

RELATÓRIO DO PROJETO
GERENCIAMENTO DE ARQUIVOS E
PERMISSÕES

Disciplina: Sistemas Operacionais

Professor: Clóvis Ferraro

Grupo: 15

Sumário

1. Introdução
2. Metodologia
3. Comparação entre os Sistemas Operacionais
 - 3.1 Windows
 - 3.2 Android
 - 3.3 Linux
4. Análise Crítica
5. Conclusão
6. Autoavaliação
7. Referencias

1.Introdução

O gerenciamento de arquivos junto com o controle de permissões são funções essenciais de qualquer sistema operacional, por garantir que dados sejam organizados e definindo quem pode acessar e modificar essas informações. Este relatório tem como objetivo principal testar e analisar comandos de gerenciamento de arquivos e permissões em ambientes Windows, Linux e Android para entender o controle de acesso e a estrutura dos sistemas de arquivos compreendendo como esses processos atuam e funcionam em cada um dos ambientes testados.

Esses mecanismos não são apenas boas práticas são estruturas fundamentais que sustentam a confiabilidade de qualquer sistema operacional moderno. Eles fazem com que a utilização dos sistemas seja segura e de maior qualidade de informação, fornecendo um conforto aos usuários.

2.Metodologia

Windows: Para realizar os testes foi utilizado um hardware com o sistema operacional do Windows 11 já instalado no aparelho. foi criado um ambiente isolado e seguro na área de trabalho juntamente com diversos comandos pesquisados e essenciais para essas atividades. **Comandos:**

- mkdir C:\Users\andre\Desktop\Teste_Relatorio (Criou uma pasta segura para testes na área de trabalho)
- cd C:\Users\andre\Desktop\Teste_Relatorio (Entrar na pasta de teste)
- echo "arquivo de teste" > teste1.txt (Criou um arquivo de texto para testes)
- mkdir pasta_teste (criou uma pasta própria para testes)
- icacls teste1.txt (fez as leituras das permissões)
- icacls teste1.txt /grant "andrébook\andre:R" (MODIFICA permissões: concedeu apenas leitura (R) para o usuário andre)
- fsutil fsinfo volumeinfo C:(mostrou informações técnicas do sistema de arquivos NTFS)
- fsutil file createnew arquivo_grande.txt 1048576 (Criou um arquivo de 1MB (1048576 bytes) vazio)
- Dir (Listou todos os arquivos criados)
- Del -Path "~\Desktop\Teste_Relatorio" -Recurse -Force (Fez a limpeza, "Recurse" apaga subpastas, "-Force" ignora erros e permissões)
- Whoami (para ver o nome de usuário)

Android: Os testes foram realizados no emulador Pixel 4a – API 36, configurado no Android Studio, com acesso ao terminal através do comando adb shell. A partir desse ambiente, foram executadas operações de criação, listagem, modificação e análise de arquivos e diretórios, todas dentro da área pública do sistema (/sdcard), que é acessível ao usuário shell. Cada etapa foi registrada por meio de capturas de tela, representando os resultados obtidos

Linux: Para a realização dos teste em Linux, foi utilizada uma abordagem prática e exploratória, com o objetivo de compreender o funcionamento do gerenciamento de arquivos. Foi utilizado uma Máquina Virtual no "Oracle VirtualBox" com o sistema operacional do "Linux Lite 7.0".

3.Comparação entre os Sistemas Operacionais

3.1 Windows

1. Criação do Ambiente de Teste

- `mkdir C:\Users\andre\Desktop\Teste_Relatorio` e `cd`

```
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows
PS C:\Users\andre> mkdir C:\Users\%env:USERNAME%\Desktop\Teste_Relatorio

Diretório: C:\Users\andre\Desktop

Mode                LastWriteTime         Length Name
----                -
d-----          12/10/2025   10:38             Teste_Relatorio

PS C:\Users\andre> cd C:\Users\%env:USERNAME%\Desktop\Teste_Relatorio
PS C:\Users\andre\Desktop\Teste_Relatorio> |
```

Para iniciarmos começamos criando uma pasta de teste seguro e isolado para evitar possíveis complicações no aparelho com o sistema Windows. Foi criado a pasta “Teste_Relatorio” e logo em seguida entramos na pasta criada através do comando “`cd C:\Users\andre\Desktop\Teste_Relatorio`”.

