

Lucas Mattos Canela  
Jhuan da Matta Juliao de Oliveira  
Endrews Elias de Oliveira Prado  
Pedro Henrique de Oliveira Vieira Martins  
Victor Griffó de Andrade Faria  
Robert Valadão da Silva

## **Relatório de Atividades: Roteiros 6 e 7**

### **Introdução ao Linux**

Vila Velha - ES  
Novembro de 2025

Lucas Mattos Canela  
Jhuan da Matta Juliao de Oliveira  
Endrews Elias de Oliveira Prado  
Pedro Henrique de Oliveira Vieira Martins  
Victor Griffó de Andrade Faria  
Robert Valadão da Silva

## **Relatório de Atividades: Roteiros 6 e 7**

### **Introdução ao Linux**

Universidade Vila Velha  
Ciência da Computação

Vila Velha - ES  
Novembro de 2025

# Sumário

|            |  |           |
|------------|--|-----------|
| <b>1</b>   | <b>ROTEIRO DE ATIVIDADES 6 . . . . .</b>                                   | <b>3</b>  |
| <b>1.1</b> | <b>Atividade 6.1 - Processos em Estado Específico . . . . .</b>            | <b>3</b>  |
| <b>1.2</b> | <b>Atividade 6.2 - Processos em Background . . . . .</b>                   | <b>3</b>  |
| <b>1.3</b> | <b>Atividade 6.3 - Daemon Cron (Backup) . . . . .</b>                      | <b>4</b>  |
| <b>1.4</b> | <b>Atividade 6.4 - Script de Informações do Usuário . . . . .</b>          | <b>5</b>  |
| <b>1.5</b> | <b>Atividade 6.6 - Leitura de Variáveis e Operações . . . . .</b>          | <b>6</b>  |
| <b>1.6</b> | <b>Atividade 6.7 - Parâmetros e Shift . . . . .</b>                        | <b>6</b>  |
| <b>1.7</b> | <b>Atividades 6.8 e 6.9 - Testes de Parâmetros e Diretórios . . . . .</b>  | <b>7</b>  |
| <b>1.8</b> | <b>Atividade 6.10 - Listando arquivos passados por parâmetro . . . . .</b> | <b>8</b>  |
| <br>       |  |           |
| <b>2</b>   | <b>ROTEIRO DE ATIVIDADES 7 . . . . .</b>                                   | <b>10</b> |
| <b>2.1</b> | <b>Atividade 7.6 - Uso de Grep na Agenda . . . . .</b>                     | <b>10</b> |
| <b>2.2</b> | <b>Atividade 7.9 - Expressões Regulares . . . . .</b>                      | <b>10</b> |
| <b>2.3</b> | <b>Atividade 7.11 - Menu de Opções . . . . .</b>                           | <b>11</b> |

# 1 Roteiro de Atividades 6

## 1.1 Atividade 6.1 - Processos em Estado Específico

Comando para listar apenas processos no estado Running (R).

**Comando:**

```
1 ps -eo pid,comm,state | grep "R"
```

Figura 1 – Execução do comando ps e grep

```
lucasmattoscanela@pop-os:~$ ps -eo pid,comm,state | grep "R"
4 kworker/R-rcu_g I
5 kworker/R-sync_I
6 kworker/R-kvfre I
7 kworker/R-slub_I
8 kworker/R-netsns I
13 kworker/R-mm_pe I
64 kworker/R-inet_I
72 kworker/R-write I
73 kworker/R-kblock I
78 kworker/R-blkcg I
79 kworker/R-kintte I
82 kworker/R-tpm_d I
83 kworker/R-ata_s I
84 kworker/R-md_I
85 kworker/R-md_b I
86 kworker/R-edac I
87 kworker/R-devfr I
95 kworker/R-kthrot I
96 kworker/R-acpi_I
99 kworker/R-mld_I
100 kworker/R-ipv6_I
115 kworker/R-kstrp I
128 kworker/R-charg I
191 kworker/R-scsi_I
193 kworker/R-scsi_I
195 kworker/R-scsi_I
197 kworker/R-scsi_I
232 kworker/R-raid5 I
389 kworker/R-kdmfl I
390 kworker/R-kcrys I
392 kworker/R-kcrys I
399 kworker/R-kdmfl I
434 kworker/R-ext4_I
730 kworker/R-kdmfl I
731 kworker/R-kcrys I
732 kworker/R-kcrys I
4282 brave R
4372 ps R
lucasmattoscanela@pop-os:~$
```

