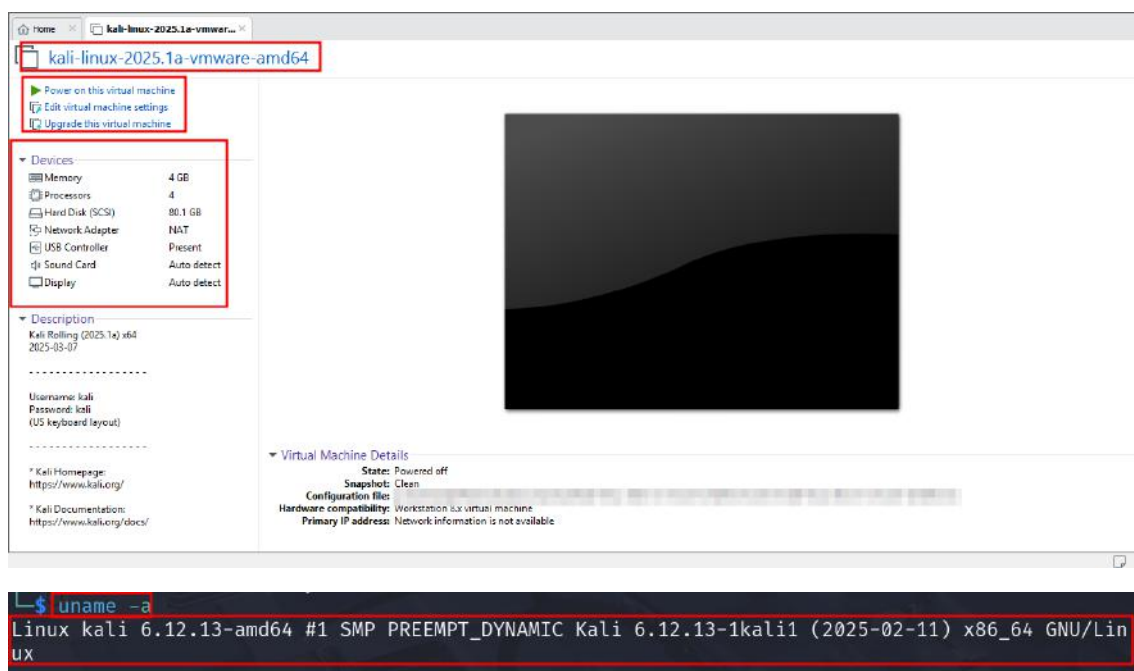


# Relatório de Atividades em Ambiente Kali Linux

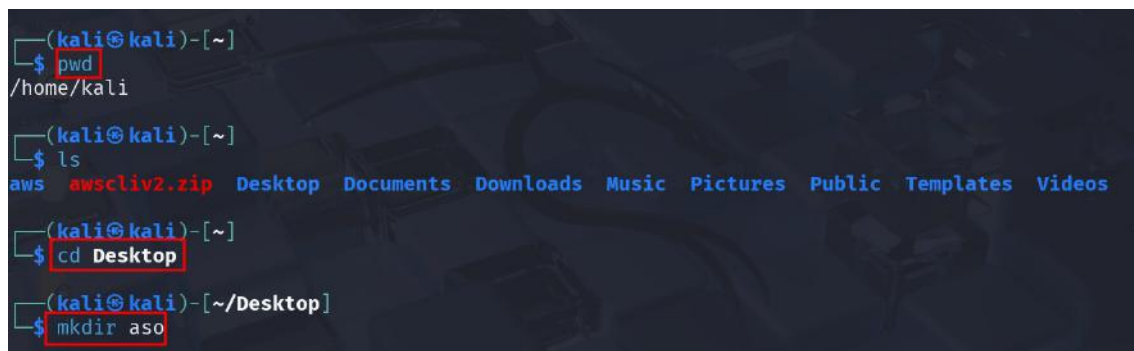
## Instalação

Este relatório detalha uma série de procedimentos realizados no sistema operacional Kali Linux, versão 2025.1, executado em uma máquina virtual com 4GB de memória RAM e um disco de 81GB. As atividades incluem a manipulação de arquivos e diretórios, edição de texto e gerenciamento de processos.