- `echo "arquivo de teste" > teste1.txt` e `mkdir pasta_teste`

```
PS C:\Users\andre\Desktop\Teste_Relatorio> echo "arquivo teste" > teste1.txt
PS C:\Users\andre\Desktop\Teste_Relatorio> mkdir pasta_teste

Diretório: C:\Users\andre\Desktop\Teste_Relatorio

Mode                LastWriteTime         Length Name
----                -
d-----          12/10/2025   10:44             pasta_teste

PS C:\Users\andre\Desktop\Teste_Relatorio>
```

Com o comando “`echo "arquivo de teste" > teste1.txt`” foi criado um arquivo simples de texto para utilizarmos nos testes. O comando “`mkdir pasta_teste`” é usado para a criação de uma pasta própria para colocarmos esses arquivos.

2. Permissões com ICACLS

- Whoami

```
PS C:\Users\andre> whoami
andrébook\andre
PS C:\Users\andre>
```

Antes de continuarmos fizemos a confirmação do nome cadastrado do dispositivo para inserirmos corretamente durante os processos.

- icac ls teste1.txt

```
PS C:\Users\andre\Desktop\Teste_Relatorio> icac ls teste1.txt
teste1.txt AUTORIDADE NT\SISTEMA:(I)(F)
          BUILTIN\Administradores:(I)(F)
          ANDRÉBOOK\andre:(I)(F)

Processados com sucesso 1 arquivos; falha no processamento de 0 arquivos
PS C:\Users\andre\Desktop\Teste_Relatorio>
```

Para prosseguir realizamos a consulta das permissões atuais dos arquivos.

- icac ls teste1.txt /grant "andrébook\andre:R"

```
PS C:\Users\andre\Desktop\Teste_Relatorio> whoami
andrébook\andre
PS C:\Users\andre\Desktop\Teste_Relatorio> icac ls teste1.txt /grant "andrébook\andre:R"
arquivo processado: teste1.txt
Processados com sucesso 1 arquivos; falha no processamento de 0 arquivos
PS C:\Users\andre\Desktop\Teste_Relatorio> |
```

Modificou as permissões: concedeu apenas leitura (R) para o usuário “andré”. Com esses comandos e alterações foi possível o ter controle preciso sobre esses dados.

3. Sistema de Arquivo com FSUTIL

- fsutil fsinfo volumeinfo C:

```

PS C:\Users\andre\Desktop\Teste_Relatorio> fsutil fsinfo volumeinfo C:
Nome do volume:
Número de série do volume: 0x7459eaa3
Tamanho máx. do componente: 255
Nome do sistema de arquivos: NTFS
É ReadWrite
Não Tem Provisionamento Dinâmico
Suporta nomes de arquivo que diferenciam maiúsculas de minúsculas
Preserva maiusc/minúsc. nos nomes de arquivos
Dá suporte a Unicode em nomes de arquivo
Preserva e aplica ACLs
Suporta compactação baseada em arquivo
Suporta cotas de disco
Suporta arquivos esparsos
Suporta pontos de nova análise
Retorna Informações do Resultado do Fechamento do Identificador
Oferece suporte para Separar e Renomear no estilo POSIX
Suporta Bypass de E/S
Dá suporte a diretórios que diferenciam maiúsculas de minúsculas
Dá suporte a identificadores de objeto
Dá suporte ao sistema de arquivos criptografados
Suporta fluxos identificados
Suporta transações
Dá Suporte a Links Físicos
Dá Suporte a Atributos Estendidos
Dá Suporte a Abertura por ID de Arquivo
Oferece suporte ao Diário de USN
PS C:\Users\andre\Desktop\Teste_Relatorio>

```

O comando mostrou informações técnicas do sistema de arquivos NTFS

- **fsutil file createnew arquivo_grande.txt 1048576**

```

PS C:\Users\andre\Desktop\Teste_Relatorio> fsutil file createnew arquivo_grande.txt 1048576
Arquivo C:\Users\andre\Desktop\Teste_Relatorio\arquivo_grande.txt criado
PS C:\Users\andre\Desktop\Teste_Relatorio>

