



CENTRO UNIVERSITARIO ANHANGUERA PITAGORAS UNOPAR DE CAMPO GRANDE

CURSO: ENGENHARIA DE SOFTWARE

DISCIPLINA: SISTEMAS OPERACIONAIS

ATIVIDADE PRÁTICA – UNIDADE 1, AULA 3

CARACTERÍSTICAS DOS SISTEMAS OPERACIONAIS

Aluno: Edmar Radanovis.

RA: 2025223493

Polo: Itapira / SP-UN944038

Ano 2025 / 2º Semestre

ANHANGUERA EDUCACIONAL
ENGENHARIA DE SOFTWARE

DISCIPLINA: Sistemas Operacionais

PROFESSOR: Romulo de Alameida Neves

ALUNO: Edmar Radanovis

RA: 2025223493

TÍTULO: Relatório da Aula Prática – Características dos Sistemas Operacionais
Unidade 1 – Aula 3

POLO: Itapira / SP-UN944038

Monte Sião, 24 de Agosto de 2025.

RESULTADOS DAS ATIVIDADES PRÁTICAS

Atividade 1 – Explorando o Kernel e Shell



Linux (via WSL - Ubuntu):

- 1 - Identificação da versão do kernel:

Comando executado: `uname -r`.

Resultado: O comando retornou a versão do kernel usada no WSL, "6.6.87.2-microsoft-standard-WSL2".

Explicação: Exibe a versão do kernel Linux, que é responsável por gerenciar processos, memória e dispositivos de entrada/saída no sistema operacional.

- 2 - Listagem de processos:

Comando executado: `ps -e`.

Resultado: O comando listou processos ativos, como init, bash, entre outros, com seus respectivos PIDs.

Explicação: Mostra todos os processos em execução, demonstrando como o kernel gerencia a execução concorrente de programas no Linux.

A screenshot of a terminal window titled 'Terminal'. The window shows the following command-line session:

```
/home/edmarradanovis/.zshrc:129: command not found: zinit
/home/edmarradanovis/.zshrc:151: command not found: zinit
/home/edmarradanovis/.zshrc:173: command not found: zinit
~> uname -r
6.6.87.2-microsoft-standard-WSL2
~> ps -e
 PID TTY      TIME CMD
  1 hvc0    00:00:00 init(Ubuntu-2@)
  6 hvc0    00:00:00 init
 7@ ?      00:00:00 SessionLeader
 71 ?      00:00:00 Relay(72)
 72 pts/1   00:00:00 zsh
 77 pts/1   00:00:00 zsh
 114 pts/1   00:00:00 zsh
 115 pts/1   00:00:00 gitstatusd-linu
 117 pts/1   00:00:00 zsh
 220 pts/1   00:00:00 ps
```

The terminal has a dark background with light-colored text. The window title bar is visible at the top.

=> Figura 1: Resultados dos comandos `uname -r` e `ps -e` no terminal WSL.



Windows (PowerShell):

- 1 - Listagem de processos:

Comando executado: `Get-Process` .

Resultado: O comando exibiu uma lista de processos ativos, como "explorer", "powershell", com detalhes como PID e uso de CPU.

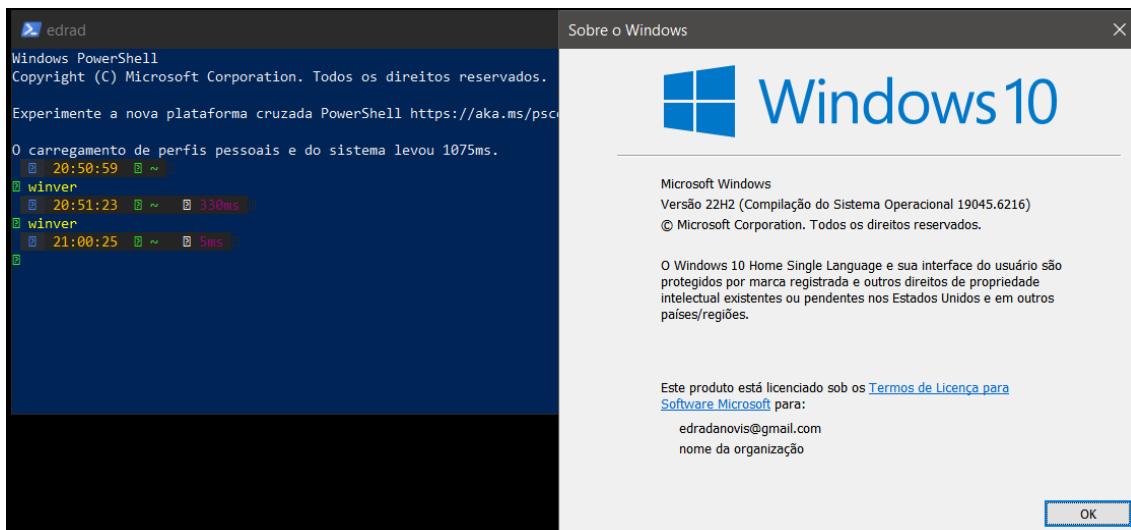
Explicação: Mostra os processos gerenciados pelo kernel do Windows, responsável por alocar recursos e gerenciar a execução.

