esse comando fez eu obter o acesso aos arquivos, pastas armazenadas no sistema e software de aplicativos abaixo como exemplo:

- Arquivo de extensão txt;
- Pastas com nome dados;
- Downloads o acesso automático a um jogo.

#### **3.1.2.pwd**

Acessei e foi exibido o diretório principal da demonstração a seguir:

/home/vboxuser/projetos

#### **3.1.3.cd**

O cd é para navegação entre os diretórios de abrir pasta principal e a subpastas que foram criadas como exemplo:

/home/vboxuser/projetos

/home/vboxuser/Documents

### **2.1.4.mkdir**

Contém a funcionalidade de criação de uma pasta conforme a descrição a seguir:

mkdir

PastaNova

### **3.1.5.touch**

Obtém a facilidade de criar um arquivo na sequência de extensão:

touch

dadosNova.txt

### **3.1.6.cp**

O comando copia o arquivo que deseja armazenado dessa maneira realizada:

cp

dadosNova.txt /home/vboxuser/Documents/

### **3.1.7.mv**

A sua finalidade é mover e renomear arquivos na formalidade abaixo:

Mover

mv

```
dadosNova.txt /home/ Downloads/
```

Renomear

```
mv
```

```
dadosNova.txt DadosNova.txt
```

### **3.1.8.rm**

Removido o arquivo que armazenado dentro do sistema com o comando:

```
rm
```

```
DadosNova.txt
```

## **3.2.Gerenciamento de Pacotes**

O Gerenciamento de Pacotes são ideais na distribuição do Linux de Ubuntu a utilização de `apt` que possui a funcionalidade para facilitar instalação, remoção e atualização dos repositórios de pacotes.

Para demonstração na FIGURA 1 vai conter o desenvolvimento de atualização do seguinte comando:

```
sudo apt update
```

## FIGURA 1:Sudo Update

```

Get:24 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:25 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [18.0 kB]
Get:26 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [28.0 kB]
Get:27 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [5,228 B]
Get:28 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:29 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [384 B]
Get:30 http://archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [426 kB]
Get:31 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:32 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [544 B]
Get:33 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,484 kB]
Get:34 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [299 kB]
Get:35 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:36 http://archive.ubuntu.com/ubuntu noble-updates/universe Icons (48x48) [232 kB]
Get:37 http://archive.ubuntu.com/ubuntu noble-updates/universe Icons (64x64) [363 kB]
Get:38 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.1 kB]
Get:39 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [32.0 kB]
Get:40 http://archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5,500 B]
Get:41 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:42 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [488 B]
Get:43 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [39.9 kB]
Get:44 http://archive.ubuntu.com/ubuntu noble-backports/main Translation-en [9,152 B]
Get:45 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,076 B]
Get:46 http://archive.ubuntu.com/ubuntu noble-backports/main Icons (48x48) [9,518 B]
Get:47 http://archive.ubuntu.com/ubuntu noble-backports/main Icons (64x64) [11.2 kB]
Get:48 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:49 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [30.2 kB]
Get:50 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17.4 kB]
Get:51 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [19.2 kB]
Get:52 http://archive.ubuntu.com/ubuntu noble-backports/universe Icons (48x48) [23.3 kB]
Get:53 http://archive.ubuntu.com/ubuntu noble-backports/universe Icons (64x64) [32.3 kB]
Get:54 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 12.5 MB in 8s (1,592 kB/s)

```

Fonte: autoria própria

O pacote informado para realizar o download é o navegador Firefox da descrição:

`sudo apt install firefox`

Para remover um pacote foi utilizado o comando:

`sudo apt remove firefox`

Já o de pesquisa para saber os pacotes ainda encontrados:

`sudo apt search`

### 3.3. Controle de Permissões

O controle é conhecido pelos comandos chmod que são arquivo para determinar alteração do modo na utilização de um operador como soma, subtração ou sinal de igual e o chown é definido do seu grupo de arquivos, diretórios e link de símbolos. Diante desses comandos obtém um grupo importante juntamente para usar descrito a seguir:

- A funcionalidade do r é permitido para leitura;
- O w é definido para escrita;
- Já x definido como executável.

### 3.4. Diferenças entre Arquiteturas

- Com diversas arquiteturas o Linux contém que cada uma em uma finalidade.