```

Com esse comando criamos um arquivo com um tamanho determinado e específico (1MB)

- **Dir**

```

PS C:\Users\andre\Desktop\Teste_Relatorio> dir

Diretório: C:\Users\andre\Desktop\Teste_Relatorio

Mode                LastWriteTime         Length Name
----                -
d-----          12/10/2025   10:44             pasta_teste
-a-----          12/10/2025   11:01       1048576 arquivo_grande.txt
-a-----          12/10/2025   10:44           32 teste1.txt

PS C:\Users\andre\Desktop\Teste_Relatorio>

```

O Dir foi utilizado para nos mostrar os resultados dos arquivos criados.

4. Limpeza

- **del -Path “~\Desktop\Teste_Relatorio” -Recurse -Force**

```
PS C:\Users\andre\Desktop> del -Path "~\Desktop\Teste_Relatorio" -Recurse -Force
PS C:\Users\andre\Desktop>
```

Com isso foi deletado a pasta em que estavam todos os testes (“-Recurse”, apaga subpastas, “-Force”, ignora erros e permissões).

3.2 Android

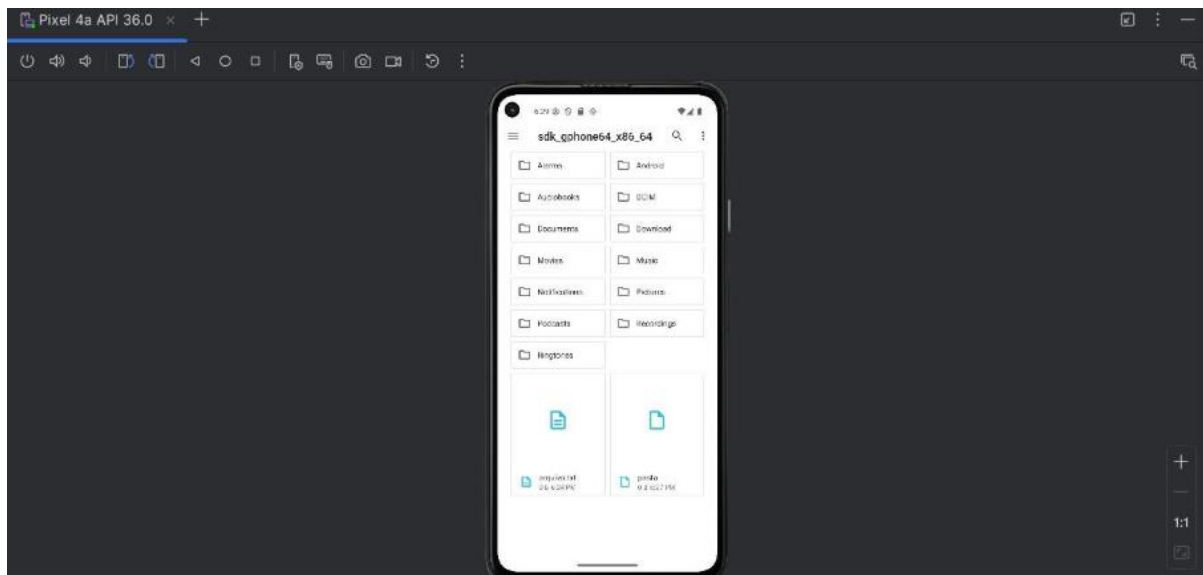


Figura 1 – Acesso ao shell do Android (adb shell) / Imagem do emulador mostrando o explorador de arquivos.

```
U Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\samsung\AndroidStudioProjects\MyApplication> cd C:\Users\samsung\AppData\Local\Android\Sdk\platform-tools
PS C:\Users\samsung\AppData\Local\Android\Sdk\platform-tools> ./adb devices
List of devices attached
emulator-5554    device

PS C:\Users\samsung\AppData\Local\Android\Sdk\platform-tools>
```