- 2 - Identificação da versão do Windows:

Comando executado: `winver` .

Resultado: Uma janela pop-up exibiu a versão do Windows 10, " Versão 22H2, Compilação do Sistema Operacional 19045.6216 ".

Explicação: Mostra informações sobre a versão do sistema operacional Windows, que é gerenciada pelo kernel.



```

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

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

O carregamento de perfis pessoais e do sistema levou 4272ms.

20:01:01 ~ | Get-Process

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
----- -- -- -- -- -- -
391 22 12528 29548 2,02 9272 1 ApplicationFrameHost
197 12 7208 12508 46,91 12324 0 audiogd
567 37 26648 46340 4,23 10056 1 backgroundTaskHost
130 9 1740 6276 4804 0 BulletService
435 19 4612 12712 0,91 420 1 CastSrv
3601 118 248908 311644 598,36 1388 1 chrome
389 25 63488 100452 6,33 2668 1 chrome
485 58 653936 655468 847,78 3848 1 chrome
446 31 51180 75468 2,09 4408 1 chrome
409 94 107992 134192 565,27 5016 1 chrome
436 31 31412 57412 81,63 5476 1 chrome
512 63 1098956 1127200 724,95 6556 1 chrome
321 20 13352 23940 21,95 7052 1 chrome
281 22 25644 60436 1,00 7608 1 chrome
224 16 14532 24632 3,73 7664 1 chrome
286 32 89476 124056 8,64 7720 1 chrome
280 11 6972 8932 0,14 7816 1 chrome
496 23 15676 28848 0,61 8516 1 chrome
271 21 23940 54484 4,17 9288 1 chrome
263 23 32004 61624 1,47 9572 1 chrome
561 68 567404 633512 482,11 10708 1 chrome
304 23 34152 75904 1,69 12272 1 chrome
285 22 31392 56720 0,83 12440 1 chrome
262 23 35100 46784 1,72 12932 1 chrome
364 33 134308 134476 30,30 13904 1 chrome
201 20 67400 37440 0,59 2980 1 Code
439 69 342780 94856 153,66 7264 1 Code
622 24 100364 76312 12,23 7916 1 Code
260 31 98808 56428 1,97 8284 1 Code
237 26 84816 55832 5,88 8764 1 Code
241 11 10768 29252 0,16 9076 1 Code
183 20 32900 37856 0,50 9716 1 Code
284 28 135844 123240 38,11 9780 1 Code
1420 68 133536 90760 33,45 10336 1 Code
180 18 31156 65636 0,36 11076 1 Code
537 41 215308 189852 48,20 11408 1 Code
343 18 16296 47552 2,70 12160 1 Code
163 10 7268 7844 0,16 5812 1 CodeSetup-stable-6f176...
309 22 24524 23588 20,06 11796 1 CodeSetup-stable-6f176...
195 11 2168 1940 0,14 7888 1 ColorEngine
133 10 6604 7120 2212 0 conhost
108 7 2820 6120 0,02 3876 1 conhost

```

=> Figuras 2 e 3: Janela do comando `winver` no Windows e resultado do comando `Get-Process` no PowerShell .

Atividade 2 – Gerenciamento de Arquivos e Permissões



Linux (via WSL - Ubuntu):

- 1 - Criação de diretório e arquivo:

Comandos executados: `mkdir test_dir` , `cd test_dir` , `touch test_file.txt` .

Resultado: O diretório `test_dir` e o arquivo `test_file.txt` foram criados com sucesso.

Explicação: O comando `mkdir` cria diretórios, e `touch` cria arquivos vazios, demonstrando a organização do sistema de arquivos Linux.

- 2 - Definição de permissões:

Comando executado: `chmod 600 test_file.txt` .

Resultado: As permissões foram alteradas para " `rw----- 1 edmarradanovis edmarradanovis 0 Aug 25 08:34 test_file.txt` ", permitindo apenas leitura e escrita para o proprietário.

Explicação: O comando `chmod 600` restringe o acesso ao arquivo, garantindo segurança no sistema de arquivos Linux.

