

Trabalho Prático – Sistemas Operacionais

Tema: Conceitos Básicos, Gerenciamento de Processos e Sistema de Arquivos no Linux

Nome: André Klipp

Disciplina: Sistemas Operacionais

Professor: Alessandro João Brassanini

Data de Entrega: 18 de setembro de 2025

Atividade 01 – Conceitos Básicos – Reconhecimento do Sistema

1. Nome do kernel e versão do sistema operacional (uname -r, uname -a).

```
[AndreKlipp@archiso ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
dev              1.4G   0    1.4G   0% /dev
run              1.5G   8.9M  1.5G   1% /run
/dev/sr0         1.2G   1.2G   0 100% /run/archiso/bootmnt
cowspace        256M   1.1M  255M   1% /run/archiso/cowspace
/dev/loop0       821M   821M   0 100% /run/archiso/airootfs
airootfs         256M   1.1M  255M   1% /
tmpfs            1.5G   0    1.5G   0% /dev/shm
tmpfs            1.5G   0    1.5G   0% /tmp
tmpfs            1.0M   0    1.0M   0% /run/credentials/systemd-journald.service
tmpfs            1.0M   0    1.0M   0% /run/credentials/systemd-resolved.service
tmpfs            1.5G   2.7M  1.5G   1% /etc/pacman.d/gnupg
tmpfs            1.0M   0    1.0M   0% /run/credentials/systemd-networkd.service
tmpfs            1.0M   0    1.0M   0% /run/credentials/getty@tty1.service
tmpfs           298M   8.0K  298M   1% /run/user/0
[AndreKlipp@archiso ~]# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0  7:0    0 820.6M  1 loop /run/archiso/airootfs
sr0    11:0    1   1.1G  0 rom  /run/archiso/bootmnt
[AndreKlipp@archiso ~]# _
```

2. Espaço em disco e partições (df -h, lsblk).

```
[AndreKlipp@archiso ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
dev              1.4G   0    1.4G   0% /dev
run              1.5G   8.9M  1.5G   1% /run
/dev/sr0         1.2G   1.2G   0 100% /run/archiso/bootmnt
cowspace        256M   1.1M  255M   1% /run/archiso/cowspace
/dev/loop0       821M   821M   0 100% /run/archiso/airootfs
airootfs         256M   1.1M  255M   1% /
tmpfs            1.5G   0    1.5G   0% /dev/shm
tmpfs            1.5G   0    1.5G   0% /tmp
tmpfs            1.0M   0    1.0M   0% /run/credentials/systemd-journald.service
tmpfs            1.0M   0    1.0M   0% /run/credentials/systemd-resolved.service
tmpfs            1.5G   2.7M  1.5G   1% /etc/pacman.d/gnupg
tmpfs            1.0M   0    1.0M   0% /run/credentials/systemd-networkd.service
tmpfs            1.0M   0    1.0M   0% /run/credentials/getty@tty1.service
tmpfs           298M   8.0K  298M   1% /run/user/0
[AndreKlipp@archiso ~]# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0  7:0    0 820.6M  1 loop /run/archiso/airootfs
sr0    11:0    1   1.1G  0 rom  /run/archiso/bootmnt
[AndreKlipp@archiso ~]# _
```

3. Usuário atual e grupos (whoami, groups).

```
[AndreKlipp@archiso ~]# whoami
AndreKlipp
[AndreKlipp@archiso ~]# groups
wheel AndreKlipp
[AndreKlipp@archiso ~]#
```

4. Criação da pasta trabalho_SO_AndreKlipp

```
[AndreKlipp@archiso ~]# mkdir ~/trabalho_SO_AndreKlipp
[AndreKlipp@archiso ~]# _
```

5. Função dos diretórios principais:

/bin – Contém binários essenciais para o sistema.

/etc – Arquivos de configuração do sistema.

/home – Diretório dos usuários.

/var – Dados variáveis, como logs e filas de impressão.

/dev – Arquivos de dispositivos.

Atividade 02 – Gerenciamento de Processos – Controle e Sinais

1. Execução de três programas simultaneamente.

```
[AndreKlipp@archiso ~]# gedit &
[1] 1064
[AndreKlipp@archiso ~]# -bash: gedit: command not found
top &
[2] 1071
[1] Exit 127          gedit
[AndreKlipp@archiso ~]# sleep 500 &
[3] 1079

[2]+ Stopped          top
[AndreKlipp@archiso ~]# _
```

2. Listagem de processos do usuário com ps -u.

```
[AndreKlipp@archiso ~]# ps -u $AndreKlipp
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
AndreKl+  1047  0.0  0.1   5492  4676 tty1    S   20:07   0:00 -bash
AndreKl+  1071  0.0  0.0   6572  3624 tty1    T   20:07   0:00 top
AndreKl+  1079  0.0  0.0   3088  2080 tty1    S   20:07   0:00 sleep 500
AndreKl+  1111 100.0  0.1   7172  4220 tty1    R+  20:08   0:00 ps -u
[AndreKlipp@archiso ~]# _
```