Figura 2 – Listagem inicial do diretório /sdcard.


```

lrw-r--r--  1 root root   21 2025-08-06 22:12 sdcard -> /storage/self/primary
:
total 68
lrw-r--r--  1 root root      30 2025-08-06 22:12 adb_keys -> /product/etc/security/adb_keys
drwxr-xr-x 101 root root    2040 2025-10-11 16:27 apex
lrw-r--r--  1 root root      11 2025-08-06 22:12 bin -> /system/bin
drwxr-xr-x 12 root root     260 2025-10-11 16:26 bootstrap-apex
lrw-r--r--  1 root root      50 2025-08-06 22:12 bugreports -> /data/user_de/0/com.android.shell/files/bugreports
drwxrwx---  2 system cache  4096 2025-08-06 22:12 cache
drwxr-xr-x  3 root root      0 2025-10-11 16:26 config
lrw-r--r--  1 root root      17 2025-08-06 22:12 d -> /sys/kernel/debug
drwxrwx--- 51 system system  4096 2025-10-11 16:30 data
d????????  ? ? ? ? ? ? ? data_mirror
drwxr-xr-x  2 root root    4096 2025-08-06 22:12 debug_ramdisk
drwxr-xr-x 19 root root    2760 2025-10-11 16:27 dev
lrw-r--r--  1 root root      11 2025-08-06 22:12 etc -> /system/etc
l????????  ? ? ? ? ? ? ? init -> ?
-????????  ? ? ? ? ? ? ? init.environ.rc
d????????  ? ? ? ? ? ? ? linkerconfig
drwx----- 2 root root   16384 2025-08-06 22:26 lost+found

```

Figura 3 – Criação de arquivo e diretório.

```

1 | emu64xa:/ $ clear
emu64xa:/ $ touch /sdcard/arquivo.txt
emu64xa:/ $ ls -l /sdcard
lrw-r--r--  1 root root 21 2025-08-06 22:12 /sdcard -> /storage/self/primary
emu64xa:/ $ chmod 644 /sdcard/arquivo.txt
emu64xa:/ $

```

Figura 4 – Alteração de permissões com chmod.

```

emu64xa:/ $ touch /sdcard/pasta
emu64xa:/ $ chmod 755 /sdcard/pasta
emu64xa:/ $ chown shell:shell /sdcard/arquivo.txt

```

Figura 5 – Alteração de proprietário com chown.

```

1 | emu64xa:/ $ clear
emu64xa:/ $ ln -s /sdcard/arquivo.txt /sdcard/link_arquivo

```

Figura 6 – Tentativa de criação de link simbólico (Permission denied).

```
emul64xa:/ $ du -h /sdcard
0      /sdcard
emul64xa:/ $
```

Figura 7 – Verificação de espaço em disco (du -h /sdcard).

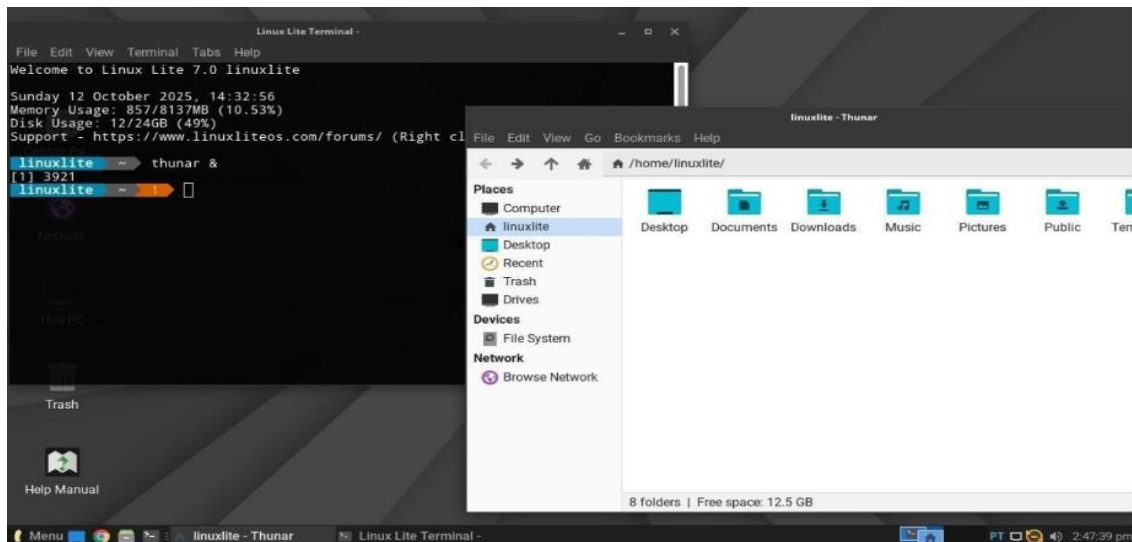
Observação: Os arquivos criados durante os testes (arquivo.txt e pasta) foram visualizados no explorador do emulador Android. A imagem do emulador (Figura 1) mostra claramente os itens `arquivo.txt` e `pasta` presentes no diretório /sdcard, comprovando que os arquivos foram criados com sucesso e estão acessíveis através da interface gráfica do dispositivo emulado.