Fonte: O autor (2025)

## 1.2 Atividade 6.2 - Processos em Background

Execução do navegador em background e análise da árvore de processos.

**Comandos:**

```
1 firefox &
2 pstree -p
3 jobs
```

Figura 2 – Visualização do firefox & e de pstree -p

```
45:2 ps x
lucasmattoescanela@pop-os:~$ firefox &
[1] 4714
lucasmattoescanela@pop-os:~$ pstree -p
ps -f
system(1)─{ModemManager(927)}─{ModemManager}(939)
└─{ModemManager}(947)
NetworkManager(806)─{NetworkManager}(883)
└─{NetworkManager}(884)
accounts-daemon(807)─{accounts-daemon}(886)
└─{accounts-daemon}(888)
acpid(888)
avahi-daemon(810)─{avahi-daemon}(865)
chrony(956)─{chrony}(957)
colord(1773)─{colord}(1780)
└─{colord}(1782)
cron(813)
cups-browsed(1162)─{cups-browsed}(1211)
└─{cups-browsed}(1212)
cupsd(1036)
dbus-broker-lau(782)─{dbus-broker}(805)
fwupd(1868)─{fwupd}(1871)
└─{fwupd}(1876)
└─{fwupd}(1877)
└─{fwupd}(1879)
gdm3(1801)─{gdm-session-wor}(1830)─{gdm-x-session}(1881)─{xorg}(1883)─{xorg}(1892)
└─{gnome-session-b}(1893)─{gnome-session-b}(1957)
└─{gnome-session-b}(1958)
└─{gnome-session-b}(1958)
└─{gnome-session-b}(1882)
└─{gnome-session-b}(1893)
└─{gnome-session-b}(1893)
└─{gdm3}(1003)
└─{gdm3}(1004)
gnome-keyring-d(1873)─{gnome-keyring-d}(1874)
└─{gnome-keyring-d}(1875)
└─{gnome-keyring-d}(1960)
networkd-dispat(817)
nvidia-persiste(842)
packagekitd(1454)─{packagekitd}(1455)
└─{packagekitd}(1456)
polkitd(828)─{polkitd}(877)
└─{polkitd}(879)
pop-system-upda(1161)─{pop-system-upda}(1178)
└─{pop-system-upda}(1172)
└─{pop-system-upda}(1174)
└─{pop-system-upda}(1177)
```

Fonte: O autor (2025)

### 1.3 Atividade 6.3 - Daemon Cron (Backup)

Configuração do Crontab para backup agendado da pasta /var/log.

