

ANHANGUERA EDUCACIONAL

Curso: Engenharia de Software

Relatório de Atividades Práticas

Disciplina: Sistemas Operacionais

Aluno: Wesley Ferreira Barbosa Lima

Unidade/Polo: Anhanguera Limeira/SP

Limeira/SP

2026

ANHANGUERA EDUCACIONAL

Curso: Engenharia de Software

Disciplina: Sistemas Operacionais

Relatório de Atividades Práticas – Sistemas Operacionais

Aluno: Wesley Ferreira Barbosa Lima

RA: 202600961

Unidade/Polo: Anhanguera Limeira/SP

Semestre/Ano: 1º semestre / 2026

Limeira/SP

2026

Este relatório apresenta os resultados das atividades práticas da disciplina de Sistemas Operacionais. O objetivo é desenvolver a compreensão dos principais componentes dos sistemas operacionais Linux e Windows, com foco no kernel, shell e gerenciamento de arquivos.

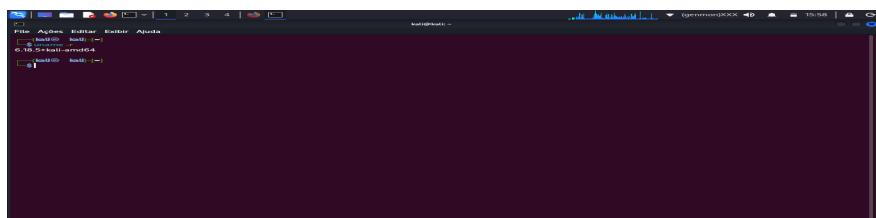
As atividades propostas permitem explorar o funcionamento do kernel e do shell, o gerenciamento de processos e permissões de arquivos, além da comparação entre as estruturas de diretórios dos sistemas Linux e Windows.

Com isso, busca-se consolidar o aprendizado sobre a organização e funcionamento dos sistemas operacionais, preparando o aluno para um gerenciamento mais eficaz desses ambientes.

Atividade 1 – Kernel e Shell (Linux)

Linux

- Comando: `uname -r`
- Resultado: `6.18.5+kali-amd64`
- Explicação: Esse comando mostra a versão do kernel em execução no Linux. O kernel é responsável por gerenciar processos e comunicação com o hardware.



- Comando: `ps -e`
- Explicação: O comando `ps -e` lista todos os processos ativos no sistema Linux.