3.3 Linux

Abertura do Gerenciador de Arquivos

O Linux Lite utiliza o **Thunar** como gerenciador gráfico de arquivos. Ele pode ser aberto de três formas:

- Pelo menu: **Menu** → **Acessórios** → **Gerenciador de Arquivos**
- Pelo atalho: **Super (tecla Windows) + E**
- Pelo Terminal

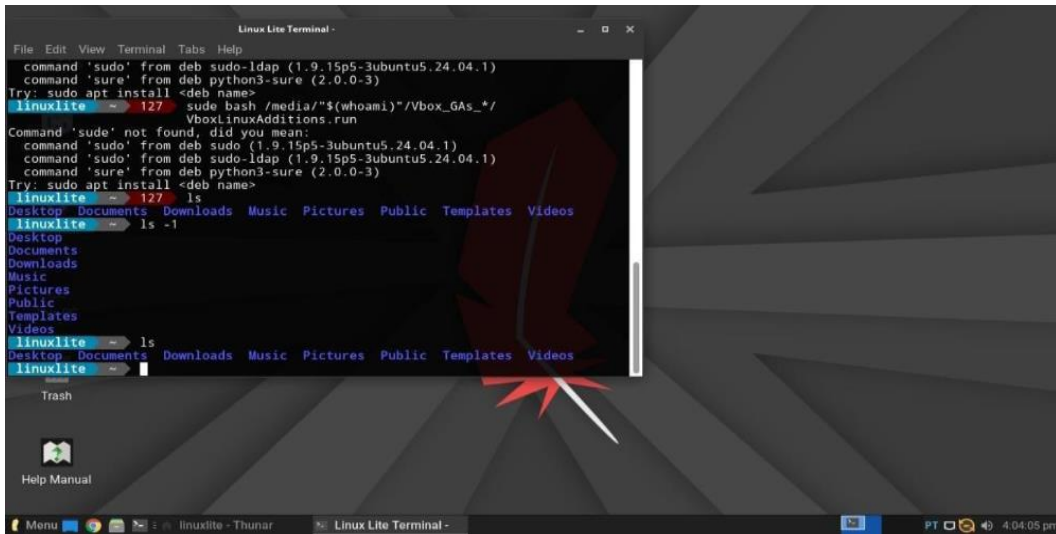


Comandos:

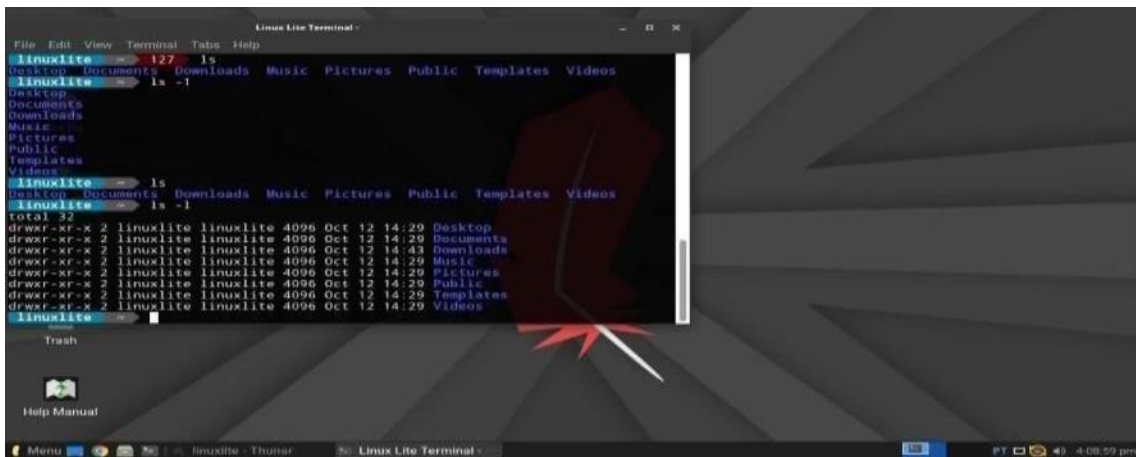
Os principais comandos usados no terminal para manipular arquivos e diretórios foram:

- `ls` Lista arquivos e pastas do diretório atual
- `ls -l` Exibe lista detalhada com permissões e datas
- `df -h` Mostra o uso do disco (tamanho e espaço livre)
- `neofetch` Mostra informações do sistema (se instalado)
- `ls` Lista arquivos e pastas do diretório atual

- **Ls**

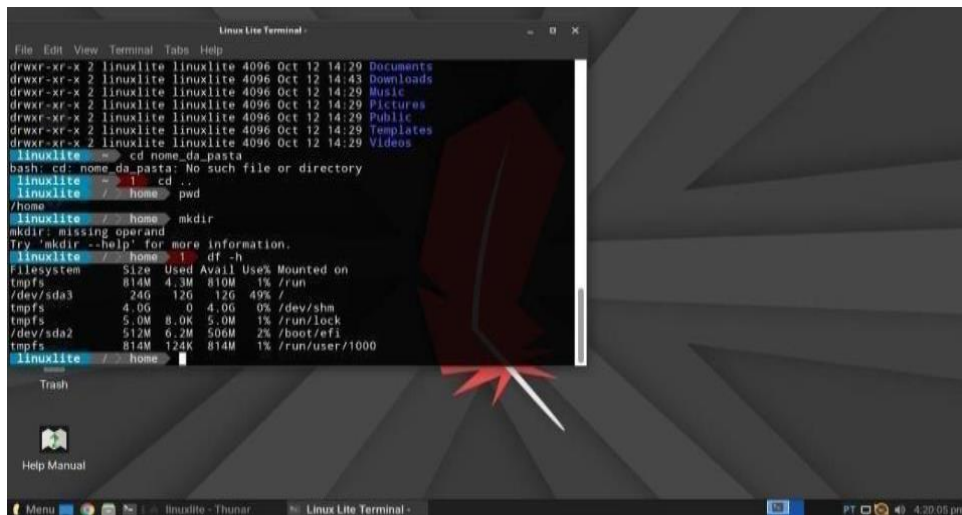


Executando o ls mostra os arquivos presentes na pasta do usuário, como Documents, Downloads, Music, Pictures e outros. Esse comando é usado para visualizar o conteúdo rapidamente, sem mostrar detalhes como permissões ou datas. ls -l Exibe lista detalhada com permissões e datas (*Tabela com permissões e acessos*)



O formato detalhado é útil para verificar quem pode ler, escrever ou executar cada arquivo.