The screenshot shows a terminal window with the following session:

```
~/test_dir
> mkdir test_dir
> cd test_dir
~/test_dir
> touch test_file.txt
~/test_dir
> ls
test_file.txt
~/test_dir
> chmod 600 test_file.txt
~/test_dir
> ls -l
total 0
-rw----- 1 edmarradanovis edmarradanovis 0 Aug 25 08:34 test_file.txt
~/test_dir
>
```

=> Figura 4: Resultado dos comandos, `ls` mostrando a criação do diretório e do arquivo solicitados e `ls -l` mostrando as permissões de `test_file.txt`.



Windows (Explorador de Arquivos / PowerShell):

- 1 - Criação de pasta e arquivo:

Ação realizada: No Explorador de Arquivos, criei a pasta **TestFolder** e o arquivo **TestFile.txt** dentro dela..

Resultado: A pasta e o arquivo foram criados com sucesso.

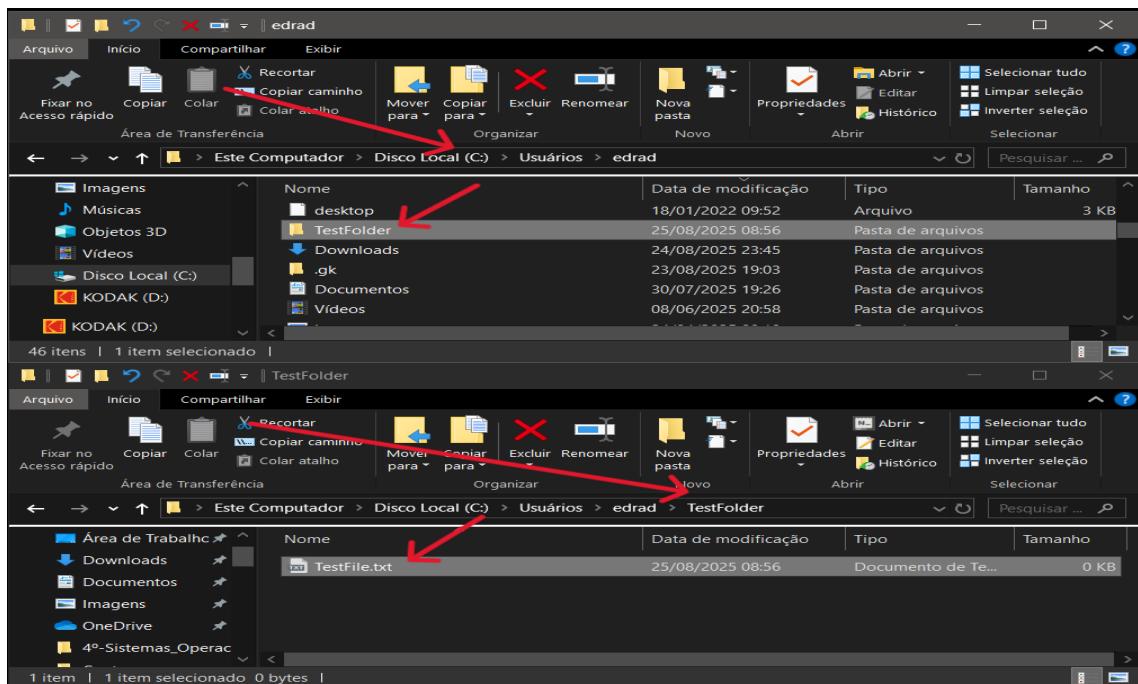
Explicação: O sistema de arquivos NTFS do Windows organiza arquivos em pastas, acessíveis via interface gráfica.