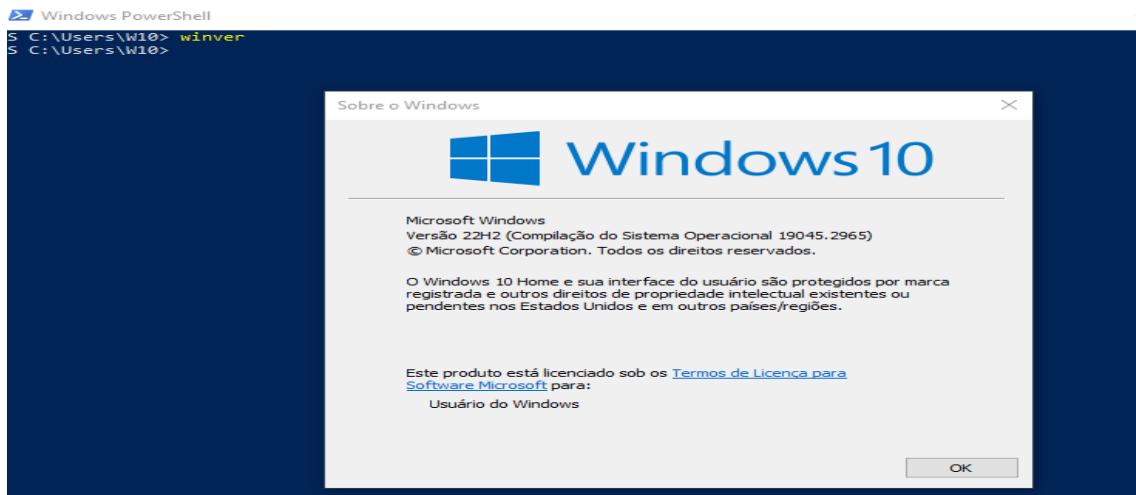


Windows

- Comando: `Get-Process`
- Explicação: O comando `Get-Process` lista todos os processos ativos no Windows. Cada processo possui um identificador (Id), consumo de memória e tempo de CPU. Isso demonstra como o **kernel do Windows** gerencia os recursos do sistema, garantindo que cada aplicação receba memória e tempo de processamento.

```
Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6
PS C:\Users\W10> Get-process
Handles  NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----  -----      -----      -----      -----      --  --  -----
    378      21        9512     30800      1,11     1664  1 ApplicationFrameHost
    267      14        4132     18584      0,25     1280  1 conhost
    311      14        1640      4756       424    0  csrss
    317      17        1772      5244       508    1  csrss
    422      16        3908     19568      1,53     2584  1 ctfmon
    255      22        4484     14516      0,48     1324  1 dllhost
    256      27        6548     16176      0,94     3672  1 dllhost
    951      48        36616    64264          920    1 dwm
```

- Comando: `winver`
- Explicação: O comando `winver` abre uma janela gráfica que mostra a versão do sistema operacional Windows em execução.



Atividade 2: Gerenciamento de Arquivos e Permissões

Linux

- Atividade realizada: Criar um diretório chamado `test_dir` e, dentro dele, criar um arquivo `test_file.txt`. Em seguida, atribuir permissões de leitura e escrita ao arquivo.

Pontos de atenção

- As permissões em sistemas Linux são representadas por três letras:
 - **R (read)** → leitura
 - **W (write)** → escrita
 - **X (execute)** → execução
- Essas permissões se aplicam a três grupos distintos
 - **Usuário (owner)**
 - **Grupo (group)**
 - **Outros (others)**
- Cada permissão tem um valor numérico:
 - **R = 4**
 - **W = 2**
 - **X = 1**
- Para definir as permissões, somamos os valores desejados para cada grupo.
 - Exemplo: **6 = 4 (leitura) + 2 (escrita) → rw-**

```

File Ações Editar Exibir Ajuda
[ kali@kali: ~/test_dir ]
$ mkdir test_dir
[ kali@kali: ~/test_dir ]
$ cd test_dir
[ kali@kali: ~/test_dir ]
$ touch test_file.txt
[ kali@kali: ~/test_dir ]
$ ls -la
total 8
drwxrwxr-x 2 kali kali 4096 fev 7 09:12 .
drwxr-xr-x 30 kali kali 4096 fev 7 09:11 ..
-rw-r--r-- 1 kali kali 0 fev 7 09:12 test_file.txt
[ kali@kali: ~/test_dir ]
$ chmod 600 test_file.txt
[ kali@kali: ~/test_dir ]
$ ls -la
total 8
drwxrwxr-x 2 kali kali 4096 fev 7 09:12 .
drwxr-xr-x 30 kali kali 4096 fev 7 09:11 ..
-rw-r--r-- 1 kali kali 0 fev 7 09:12 test_file.txt
[ kali@kali: ~/test_dir ]
$ 