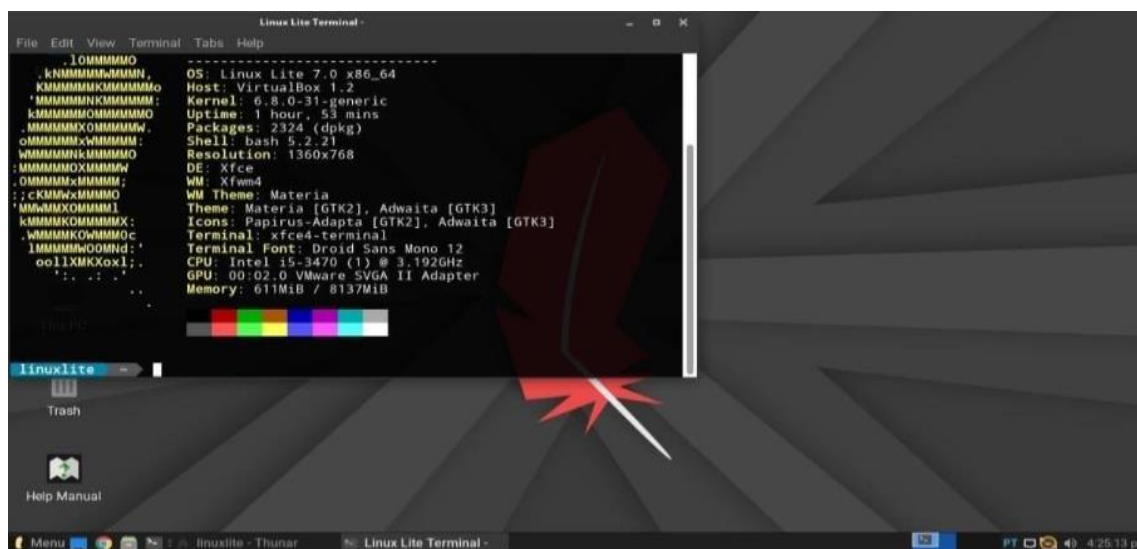
- **df -h**



O comando `df -h` foi utilizado para mostrar o uso do disco, ou seja, quanto espaço está ocupado e quanto está livre em cada partição. Na imagem, o terminal exibe uma tabela com as colunas:

Tipo	Função
Filesystem	Sistema de arquivo
Size	Tamanho total da partição
Used	espaço utilizado
Avail	espaço disponível
Use%	porcentagem de uso
Mounted on	ponto de montagem no sistema

- Neofetch



O comando neofetch foi executado para exibir informações detalhadas do sistema. Esse comando precisa estar instalado previamente. Na imagem, o terminal mostra um resumo com dados do sistema, como:

Nome e versão do Linux Lite
Kernel utilizado
Tempo de atividade (uptime)
Número de pacotes instalados
Ambiente gráfico e tema
Resolução da tela
Modelo do processador e da GPU
Uso de memória.

Além disso, o neofetch exibe o logotipo do sistema em ASCII no lado esquerdo, deixando o terminal visualmente informativo e organizado.

4. Análise Crítica

- **Windows (André):** O ICACLS demonstrou ser uma ferramenta ótima para gerenciamento de ACLs, permitindo controle de permissões por usuário específico. O FSUtil mostrou-se eficaz para operações de baixo nível com o sistema de arquivos NTFS. A abordagem de ambiente isolado provou ser segura para testes de permissões sem riscos ao sistema principal, mostrando uma forma de se adaptar a essas dificuldades.
- **Android (Eduardo):** Os experimentos realizados permitiram compreender profundamente o funcionamento do gerenciamento de arquivos e permissões no Android. Por meio dos comandos ls, chmod, chown, du e ln, foi possível identificar como o sistema lida com criação, modificação, análise e proteção de arquivos.

O Android demonstrou ser um ambiente altamente seguro, impondo restrições rigorosas ao acesso de áreas críticas do sistema e impedindo a criação de links simbólicos ou alterações indevidas por usuários não privilegiados. Essa estrutura reforça o modelo de isolamento por aplicativo (UIDs únicos), que garante integridade, estabilidade e proteção contra falhas e acessos indevidos. Em síntese, o estudo evidenciou como o Android mantém o equilíbrio entre funcionalidade e segurança, aplicando conceitos do Linux de forma adaptada a um ambiente móvel, e como as permissões e políticas de acesso são fundamentais para sua robustez operacional.

- **Linux (Vinicius):** Durante o uso do Linux Lite no VirtualBox, foi possível aprender na prática como o sistema gerencia arquivos por meio de comandos simples no terminal. A atividade ajudou a compreender a criação, movimentação e exclusão de arquivos e pastas, além de reforçar o uso do gerenciador gráfico Thunar.

5. Conclusão

Como conclusão o modulo do relatório se aprofunda ainda mais na utilização das linhas de terminais e na troca direta com o sistema operacional aprimorando conhecimentos adquiridos em módulos anteriores. A etapa do projeto também se

aprofunda no tema de segurança, focando principalmente nas permissões de acessos aos dados e informações do dispositivo.