3. Encerramento de processo com kill -15 <PID>.

```
AndreKlipp@archiso ~]# ps -u $AndreKlipp
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
ndreKl+   1047  0.0  0.1   5492  4676 tty1    S   20:07   0:00 -bash
ndreKl+   1071  0.0  0.0   6572  3624 tty1    T   20:07   0:00 top
ndreKl+   1079  0.0  0.0   3088  2080 tty1    S   20:07   0:00 sleep 500
ndreKl+   1111 100.0  0.1   7172  4220 tty1    R+  20:08   0:00 ps -u
AndreKlipp@archiso ~]# kill -15 1111
bash: kill: (1111) - No such process
AndreKlipp@archiso ~]# kill -15 1079
AndreKlipp@archiso ~]# _
```

4. Pausa com kill -19 <PID> e continuação com kill -18 <PID>.

```
[AndreKlipp@archiso ~]# ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
AndreKl+  1047  0.0  0.1   5492  4676 tty1    S   20:07   0:00 -bash
AndreKl+  1236  0.1  0.0   5752  3740 tty1    T   20:11   0:00 nano
AndreKl+  1244  0.0  0.0   3088  1940 tty1    S   20:11   0:00 sleep 500
AndreKl+  1252  0.0  0.1   7172  4248 tty1    R+  20:11   0:00 ps -u
[AndreKlipp@archiso ~]# kill -19 1244
[AndreKlipp@archiso ~]# kill -18 1244

[3]+ Stopped          sleep 500
[AndreKlipp@archiso ~]# _
```

5. Alteração de prioridade com nice/renice.

```
[AndreKlipp@archiso ~]# renice +5 -p 1244
1244 (process ID) old priority 0, new priority 5
[AndreKlipp@archiso ~]#
```

Explicações:

- Sinais no Linux são formas de comunicação entre processos e o kernel, permitindo controlar sua execução.

- Processos em foreground são executados na linha de comando, bloqueando-a até terminarem. Já processos em background são executados em paralelo, permitindo continuar usando o terminal.

Atividade 03 – Sistema de Arquivos – Permissões e Organização

1. Estrutura de diretórios do projeto criada (projeto/docs/scripts/dados).
2. Criação de arquivos fictícios com touch/echo.

```
[AndreKlipp@archiso ~]$ mkdir -p projeto/docs projeto/scripts projeto/dados
[AndreKlipp@archiso ~]$ touch projeto/docs/arquivo1.txt
[AndreKlipp@archiso ~]$ echo "echo hello world" > projeto/scripts/script1.sh
[AndreKlipp@archiso ~]$ echo "dados privados" > projeto/dados/dados1.txt
[AndreKlipp@archiso ~]$ _
```

3. Ajuste de permissões com chmod.

```
[AndreKlipp@archiso ~]$ chmod 700 projeto/scripts/script1.sh
[AndreKlipp@archiso ~]$ chmod 644 projeto/docs/arquivo1.txt
[AndreKlipp@archiso ~]$ chmod 600 projeto/dados/dados1.txt
[AndreKlipp@archiso ~]$ _
```

4.

4. Criação de links simbólicos e físicos.

```
[AndreKlipp@archiso ~]$ ln projeto/docs/arquivo1.txt link_fisico.txt
[AndreKlipp@archiso ~]$ ln -s projeto/docs/arquivo1.txt link_simbolico.txt
[AndreKlipp@archiso ~]$ ls -li
total 0
139 -rw-r--r-- 2 AndreKlipp AndreKlipp 0 Sep 18 20:19 link_fisico.txt
210 lrwxrwxrwx 1 AndreKlipp AndreKlipp 25 Sep 18 20:25 link_simbolico.txt -> projeto/docs/arquivo1.txt
133 drwxr-xr-x 5 AndreKlipp AndreKlipp 100 Sep 18 20:19 projeto
[AndreKlipp@archiso ~]$
```

5. Compactação do projeto com tar.

```
[AndreKlipp@archiso ~]$ tar -czvf projeto.tar.gz projeto/
projeto/
projeto/docs/
projeto/docs/arquivo1.txt
projeto/scripts/
projeto/scripts/script1.sh
projeto/dados/
projeto/dados/dados1.txt
[AndreKlipp@archiso ~]$ ls -lh projeto.tar.gz
-rw-r--r-- 1 AndreKlipp AndreKlipp 298 Sep 18 20:25 projeto.tar.gz
[AndreKlipp@archiso ~]$ _
```

Respostas:

- As permissões rwx indicam: r (read) leitura, w (write) escrita, x (execute) execução.
- O dono, grupo e outros possuem níveis diferentes de acesso, definidos separadamente.
- Links físicos compartilham o mesmo inode do arquivo original, enquanto links simbólicos são atalhos que apontam para o caminho do arquivo.

Conclusão

Com este trabalho, foi possível compreender na prática conceitos fundamentais do Linux: reconhecimento do sistema, gerenciamento de processos e manipulação do sistema de arquivos. A execução dos comandos permitiu entender melhor como o kernel, usuários e permissões interagem para garantir o funcionamento e a segurança do sistema operacional.