```

Windows

- Criei uma pasta chamada **TestFolder**, acessei seu conteúdo e dentro dela criei um arquivo chamado **TestFile.txt**. Em seguida, atribuí ao usuário **W10** a permissão **F (Full Control)** sobre esse arquivo, garantindo acesso total às suas funcionalidades.

```

Windows PowerShell (x86)
PS C:\Users\W10> New-Item -ItemType Directory -Name TestFolder

Diretório: C:\Users\W10

Mode          LastWriteTime      Length Name
----          -----          ---- 
d-----       07/02/2026     10:16   TestFolder

PS C:\Users\W10> Set-Location TestFolder
PS C:\Users\W10\TestFolder> New-Item -ItemType File -Name TestFile.txt

Diretório: C:\Users\W10\TestFolder

Mode          LastWriteTime      Length Name
----          -----          ---- 
-a---       07/02/2026     10:17      0 TestFile.txt

PS C:\Users\W10\TestFolder> icacls TestFile.txt /grant W10:F
arquivo processado: TestFile.txt
Processados com sucesso 1 arquivos; falha no processamento de 0 arquivos
PS C:\Users\W10\TestFolder> icacls TestFile.txt
TestFile.txt WINDOWS10\W10:(F)
                  AUTORIDADE NT\SISTEMA:(I)(F)
                  BUILTIN\Administradores:(I)(F)
                  WINDOWS10\W10:(I)(F)

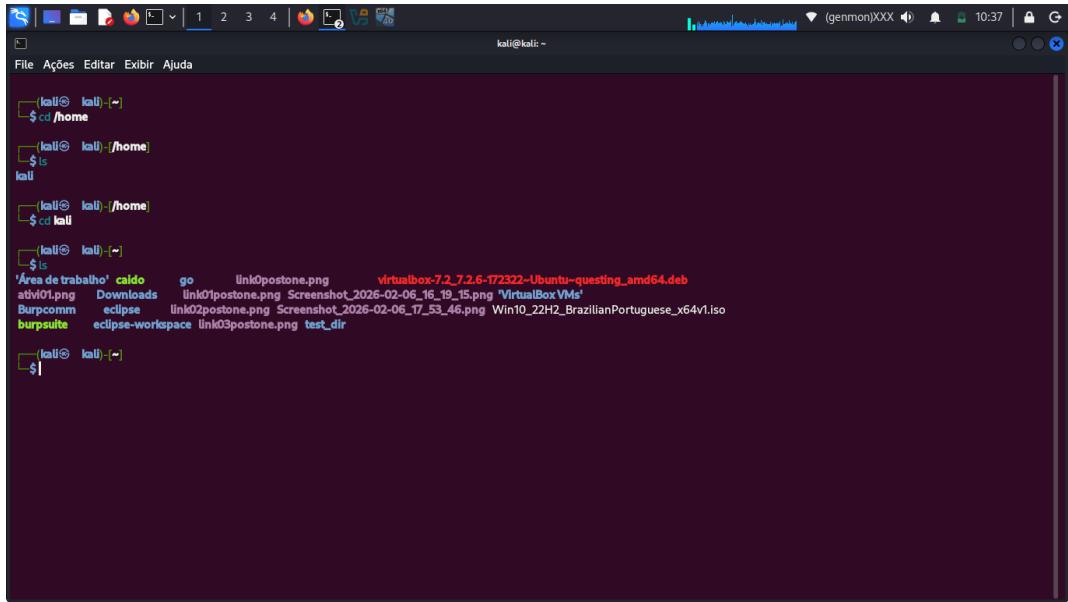
Processados com sucesso 1 arquivos; falha no processamento de 0 arquivos
PS C:\Users\W10\TestFolder>