## Linha do Crontab:

```
1 15 17 * * * tar -zcvf /home/lucasmattoscanela/backup_log.tar.gz /var/log  
    /*.log
```

Figura 3 – Edição e verificação do Crontab

```

lucasmattoscanela@pop-os:~$ crontab -e
crontab: installing new crontab

22 de nov 5:20 PM lucasmattoscanela@pop-os ~

systemdudevd(525)
└─systemd(628) └─{thermald}(277)
    ├─{touchegg}(827) └─{touchegg}(1181)
    └─{touchegg}(1182) └─{touchegg}(2120)
udisksd(828) └─{udisksd}(900)
    ├─{udisksd}(911) └─{udisksd}(1105)
    └─{udisksd}(1105)
upowerd(1076) └─{upow} < > └─{upow}
wpa_supplicant(829)

UID      PID  PPID  CPUTIME  CPU%  COMMAND
lucasma... 4019  3993  0 16:38   0,00
lucasma... 4714  4019  14 17:08   0,00 ★ FAVORITO
lucasma... 4786  4714  0 17:08   0,00
lucasma... 4789  4786  3 17:08   0,00
lucasma... 4809  4786  0 17:08   0,00
lucasma... 4839  4786  0 17:08   0,00
lucasma... 4895  4786  15 17:08   0,00
lucasma... 4936  4786  3 17:08   0,00
lucasma... 4938  4786  0 17:08   0,00
lucasma... 4977  4786  0 17:08   0,00
lucasma... 4980  4786  0 17:08   0,00
lucasma... 5009  4786  0 17:08   0,00
lucasma... 5035  4019  0 17:08   0,00
lucasmattoscanela@pop-os:~$ jobs
[1]+  Executando          fire          [0] 05 14
lucasmattoscanela@pop-os:~$ 05 14
05: comando não encontrado
lucasmattoscanela@pop-os:~$ crontab
no crontab for lucasmattoscanela -
Select an editor. To change later
1. /bin/nano
2. /usr/bin/vim.tiny
Choose 1-2 [1]: 1
crontab: installing new crontab
lucasmattoscanela@pop-os:~$ crontab -e
crontab: installing new crontab
lucasmattoscanela@pop-os:~$ crontab -e
crontab: installing new crontab
lucasmattoscanela@pop-os:~$ crontab
No modification made
lucasmattoscanela@pop-os:~$ xdg-open /home/lucasmattoscanela
lucasmattoscanela@pop-os:~$ 
```

Fonte: O autor (2025)

## 1.4 Atividade 6.4 - Script de Informações do Usuário

Script que exibe usuário, diretório atual e shell padrão.

**Script (info.sh):**

```

1 #!/bin/bash
2 echo "Usuário atual: $USER"
3 echo "Diretório atual: $(pwd)"
4 echo "Shell atual: $(grep "^\$USER:" /etc/passwd | cut -d: -f7)" 
```

Figura 4 – Execução do script info.sh

```

lucasmattoscanela@pop-os:~/atividades_linux$ chmod +x info.sh
lucasmattoscanela@pop-os:~/atividades_linux$ ./info.sh
Usuário atual: lucasmattoscanela
Diretório atual: /home/lucasmattoscanela/atividades_linux
Shell atual: /bin/bash 
```

Fonte: O autor (2025)

## 1.5 Atividade 6.6 - Leitura de Variáveis e Operações

Script que lê três números e exibe soma e subtração.

**Script (calc.sh):**

```

1 #!/bin/bash
2 echo "Digite o primeiro número:"
3 read n1
4 echo "Digite o segundo número:"
5 read n2
6 echo "Digite o terceiro número:"
7 read n3
8
9 soma=$((n1 + n2 + n3))
10 sub=$((n1 - n2 - n3))
11
12 echo "A soma destes números é $soma"
13 echo "A subtração destes números é $sub"
```

Figura 5 – Execução do script calc.sh

```

lucasmattoscanela@pop-os:~/atividades_linux$ nano 66.sh
lucasmattoscanela@pop-os:~/atividades_linux$ chmod +x 66.sh
lucasmattoscanela@pop-os:~/atividades_linux$ ./66.sh
Digite o primeiro número:
20
Digite o segundo número:
40
Digite o terceiro número:
60
A soma destes números é 120
A subtração destes números é -80
```

Fonte: O autor (2025)

## 1.6 Atividade 6.7 - Parâmetros e Shift

Script que recebe 12 parâmetros e utiliza `shift` para deslocamento.