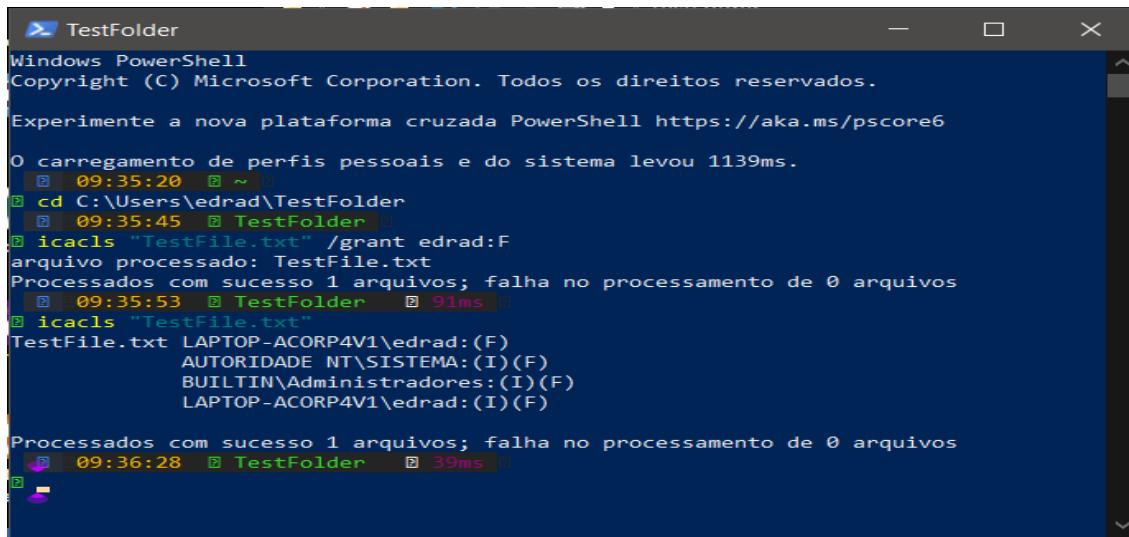
- 2 - Alteração de permissões:

Comando executado: `icacls TestFile.txt /grant %username%:F` .

Resultado: O comando concedeu permissões totais ao usuário atual.

Explicação: Gerencia permissões no Windows, garantindo que apenas o proprietário tenha acesso total.





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

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

O carregamento de perfis pessoais e do sistema levou 1139ms.
  09:35:20 ~
cd C:\Users\edrad\TestFolder
  09:35:45 TestFolder
icacls "TestFile.txt" /grant edrad:F
arquivo processado: TestFile.txt
Processados com sucesso 1 arquivos; falha no processamento de 0 arquivos
  09:35:53 TestFolder 91ms
icacls "TestFile.txt"
TestFile.txt LAPTOP-ACORP4V1\edrad:(F)
    AUTORIDADE NT\SISTEMA:(I)(F)
    BUILTIN\Administradores:(I)(F)
    LAPTOP-ACORP4V1\edrad:(I)(F)

Processados com sucesso 1 arquivos; falha no processamento de 0 arquivos
  09:36:28 TestFolder 39ms
```

=> Figuras 5 e 6: Resultado da criação da pasta `TestFolder` com o arquivo `TestFile.txt` no Explorador de Arquivos e do comando `icacls TestFile.txt` no PowerShell.

Atividade 3 – Comparando Estruturas de Diretórios



Linux (via WSL - Ubuntu):

- 1 - Exploração da estrutura de diretórios:

Comandos executados: `ls /` , `cd /home` , `cd /etc` , `cd /var` , `ls`.

Resultado: O comando `ls /` listou diretórios como `bin` , `dev` , `etc` , `home` , `var` , `etc`.

Explicação: A estrutura de diretórios Linux começa na raiz (`/`). O diretório `/home` armazena arquivos de usuários, `/etc` contém configurações do sistema, e `/var` guarda dados variáveis como logs.

```
/home/edmarradanovis/.zshrc:129: command not found: zinit
/home/edmarradanovis/.zshrc:151: command not found: zinit
/home/edmarradanovis/.zshrc:173: command not found: zinit
~
> ls /
Docker dev init lib64 media proc sbin sys var
bin etc lib libx32 mnt root snap tmp
boot home lib32 lost+found opt run srv usr

~
> cd /home
e /home
> ls
edmarradanovis

e /home
> cd /etc
e /etc
> ls
ModemManager
NetworkManager
PackageKit
X11
adduser.conf
alternatives
apparmor
apparmor.d
aport
apt
at.deny
bash.bashrc
bash_completion
bindresvport.blacklist
brinfmt.d
byobu
ca-certificates
ca-certificates.conf
ca-certificates.conf.dpkg-old
calendar
cloud
console-setup
cron.d
cron.daily
cron.hourly
cron.monthly
cron.weekly
crontab
cryptsetup-initramfs
crypttab
dbus-1
dconf
debcnf.conf
debian_version
default
deluser.conf
dpmode.d
dhcp
dpkg
e2scrub.conf
ec2_version
environment
ethertypes
fonts
fstab
fuse.conf
twpd
gai.conf
groff
group
group-
uid
gshadow
gshadow-
e
hdparm.conf
host.conf
hostname
hosts
hosts.allow
hosts.deny
init.d