```

Atividade 3: Comparando Estruturas de Diretórios

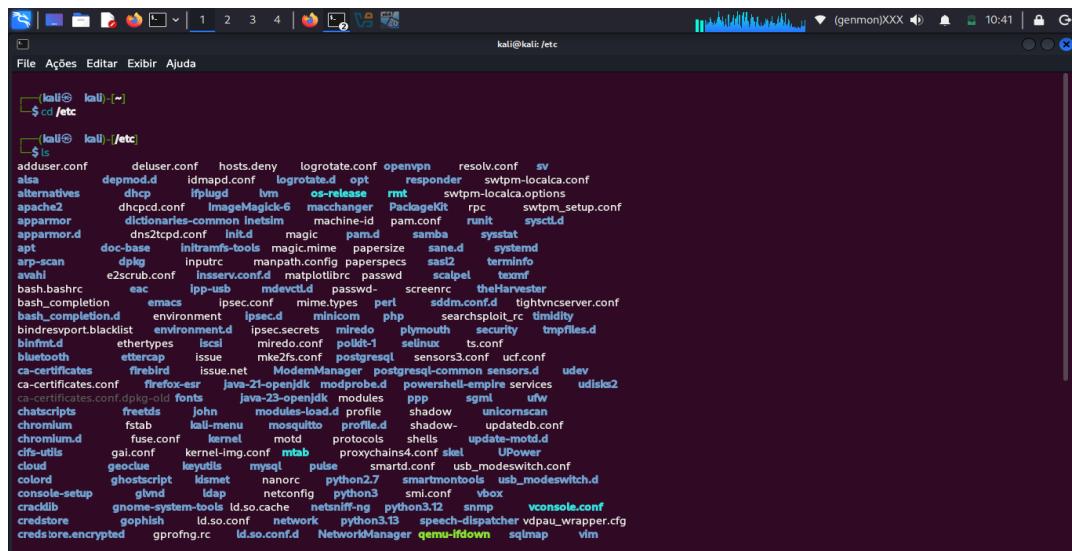
Linux

- Diretórios: /home
- Função: Armazenar as informações e arquivos pessoais de cada usuário comum do sistema.



```
(kali㉿ kali) [~]
$ cd /home
(kali㉿ kali) [/home]
$ ls
kali
(kali㉿ kali) [/home]
$ cd kali
(kali㉿ kali) [~]
$ ls
'Área de trabalho' calido  go  link0postone.png  virtualbox-7.2.7.2.6-172322-Ubuntu-questioning_amd64.deb
ativ01.png  Downloads  Link0postone.png  Screenshot_2026-02-06_16_19_15.png  'VirtualBox VMs'
Burpcomm  eclipse  Link02postone.png  Screenshot_2026-02-06_17_53_46.png  Win10_22H2_BrazilianPortuguese_x64v1.iso
burpsuite  eclipse-workspace  Link03postone.png  test_dlr
(kali㉿ kali) [~]
$
```

- Diretório: /etc
- Função: armazenar arquivos de configuração do sistema.



```
(kali㉿ kali) [~]
$ cd /etc
(kali㉿ kali) [/etc]
$ ls
adduser.conf  deluser.conf  hosts.deny  logrotate.conf  openvpn  resolv.conf  sv
alsa          depmod.d      idmapd.conf  logrotate.d  opt    responder  swtpm-localta.conf
alternatives  dhcpc        ifplugd     km            os-release  rm      swtpm-localta.options
apache2       dhpcd.conf   ImageMagick-6  machanger   PackageKit  rpc    swtpm-setup.conf
apparmor      dictionaries-common  instsim  machine-id  pam.conf   runit  sysctl.d
apparmor.d    dns2tcp.conf  initd      magic      pam.d    samza  sysstat
apt           docbook-dict-tools  magic.mime  papersize  sane.d  systemd
arp-scan      dplg         inutrc     manpath     papercspec  sanz2  terminfo
avahi        e2scrub.conf  insserv.conf.d  matpolibrc  passwd    scalpel  tomcat
bash.bashrc   esc          lpp-usb    mdevctl.d  password  screenrc  theHarvester
bash_completion  emacs      ipsec.conf  mime.types  perl    sddm.conf.d  tightvncserver.conf
bash_completion.d  environment  ipsec.d  minicom  php    searchsploit_rc  timidity
bindresport.blacklist  environment.d  ipsec.secrets  miredo  Plymouth  security  tmpfiles.d
bimfd.d       ethertap     issued     miredo.conf  polkit-1  selinux  ts.conf
bluetooth     ettercap    issued     mke2fs.conf  postgresql  sensors3.conf  ucconf
ca-certificates  firebird   issue.net  ModemManager  postgresql-common  sensors.d  udev
ca-certificates.conf  firefox-esr  java-21-openjdk  modprobe.d  powershell-empire  services  udisks2
ca-certificates.dpkgo-old  fonts    java-23-openjdk  modules  ppp    sgml    ufw
chatscripts   freetds    john      modules-load.d  profile  shadow  unicorncan
chromium      fstab      kall-menu  mosquitto  profile.d  shadow-  updatedb.conf
chromium.d    fuse.conf  kernel    motd      protocols  shells  update-motd.d
cifs-utils    gal.conf   kernel-lmg.conf  mtab  proxychains4.conf  siel  UPower
cloud         geoclue    keyutils  mysql    pulse  smartd.com  usb_modeswitch.conf
colord        ghostscript  kismet   nanorc  python2.7  smartmontools  usb_modeswitch.d
console-setup  glibmd    ldap      netconfig  python3  smi.conf  vbox
cracklib     gnome-system-tools  ld.so.cache  netsniffng  python3.12  snmp  vconsole.conf
credstore    gophish    ld.so.conf  network  python3.13  speech-dispatcher  vdpau_wrapper.cfg
credsstore.encrypted  gprofng.rc  ld.so.conf.d  NetworkManager  qemu-ifdown  sqlmap  vim
```

- Diretório: /var
- Função: serve para armazenar informações variáveis do sistema,

Windows

- Diretório: C:/
- Função: é o **ponto mais alto da hierarquia de arquivos** no Windows, equivalente à raiz do sistema de arquivos.