**Script (params.sh):**

```

1 #!/bin/bash
2 echo "Os primeiros 9 parâmetros são: $1 $2 $3 $4 $5 $6 $7 $8 $9"
3 shift 3
4 echo "Após shift 3, os próximos são: $7 $8 $9"
```

Figura 6 – Execução do script params.sh

```
A Substituição destes números é: 00
lucasmattoscanela@pop-os:~/atividades_linux$ nano params.sh
lucasmattoscanela@pop-os:~/atividades_linux$ chmod +x params.sh
lucasmattoscanela@pop-os:~/atividades_linux$ ./params.sh 1 2 3 4 5 6 7 8 9 10 11 12
Os primeiros 9 parâmetros são: 1 2 3 4 5 6 7 8 9
Após shift 3 (deslocar), os próximos são: 10 11 12
lucasmattoscanela@pop-os:~/atividades_linux$ 
```

Fonte: O autor (2025)

## 1.7 Atividades 6.8 e 6.9 - Testes de Parâmetros e Diretórios

Script que verifica número de parâmetros e existência de diretório.

**Script (testadir.sh):**

```
1 #!/bin/bash
2 if [ $# -ne 1 ]; then
3     echo "Erro: Digite exatamente um parâmetro."
4     exit 1
5 fi
6
7 if [[ -d "$1" ]]; then
8     echo "O parâmetro $1 é um diretório válido."
9 else
10    # Acentos como "Ã" agora funcionam graças ao bloco literate
11    echo "O parâmetro $1 NÃO é um diretório."
12 fi
```

Figura 7 – Execução do script testadir.sh

```
Após SHIFT 3 (deslocar), os próximos são: 10 11 12
lucasmattoscanela@pop-os:~/atividades_linux$ nano testadir.sh
lucasmattoscanela@pop-os:~/atividades_linux$ chmod +x testadir.sh
lucasmattoscanela@pop-os:~/atividades_linux$ ./testadir.sh
Erro: Digite exatamente um parâmetro.
lucasmattoscanela@pop-os:~/atividades_linux$ ./testadir.sh 1
O parâmetro 1 NÃO é um diretório.
lucasmattoscanela@pop-os:~/atividades_linux$ ./testadir.sh /home
O parâmetro /home é um diretório válido.
lucasmattoscanela@pop-os:~/atividades_linux$ ./testadir.sh
bash: ./testadir.sh: Arquivo ou diretório inexistente
lucasmattoscanela@pop-os:~/atividades_linux$ ./testadir.sh
Erro: Digite exatamente um parâmetro.
lucasmattoscanela@pop-os:~/atividades_linux$ 
```

Fonte: O autor (2025)

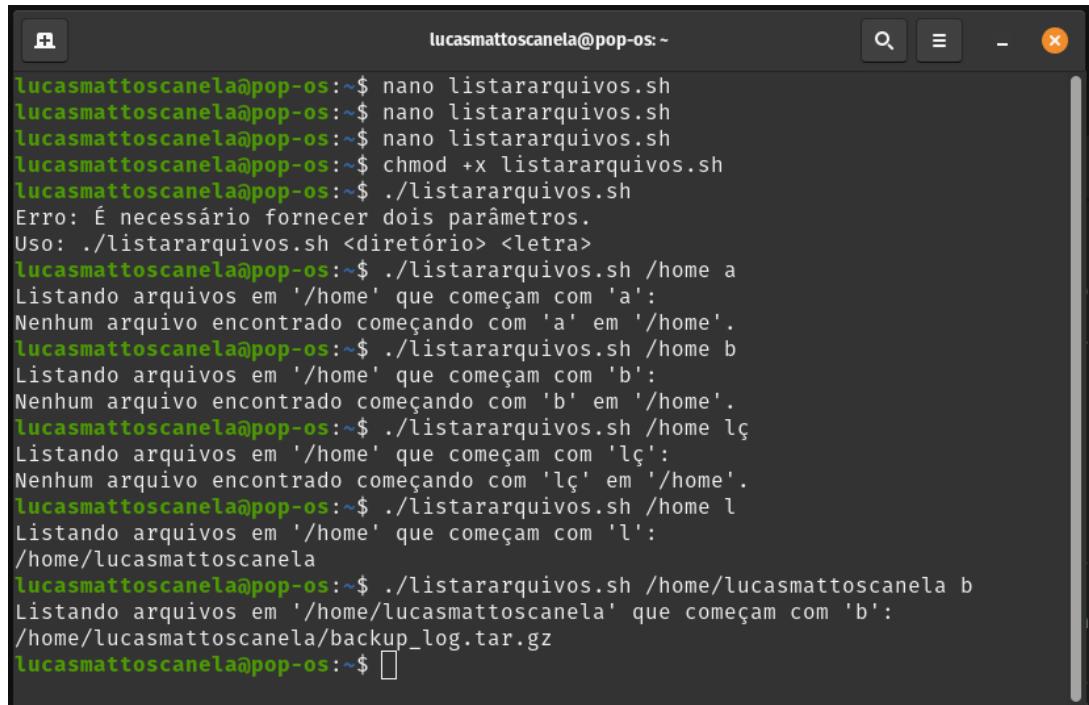
## 1.8 Atividade 6.10 - Listando arquivos passados por parâmetro