#### 3.4.1. Debian

- Um software livre e de código aberto na oficialização em 15 de setembro de 1993 na primeira versão estável do lançamento em 17 de junho de 1996 com acesso a repositório online de mais 51 mil pacotes de software livre dos núcleos que é disponível em 76 línguas das operações do sistema gestor de pacotes para realizar diferentes ferramentas pela administração de APT compartilham informações de dependência e pacotes armazenados em cache.

#### 3.4.2. Red Hat

- Uma distribuição GNU/Linux estabelecida estável e acessível pelo software de código aberto lançada em 1993 no Estados Unidos pela colaboração aberta para criar melhor formas, com passar do tempo começou a incorporar o open source aos data centers que tornou uma das soluções de software da história na finalidade de atender demanda na nuvem híbrida, middleware, integração ágil, desenvolvimento de aplicações, gerenciamento e automação.

## 4.Resolução

A realização das etapas foi descrita neste que conteve a necessidade de explicar os comandos de Linux relacionado sobre criações, renomeação, excluir um arquivo, pasta de um diretório e reconhecer onde foi armazenado e compreender sobre outra arquitetura de distribuição do sistema operacional.

## 5.Conclusão

Diante a descrição do gerenciamento de arquivo dos requisitos e dados das informações na utilidade em linhas de comandos no terminar para a parte inicial e finalização no conjunto elaborado sobre pesquisas diferentes das distribuições de lançamentos e funcionalidades.

## 6. Referências Bibliográficas

SEMPREUPDATE.Gerenciamento de Pacotes no Linux:apt,dnf,pacman e flatpak. Disponível em:<https://sempreupdate.com.br/gerenciamento-de-pacotes-no-linux-apt-dnf-pacman-e-flatpak/> .Acesso em:28/09/2025.

HOSTINGER.Como Alterar Permissões de Pasta Linux via Linha de Comando. Disponível em:<https://www.hostinger.com.br/tutoriais/como-alterar-permissoes-e-proprietarios-via-linha-de-comando#:~:text=Neste%20tutorial%2C%20voc%C3%AA%20aprender%C3%A1%20como%20alterar%20permiss%C3%B5es,tarefa%3A%20chmod%20e%20chown.%20O%20que%20voc%C3%AA%20precisa%3F>.Acesso em:28/09/2025.

FREECODECAMP.O Comando Chmod-Como Alterar Permissões de Arquivos no Linux. Disponível em:<https://www.freecodecamp.org/portuguese/news/o-comando-chmod-como-alterar-permissoes-de-arquivos-no-linux/>.Acesso em:28/09/2025.

HOSTINGER.O Que é Comando Chown Linux e Como Utilizá-lo. Disponível em:<https://www.hostinger.com.br/tutoriais/comando-chown-linux>.Acesso em:28/09/2025.

LINUXIZE.Chown Comando em Linux(Arquivo Ownership).Disponível em: <https://linuxize.com/post/linux-chown-command/> Acesso em:28/09/2025.

WIKIPÉDIA. Debian. Disponível em: <https://pt.wikipedia.org/wiki/Debian> .Acesso em:28/09/2025.

WIKIPÉDIA. Red Hat. Disponível em: [https://pt.wikipedia.org/wiki/Red\\_Hat](https://pt.wikipedia.org/wiki/Red_Hat) .Acesso em:28/09/2025.

Red Hat. Nossa História. Disponível em: <https://www.redhat.com/pt-br/about/brand/standards/history>.Acesso em:28/09/2025.

# **GERENCIAMENTO DE MEMÓRIA: ALOCAÇÃO E MEMÓRIA VIRTUAL**

## **1. Introdução**

Em sistemas operacionais contém várias funcionalidades como o gerenciamento de memória destinado nos armazenamentos de dados da sequência de alguns requisitos das etapas destinadas para alocação física ou virtual.

Assim, consegue o acesso no terminar o tempo exato dos aplicativos e informações estão executando pelos números observa-se que alguns obtém mais utilidade de espaço e os outros menos.

## **2. Objetivos**

- O objetivo é descrever sobre o que foi proposto para saber o comando do linux e assunto destacado de pesquisa.

## **3. Métodos**

- A seguir contém cada etapas mais importante definida para realizar.

### 3. 1. HTOP

- O comando HTOP é destinado para acessar os processos e gerenciador interativo de uma lista completa em execução. Para analisar é necessário realizar o download no terminal e descrever htop como na FIGURA 1 que observa o processador da CPU, Mem e Swap em tempo real da diferença de minutos e semelhanças.