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

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\W10> Set-Location C:/
PS C:\> ls

Diretório: C:\

Mode           LastWriteTime      Length Name
----           -----          ---- 
d----
```

- Diretório: C:/ Users
- Função: Guarda os perfis e dados pessoais de cada usuário .

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

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

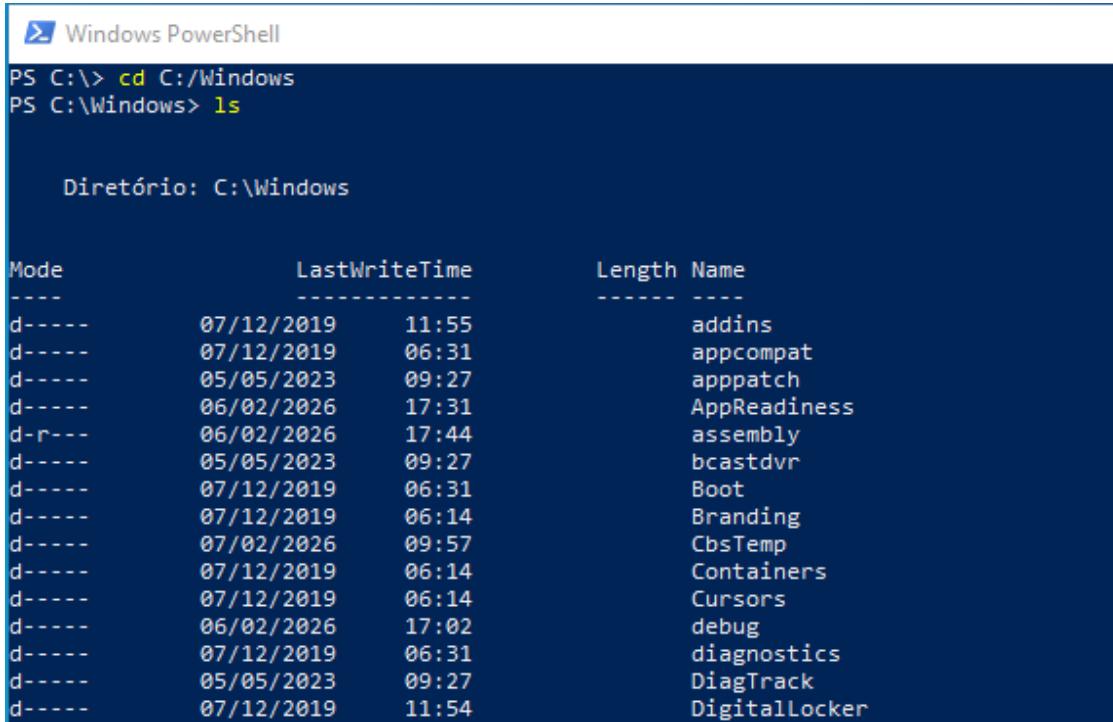
PS C:\Users\W10> cd  C:/
PS C:\> cd C:/Users
PS C:\Users> ls

Diretório: C:\Users

Mode           LastWriteTime      Length Name
----           -----          ---- 
d-r--d-r----- 06/02/2026       17:09      Public
d-----d----- 07/02/2026       10:16      W10

PS C:\Users>
```

- Diretório: C:/Windows
- Função: Onde ficam os **arquivos essenciais para o funcionamento do sistema**.



```
Windows PowerShell
PS C:\> cd C:/Windows
PS C:\Windows> ls

Diretório: C:\Windows

Mode                LastWriteTime       Length Name
----                -              ----
d----
```

- Diretório: "C:/Program Files"
- Função: Função **organizar e armazenar os programas instalados no Windows**.

Resultados de aprendizagem

As atividades realizadas permitiram compreender melhor a função do kernel e do shell, bem como a organização da estrutura de diretórios em Linux e Windows. A prática mostrou como os sistemas operacionais gerenciam processos, arquivos e permissões, consolidando a compreensão sobre segurança, eficiência e interação entre componentes fundamentais do sistema.