Script que recebe um diretório e uma letra como parâmetros, verificando se o primeiro parâmetro é um diretório válido e listando os arquivos que iniciam com a letra especificada.

**Script (atividade6\_10.sh):**

```
1 #!/bin/bash
2
3 # Verifica se foram passados exatamente 2 parâmetros
4 if [ $# -ne 2 ]; then
5     echo "Erro: É necessário fornecer dois parâmetros."
6     echo "Uso: $0 <diretório> <letra>"
7     exit 1
8 fi
9
10 diretorio=$1
11 letra=$2
12
13 # Verifica se o primeiro parâmetro é um diretório válido (-d)
14 if [[ -d "$diretorio" ]]; then
15     echo "Listando arquivos em '$diretorio' que começam com '$letra':"
16
17     # Lista os arquivos que começam com a letra fornecida
18     ls -d "$diretorio/$letra"* 2>/dev/null
19
20     # Verifica se o comando ls encontrou algo
21     if [ $? -ne 0 ]; then
22         echo "Nenhum arquivo encontrado com esse padrão."
23     fi
24 else
25     echo "Erro: O parâmetro '$diretorio' NÃO é um diretório válido."
26 fi
```

Figura 8 – Execução do script da Atividade 6.10



The screenshot shows a terminal window titled "lucasmattoscanela@pop-os: ~". The user has run the script "listararquivos.sh" multiple times with different parameters:

- First run: nano listararquivos.sh (script creation)
- Second run: nano listararquivos.sh (script modification)
- Third run: chmod +x listararquivos.sh (script execution)
- Fourth run: ./listararquivos.sh (script execution with error message: "Erro: É necessário fornecer dois parâmetros.")
- Fifth run: ./listararquivos.sh /home a (listing files starting with 'a' in the home directory)
- Sixth run: ./listararquivos.sh /home b (listing files starting with 'b' in the home directory)
- Seventh run: ./listararquivos.sh /home lç (listing files starting with 'lç' in the home directory)
- Eighth run: ./listararquivos.sh /home l (listing files starting with 'l' in the home directory)
- Ninth run: ./listararquivos.sh /home/lucasmattoscanela b (listing files starting with 'b' in the user's home directory)

The terminal also shows the path "/home/lucasmattoscanela" and the file "backup\_log.tar.gz".

Fonte: O autor (2025)

## 2 Roteiro de Atividades 7

### 2.1 Atividade 7.6 - Uso de Grep na Agenda

Script que busca um nome em um arquivo de agenda.

**Script (busca.sh):**

```
1 #!/bin/bash
2 if [ -z "$1" ]; then
3     echo "Erro: Você deve passar o nome de um usuário"
4     exit 1
5 else
6     echo "Recuperando informações..."
7     grep "$1" agenda
8 fi
```