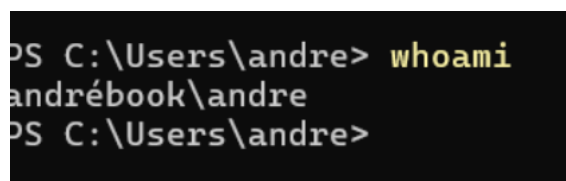
Outro foco é o controle e manipulação de arquivos, nos mostrando como podemos utilizar as ferramentas e características dos sistemas operacionais para uma maior autonomia e manuseio dos nossos arquivos, desde criar pastas até manipular o tamanho que queremos que um arquivo possua.

Em conclusão o modulo cumpriu em nos mostrar como essas duas etapas são importantes para o funcionamento de um dispositivo, por representarem bases que estão conectadas com os usuários: A manipulação de arquivos que é feita constantemente por qualquer usuário e as permissões que é algo de extrema importância para nós que utilizamos nossos aparelhos e desejamos conhecer o que podemos e não podemos fazer.

6. Autoavaliação

Dificuldades encontradas

André (Windows): Dificuldades: A principal dificuldade que enfrentei foram uma série de erros constantes na fase de permissões de ICACLS pois o código: “**icaccls teste1.txt /grant**” necessita do nome cadastrado do aparelho que no meu caso está como “andrébook”. Para resolver isso realizei uma pesquisa e descobri que o comando “whoami” realiza justamente essa ação: Mostra o nome do dispositivo.



```
PS C:\Users\andre> whoami
andrébook\andre
PS C:\Users\andre>
```

Eduardo (Android): Durante os experimentos, algumas dificuldades foram observadas, principalmente relacionadas às restrições de permissões do Android. Ao tentar criar um link simbólico com `ln -s`, o sistema retornou “Permission denied”, demonstrando que o Android limita operações desse tipo em `/sdcard`, mesmo para o usuário shell.

Outra dificuldade foi a tentativa de alterar permissões em diretórios do sistema (/data/data/), que retornou “Read-only file system” ou “No such file or directory”, já que essas áreas são protegidas e só podem ser manipuladas com acesso root. Esses comportamentos refletem a forte política de isolamento do Android, baseada em segurança por usuário (UID), impedindo que aplicativos ou usuários não autorizados alterem dados sensíveis.

Vinicius (Linux): As principais dificuldades foram a integração com o VirtualBox (instalação das Guest Additions e acesso à pasta compartilhada) e a adaptação inicial aos comandos do terminal. Mesmo assim, a experiência trouxe resultados positivos, ampliando o conhecimento sobre o sistema Linux e fortalecendo a autonomia no uso do ambiente virtual.

7. Referências

Linux Lite – Site Oficial: <https://www.linuxliteos.com/>

(Documentação e downloads oficiais do sistema Linux Lite, utilizado para o gerenciamento de arquivos.) Oracle VirtualBox – Manual Oficial:

<https://www.virtualbox.org/manual/>

(Guia oficial de uso do VirtualBox, incluindo configuração de máquinas virtuais e pastas compartilhadas.) Ubuntu Documentation

(Base do Linux Lite): <https://help.ubuntu.com/>

(Documentação oficial do Ubuntu, sistema no qual o Linux Lite é baseado, com explicações sobre comandos e terminal.)

GNU/Linux Command Manual – GNU Project:

<https://www.gnu.org/software/coreutils/manual/> (Manual oficial dos comandos básicos do Linux, como ls, cd, mkdir e rm.)

- [Android Open Source Project \(AOSP\) — Android File System Overview](#)
- [Linux Manual Pages — chmod\(1\), chown\(1\), ln\(1\), du\(1\)](#)
- [Documentação oficial do ADB — developer.android.com/studio/command-line/adb](#)- Videoaula: Cadê o Disco 'C:\' do Linux? Entenda a Estrutura de Pastas do Sistema — YouTube, Canal Diolinux.
- Videoaula: Explorador de Arquivos do Windows 11 — Mauricio Casseiro.