FIGURA 1:Htop

```
CPU[| 1.5%] Tasks: 155, 468 thr, 70 kthr; 1 running
Mem[||||| 1.10G/1.92G] Load average: 0.54 0.41 0.25
Swp[ 0K/0K] Uptime: 14:49:02

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
4774 vboxuser 20 0 3390M 350M 136M S 1.5 17.8 0:09.82 /usr/bin/gnom
5584 vboxuser 20 0 11372 5004 3596 R 1.5 0.2 0:06.05 htop
4830 vboxuser 20 0 3390M 350M 136M S 0.8 17.8 0:00.22 /usr/bin/gnom
5566 vboxuser 20 0 542M 54044 42632 S 0.8 2.7 0:01.21 /usr/libexec/
  1 root 20 0 23116 14132 9396 S 0.0 0.7 0:16.58 /sbin/init sp
  624 avahi 20 0 8668 4480 4096 S 0.0 0.2 0:01.33 avahi-daemon:
  625 messagebus 20 0 12420 7816 4616 S 0.0 0.4 0:05.44 @dbus-daemon
  628 gnome-remo 20 0 356M 16380 13948 S 0.0 0.8 0:00.79 /usr/libexec/
  643 polkitd 20 0 310M 12748 8204 S 0.0 0.6 0:01.88 /usr/lib/polk
  644 root 20 0 306M 7600 6832 S 0.0 0.4 0:00.38 /usr/libexec/
  658 root 20 0 305M 7636 6868 S 0.0 0.4 0:00.38 /usr/libexec/
  672 root 20 0 9424 2764 2636 S 0.0 0.1 0:00.64 /usr/sbin/cro
  675 root 20 0 302M 7076 6308 S 0.0 0.4 0:00.32 /usr/libexec/
  683 root 20 0 458M 14532 11716 S 0.0 0.7 0:01.15 /usr/libexec/
  684 root 20 0 306M 7600 6832 S 0.0 0.4 0:00.00 /usr/libexec/
  685 root 20 0 306M 7600 6832 S 0.0 0.4 0:00.00 /usr/libexec/
F1Help F2Setup F3Search F4Filter F5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit
```

Fonte: autoria própria

### 3.2. Comandos de Consulta de Memória

Alguns comandos são destinados para consultar memória na finalidade de exibir informações encontrada no sistema operacional conforme a seguir:

- free –h é destinada para informação da memória física e swap do exemplo de execução.

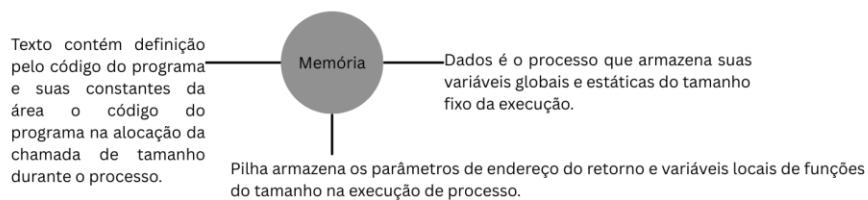
	total	used	free	sharred	buff/cache	available
Mem:	1.9Gi	1.3gi	92Mi	49Mi	755Mi	639Mi
Swap	0b	0b	0b			

- top monitora o processo como dos rots;
- ps aux demonstra os processos em execução do tempo como do root;
- pmap é um processo da memória destinado a demonstrar sobre options como visualizar, leitura, criar e limite.

### 3.3. Divisão de Memória

- Na memória contém algumas funcionalidades como texto, dados e pilha que estão descritas na FIGURA 2 do diagrama.

FIGURA 2:Diagrama de Memória



Fonte: autoria própria

### 3.4. Comparação de Arquiteturas de Sistemas

O x86 contém placa gráficas e de som, memória, armazenamento e CPU da maioria dos componentes de chips independentes na troca ou expandir alteração conectividade ou plataforma de hardware nos computadores e servidores. Já o ARM não possui uma CPU da unidade de processamento para físico que os outros controladores de hardware dos seus próprios designs das aplicações.