Figura 9 – Busca na agenda

```
lucasmattoscanela@pop-os:~/atividades_linux
lucasmattoscanela@pop-os:~$ ./busca.sh
bash: ./busca.sh: Arquivo ou diretório inexistente
lucasmattoscanela@pop-os:~$ ls
'Área de Trabalho' Documentos Imagens Postman
atividades_linux Downloads Modelos Público
backup_log.tar.gz IdeaProjects Música Vídeos
lucasmattoscanela@pop-os:~$ cd atividades_linux
lucasmattoscanela@pop-os:~/atividades_linux$ ./busca.sh Lucas
Erro: Você deve passar o nome de um usuário
lucasmattoscanela@pop-os:~/atividades_linux$ ./busca.sh Lucas
Recuperando informações...
Lucas Mattos 1234-5678
lucasmattoscanela@pop-os:~/atividades_linux$ ./busca.sh Jorge
Recuperando informações...
lucasmattoscanela@pop-os:~/atividades_linux$ ./busca.sh Eduardo
Recuperando informações...
Eduardo Silva 9876-5432
lucasmattoscanela@pop-os:~/atividades_linux$
```

Fonte: O autor (2025)

### 2.2 Atividade 7.9 - Expressões Regulares

Script que verifica permissão de leitura e conta linhas/palavras do arquivo.

**Script (info\_arquivo.sh):**

```

1 #!/bin/bash
2 arquivo=$1
3 if [ -r "$arquivo" ]; then
4     echo "Permissão de leitura OK."
5     wc "$arquivo"
6 else
7     echo "Sem permissão de leitura ou arquivo inexistente."
8 fi

```

Figura 10 – Verificação de arquivo

```

lucasmattoscanela@pop-os:~/atividades_linux$ ./info_arquivo.sh
Sem permissão de leitura ou arquivo inexistente.
lucasmattoscanela@pop-os:~/atividades_linux$ ./info_arquivo.sh /home
Permissão de leitura OK.
wc: /home: É um diretório
    0      0      0 /home
lucasmattoscanela@pop-os:~/atividades_linux$ □

```

Fonte: O autor (2025)

**2.3 Atividade 7.11 - Menu de Opções**

Menu interativo para exibir status, usuários ou data.

**Script (menu.sh):**

```

1 #!/bin/bash
2 echo "1. Exibir status partições"
3 echo "2. Exibir usuários logados"
4 echo "3. Exibir data/hora"
5 echo "4. Sair"
6 read -p "Escolha uma opção: " opcao
7
8 case $opcao in
9     1) df -h ;;
10    2) who ;;
11    3) date ;;
12    4) exit 0 ;;
13    *) echo "Opção inválida" ;;
14 esac

```

Figura 11 – Execução do menu

```
lucasmattoscanela@pop-os:~/atividades_linux$ nano menu.sh
lucasmattoscanela@pop-os:~/atividades_linux$ chmod +x menu.sh
lucasmattoscanela@pop-os:~/atividades_linux$ ./menu.sh
1. Exibir status partições
2. Exibir usuários logados
3. Exibir data/hora
4. Sair
Escolha uma opção: 2
lucasmattoscanela :1           2025-11-22 16:35 (:1)
lucasmattoscanela@pop-os:~/atividades_linux$ ./menu.sh
1. Exibir status partições
2. Exibir usuários logados
3. Exibir data/hora
4. Sair
Escolha uma opção: 3
sáb 22 nov 2025 17:50:09 -03
lucasmattoscanela@pop-os:~/atividades_linux$ ./menu.sh
1. Exibir status partições
2. Exibir usuários logados
3. Exibir data/hora
4. Sair
Escolha uma opção: 1
Sist. Arq.          Tam. Usado Disp. Uso% Montado em
tmpfs              1,6G  2,0M  1,6G   1% /run
efivarfs           192K   63K  125K  34% /sys/firmware/efi/efivars
/dev/mapper/data-root 907G  28G  834G   4% /
tmpfs              7,8G  28K  7,8G   1% /dev/shm
tmpfs              5,0M    0  5,0M   0% /run/lock
/dev/sdb1          1020M 214M  807M  21% /boot/efi
/dev/sdb2           4,0G  3,1G 1016M  76% /recovery
tmpfs              1,6G  1,8M  1,6G   1% /run/user/1000
lucasmattoscanela@pop-os:~/atividades_linux$ █
```

Fonte: O autor (2025)