## Comandos básicos

Inicialmente, foram executados comandos básicos para a manipulação de diretórios e arquivos. No diretório Desktop, um novo diretório chamado aso foi criado com o comando mkdir aso. Em seguida, o diretório aso foi acessado.



## Arquivos e Diretórios

Dentro do diretório aso, o comando `echo "acaba 2025.1 pelo o amor de deus" >> pratica.txt` foi utilizado para criar o arquivo `pratica.txt` e inserir um texto em seu conteúdo. O conteúdo do arquivo foi verificado com o comando `cat pratica.txt`. Por fim, o arquivo foi removido com o comando `rm pratica.txt`.

```
(kali@kali)-[~/Desktop]
$ cd aso

(kali@kali)-[~/Desktop/aso]
$ echo "acaba 2025.1 pelo o amor de deus" >> pratica.txt

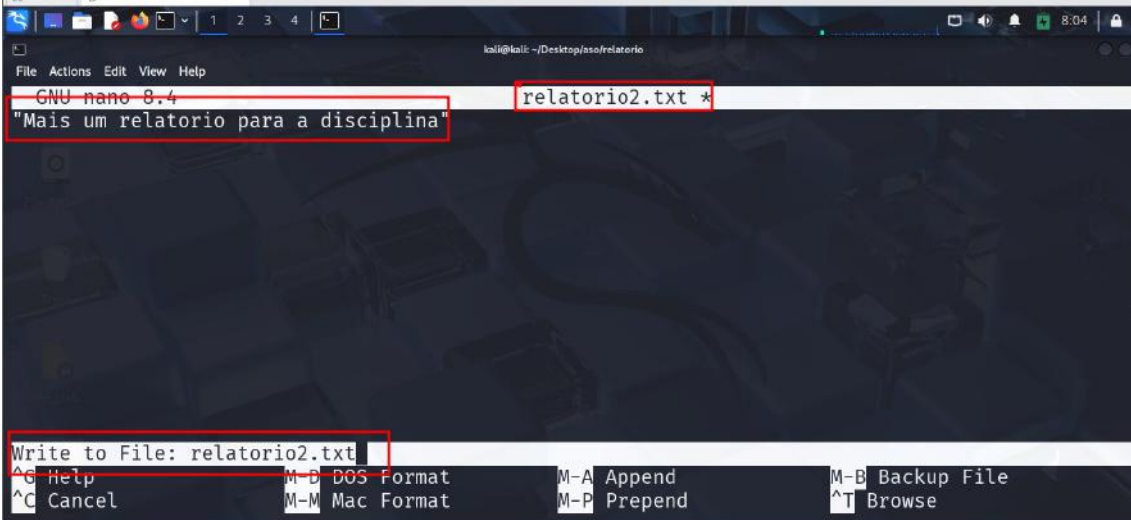
(kali@kali)-[~/Desktop/aso]
$ cat pratica.txt
acaba 2025.1 pelo o amor de deus

(kali@kali)-[~/Desktop/aso]
$ rm pratica.txt
```

## Edição de texto:

Para a edição de arquivos de texto, foi utilizado o editor nano. Um arquivo chamado `relatorio2.txt` foi criado e aberto com o comando `nano relatorio2.txt`. Dentro do editor, a frase "Mais um relatorio para a disciplina" foi inserida.

```
(kali@kali)-[~/Desktop/aso/relatorio]
$ nano relatorio2.txt
```



The screenshot shows the nano text editor interface. The title bar indicates the file is `relatorio2.txt *`. The editor content shows the text `"Mais um relatorio para a disciplina"` on the first line. The status bar at the bottom shows the command `Write to File: relatorio2.txt` and various keyboard shortcuts for file operations.

## Verificar conteúdo de texto

A verificação do conteúdo do arquivo foi realizada com o comando `cat relatorio2.txt`, que exibiu a frase inserida. Adicionalmente, os comandos `head /etc/passwd` e `tail /etc/passwd` foram usados para inspecionar as primeiras e as últimas linhas do arquivo de usuários do sistema, respectivamente.

```
(kali@kali)-[~/Desktop/aso/relatorio]
$ cat relatorio2.txt
"Mais um relatorio para a disciplina"

(kali@kali)-[~/Desktop/aso/relatorio]
$ head /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

```
(kali@kali)-[~/Desktop/aso/relatorio]
$ tail /etc/passwd
_gophish:x:123:129::/var/lib/gophish:/usr/sbin/nologin
iodine:x:124:65534::/run/iodine:/usr/sbin/nologin
miredo:x:125:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:126:65534::/var/lib/nfs:/usr/sbin/nologin
redis:x:127:130::/var/lib/redis:/usr/sbin/nologin
postgres:x:128:131:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:129:132::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:130:133::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:131:135::/var/lib/openvas:/usr/sbin/nologin
kali:x:1000:1000::,/home/kali:/usr/bin/zsh
```



## Processos

O gerenciamento de processos foi iniciado com a listagem de todos os processos em execução utilizando o comando ps aux.