e /etc
> cd /var
e /var
> ls
backups cache crash lib local lock log mail opt run snap spool tmp
e /var
> 
```

=> Figura 7: Resultado do comando `ls` no terminal Ubuntu.



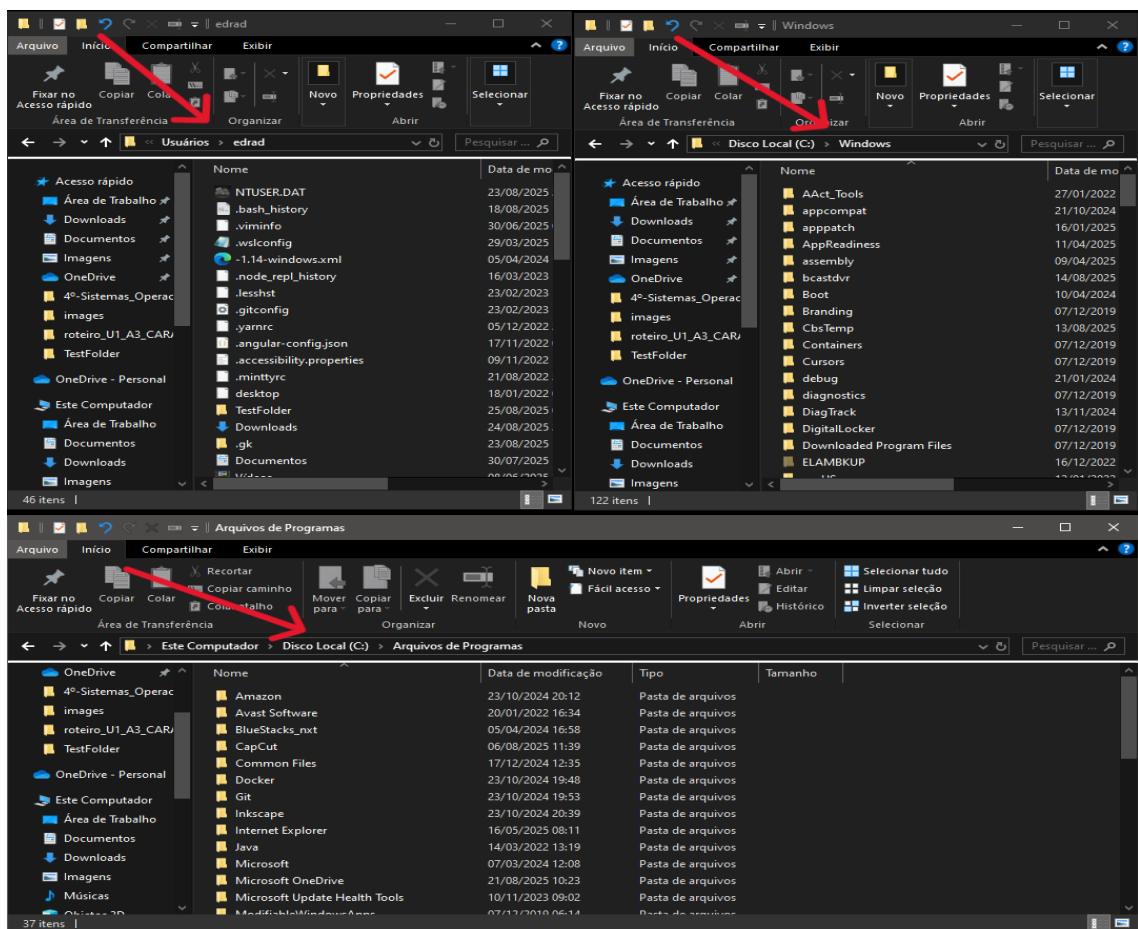
Windows (Explorador de Arquivos):

- 1 - Exploração da estrutura de diretórios:

Ação realizada: Naveguei pelo Explorador de Arquivos até C:\ e explorei C:\Users, C:\Windows, C:\Program Files.

Resultado: Identifiquei que C:\Users armazena dados de usuários, C:\Windows contém arquivos do sistema, e C:\Program Files guarda programas instalados.

Explicação: O sistema de arquivos NTFS do Windows organiza dados em uma estrutura hierárquica, com funções semelhantes às do Linux, mas com nomenclatura diferente.



=> Figura 8: Captura do Explorador de Arquivos mostrando as estruturas de C:\ e explorando C:\Users, C:\Windows, C:\Program Files .

REFERÊNCIAS

MACHADO, F. B.; MAIA, L. L. Sistemas Operacionais: Conceitos e Mecanismos. 7. ed. Rio de Janeiro: LTC, 2013.

MICROSOFT. Documentação do PowerShell. Disponível em:
<https://learn.microsoft.com/powershell/>. Acesso em: 23 ago. 2025.

CANONICAL. Documentação Ubuntu. Disponível em: <https://ubuntu.com/>. Acesso em: 23 ago. 2025.