Nas semelhanças e diferenças entre essas duas arquiteturas de sistemas da tabela próxima alguns pontos extras:

<b>Arquiteturas</b>	<b>x86</b>	<b>ARM</b>
Consumo de energia	Mais alto, adequado para servidores	Baixo, ideal para dispositivos IoT
Casos de uso	Estações de trabalho e jogos	Dispositivos móveis e computação de ponta
Complexidade da Instrução	Instruções complexas e multietapas	Instruções simples do ciclo único

## 4. Resolução

Obteve a realização das etapas deste de compreender o sistema dos comandos para o gerenciamento da memória de escrita, dados e pilhas dos fundamentos sobre arquitetura com o tempo da execução dos processos envolvendo os hardwares de forma física ou virtual utilizada nos exemplos elaborados.

## 5. Conclusão

O desenvolvimento envolveu pesquisas para o sentido sobre dados das arquiteturas na interligação da memória em execução no terminal Linux em tempo real de todos os processos mais importante das ocupações abordando na facilidade do usuário acessar, executar e retirar da utilidade dos comandos.

## 6. Referências Bibliográficas

WIKIPÉDIA.Htop.Disponível em: <https://pt.wikipedia.org/wiki/Htop> .Acesso em:28/09/2025.

WIKI.Gerência de Memória.Disponível em:

[https://wiki.inf.ufpr.br/maziero/doku.php?id=pua:gerencia\\_de\\_memoria#:~:text=Data%20area%3A%20%C3%A9%20a%20mem%C3%B3ria%20de%20trabalho%20do,endere%C3%A7os%20de%20retorno%20e%20vari%C3%A1veis%20locais%20de%20fun%C3%A7%C3%B5es.](https://wiki.inf.ufpr.br/maziero/doku.php?id=pua:gerencia_de_memoria#:~:text=Data%20area%3A%20%C3%A9%20a%20mem%C3%B3ria%20de%20trabalho%20do,endere%C3%A7os%20de%20retorno%20e%20vari%C3%A1veis%20locais%20de%20fun%C3%A7%C3%B5es.) .Acesso em:28/09/2025.

RED HAT.Qual é a diferença entre ARM e x86?.Disponível em:<https://www.redhat.com/pt-br/topics/linux/ARM-vs-x86> .Acesso em:28/09/2025.

SINSMART.Arquitetura x86 vs.ARM:Uma comparação abrangente.Disponível em:<https://www.sinsmarts.com/pt/blog/x86-architecture-vs-arm-a-comprehensive-comparison/> .Acesso em:28/09/2025.

# VIRTUALIZAÇÃO E CONTAINERIZAÇÃO

## 1. Introdução

Na evolução do acesso aos sistemas operacionais contém o funcionamento da virtualização na finalidade de uma máquina virtual que aumenta mais de um sistema no container sem que ocupa muito espaço, perda ou não conseguir visualizar e rendimento de utilidade.

## 2. Objetivos

- O objetivo é descrever sobre o download do Docker e diferenças de virtualização e contêineres.

## 3. Métodos

- A seguir contém sobre descrição principal da pesquisa elaborada.

### 3.1. Docker

Docker é uma máquina virtual com um propósito para o desempenho, portabilidade e suporte de compartilhar o sistema operacional do host da necessidade de hardware.

Para realizar o download no Linux possui a necessidade de descrever alguns comandos no terminal como em descrição e na FIGURA 1:

```
sudo apt-get update
```

```
sudo apt-get install docker-ce
```

FIGURA 1:Sudo apt install

```
vboxuser@Ubuntu:~$ sudo apt-get update
[sudo] password for vboxuser:
Get:1 file:/cdrom noble InRelease
Ign:1 file:/cdrom noble InRelease
Get:2 file:/cdrom noble Release
Err:2 file:/cdrom noble Release
  File not found - /cdrom/dists/noble/Release (2: No such file or directory)
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
E: The repository 'file:/cdrom noble Release' no longer has a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
vboxuser@Ubuntu:~$ sudo apt-get install docker-ce
[sudo] password for vboxuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package docker-ce is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
```