```
└─$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.3  0.3  23292 14056 ?        Ss   07:30   0:08 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    07:30   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    07:30   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   07:30   0:00 [kworker/R-rcu_gp]
root           5  0.0  0.0      0     0 ?        I<   07:30   0:00 [kworker/R-sync_wq]
root           6  0.0  0.0      0     0 ?        I<   07:30   0:00 [kworker/R-slub_flushwq]
root           7  0.0  0.0      0     0 ?        I<   07:30   0:00 [kworker/R-netns]
root           8  0.0  0.0      0     0 ?        I    07:30   0:00 [kworker/0:0-events]
root          11  0.0  0.0      0     0 ?        I    07:30   0:00 [kworker/u128:0-ipv6_addrco
root          12  0.0  0.0      0     0 ?        I<   07:30   0:00 [kworker/R-mm_percpu_wq]
root          13  0.0  0.0      0     0 ?        I    07:30   0:00 [rcu_tasks_kthread]
```

Para um monitoramento em tempo real dos recursos do sistema e dos processos, o comando top foi executado. A saída do top mostrou informações como o uso de CPU, memória total (3912.8 MiB) e memória em uso.

```
top - 08:14:01 up 43 min, 2 users, load average: 0.04, 0.03, 0.07
Tasks: 213 total, 1 running, 212 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 1.7 sy, 0.0 ni, 97.9 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
MiB Mem : 3912.8 total, 2449.9 free, 867.9 used, 827.8 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used, 3044.9 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU  %MEM    TIME+  COMMAND
 1056 root        20   0 382932 100396 57148 S   1.7   2.5   0:42.04 Xorg
21493 kali        20   0 10392   5680 3632 R   1.0   0.1   0:00.12 top
 1350 kali        20   0 1223792 135416 83640 S   0.7   3.4   0:19.51 xfwm4
 1410 kali        20   0 236464 60336 19560 S   0.7   1.5   0:14.76 wrapper-2.0
 2043 kali        20   0 725772 63108 48304 S   0.7   1.6   0:09.75 qterminal
   610 root        20   0 113176 9500 8220 S   0.3   0.2   0:21.09 vmtoolsd
 1412 kali        20   0 273852 28432 21332 S   0.3   0.7   0:12.30 wrapper-2.0
 1498 kali        20   0 152564 43500 30332 S   0.3   1.1   0:22.62 vmtoolsd
     1 root        20   0 23292 14056 10336 S   0.0   0.4   0:08.52 systemd
     2 root        20   0      0      0      0 S   0.0   0.0   0:00.05 kthreadd
     3 root        20   0      0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
```

Foi demonstrado como localizar um processo específico. O comando ps aux | grep "firefox" foi utilizado para encontrar o processo do navegador Firefox. Após identificar o PID (Process ID) do processo principal do Firefox como 22173, o comando kill -9 22173 foi executado para forçar o encerramento do processo.

```
└─$ ps aux | grep "firefox"
kali        22173 107  4.6 11018348 185036 pts/1  RL+  08:15   0:12 firefox-esr
kali        22308  9.3  1.0 214816 41632 pts/1   Sl+   08:15   0:00 /usr/lib/firefox-esr/firefox
-esr -contentproc -parentBuildID 20250127191809 -prefsLen 21633 -prefMapSize 247044 -appDir /us
r/lib/firefox-esr/browser {74797ed7-f6d6-42fc-b9f0-234859fe8783} 22173 true socket
kali        22377  0.0  0.0 6520 2292 pts/0    S+    08:15   0:00 grep --color=auto firefox

(kali@kali)~[~/Desktop/aso/relatorio]
└─$ kill -9 22173
```