Fonte: autoria própria

### 3.2. Aplicação simples de contêiner

Essa aplicação foi utilizada o Nginx descrevendo alguns comandos:

sudo	apt	update
sudo apt install nginx		

Depois informar no terminal sobre seguinte forma de desenvolvimento do html :

cd	/var/www
sudo	exemplo
cd	exemplo
sudo "\${EDITOR:-vi}" index.html	

Assim, consegue visualizar um simples html de um exemplo na FIGURA 2.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Hello, Nginx!</title>
</head>
<body>
    <h1>Hello, Nginx!</h1>
    <p>We have just configured our Nginx web server!</p>
</body>
</html>
```

FIGURA 2:Exemplo

# Hello, Nginx!

We have just configured our Nginx web server on Ubuntu Server!

Fonte: autoria própria

## 3.4. Comparação das Arquiteturas

- A seguir vai conter sobre o assunto interligando virtualização e contêineres das suas vantagens e desvantagens.

### 3.4.1. Virtualização

Uma tecnologia que simula os recursos físicos do hardware, sistemas operacionais, armazenamento que criam uma camada de abstração sobre o hardware do computador físico dividindo máquinas virtuais de forma própria do mesmo compartilhamento do recurso físico.

#### 3.4.1.1. Vantagens

- Eficiência de recursos são múltiplos aplicativos que executam no servidor físico único para maximizar o hardware de utilidade;
- Redução de custos para adquirir e manter vários servidores físicos economizado e diminuir a energia e espaço;
- Escalabilidade facilita sobre redução ou aumento dos recursos conforme a demanda;

- Gerenciamento simplificado são tarefas provisoriamente e configurada para reduzir erros;
- Redução de tempo de inatividade podem restaurar ou replicadas pelo caso de falhas.

### **3.4.1.2. Desvantagens**

- Sobrecarga de tarefas que o usuário não pode causar uma grande sobrecarga para executar pelo motivo que pode parar de executar;
- Segurança um software responsável que pode acontecer vulneráveis de ataques de vírus e afetar outro ambiente;
- Gastos de mais manutenção dos desgastes de uso da máquina é muito maior referindo aos hardwares;
- Memória RAM é uma técnica de muitas operações da avaliação do profissional que apresenta situações das falhas.

## **3.4.2. Contêineres**

Os contêineres são unidades isoladas capazes de agrupar bibliotecas dependências e configurações em um único pacote para executar o ambiente para garantir padronização e evitando conflitos e versão.

### **3.4.2.1. Vantagens**

- Redução de custos ajuda a manter uma infraestrutura mais enxuta reduzindo custos operacionais;
- Escalabilidade são horizontal que permite aplicações rapidamente ajustadas para atender a picos de demanda;
- Isolamento oferece problema em um contêiner que não afetam os demais;
- Portabilidade pode facilitar os movidos das diferenças plataformas que facilita a portabilidade de aplicações;

- Segurança é o isolamento de processos com maior segurança e reduzindo o risco de falhas;
- Ciclo de desenvolvimento e implantação são contêineres para acelerar o ciclo de desenvolvimento e implantação que permite testes mais rápidos e liberações mais rápidas de novas versões.

#### **3.4.2.2. Desvantagens**

- Complexidade adicional para implementar e gerenciamento da virtualização;
- Riscos de segurança é afetar o sistema para preocupação de organizar com os dados sensíveis;
- Isolamento limitado é o núcleo para compartilhar que não pode oferecer o isolamento completo necessário para algumas aplicações;
- Desempenho são leves que pode oferecer o mesmo desempenho das máquinas virtuais para o ambiente de alto desempenho.

## **4.Resolução**

Obteve a realização do download e conhecendo sobre docker na utilização da virtualização que compreende o entendimento de vantagens e desvantagens para o usuário está atualizado.

## **5.Conclusão**

No desenvolvimento foi descrito sobre a importância da virtualização para usar mais de um sistema ao mesmo tempo em uma máquina física e pelo meio do terminal o comando de Linux para conhecer sobre docker.

## 6. Referências Bibliográficas

HOSTINGER. O que é Docker e Como Ele Funciona! - Docker Explicado. Disponível em:<https://www.hostinger.com.br/tutoriais/o-que-e-docker> .Acesso em:28/09/2025.

UBUNTU. Tutorial. Disponível em:<https://ubuntu.com/tutorials/install-and-configure-nginx#1-overview> .Acesso em:28/09/2025.

IBM. O que é Virtualização?. Disponível em:<https://www.ibm.com/br-pt/think/topics/virtualization> .Acesso em:28/09/2025.

MICROSOFT. O que é Virtualização?. Disponível em:<https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-virtualization/> .Acesso em:28/09/2025.

DATAACAMP. Contêiners Versus Máquinas Virtuais:Uma Comparação Detalhada para Desenvolvedores. Disponível em:<https://www.datacamp.com/pt/blog/containers-vs-virtual-machines>.Acesso em:28/09/2025.