Parte 1: Ambiente Linux (GABARITO)

1. Atividades em Ambiente LINUX

1.1 – O comando mostra o nome do usuário, qual terminal o usuário está utilizando, a data e hora de login e o hostname ou IP do usuário. Exemplo:

```
user1@ABS:~$ who
user1    tty7         2021-03-15 15:13 (:0)
user1@ABS:~$
```

1.2.a –

```
user1@ABS:~$ ls -l
total 32
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13 'Área de Trabalho'
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13  Documentos
drwxr-xr-x 3 user1 user1 4096 mar 15 15:29  Downloads
drwxr-xr-x 2 user1 user1 4096 mar 15 15:31  Imagens
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13  Modelos
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13  Música
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13  Público
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13  Vídeos
user1@ABS:~$
```

1.2.b – São exibidas informações de tipo de arquivo e permissões, número de links para o arquivo, usuário dono do arquivo, grupo do arquivo, tamanho, data de modificação e nome do arquivo/diretório.

1.3.a – O parâmetro **-h** representa o modo para leitura humana, onde o tamanho das partições é exibido em Mbytes, Gbytes e afins ao invés de Kbytes (por exemplo: ele mostra 966M ao invés de 989004K).

```
user1@ABS:~$ df -h
Sist. Arq.      Tam. Usado Disp. Uso% Montado em
udev            966M     0  966M   0% /dev
tmpfs           199M  1,3M  198M   1% /run
/dev/sda1        14G  6,2G  6,8G  48% /
tmpfs           994M     0  994M   0% /dev/shm
tmpfs           5,0M  4,0K  5,0M   1% /run/lock
tmpfs           994M     0  994M   0% /sys/fs/cgroup
/dev/sda2       5,9G  483M  5,1G   9% /home
tmpfs           199M  8,0K  199M   1% /run/user/1000
tmpfs           199M  8,0K  199M   1% /run/user/1001
tmpfs           199M   24K  199M   1% /run/user/1002
user1@ABS:~$
```

1.4.a –

```
user1@ABS:~/Downloads$ cat teste.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam vulputate vitae o
dio a pretium. Nullam consequat nunc ipsum, non consequat eros rutrum non. Vivamu
s arcu nisl, tincidunt facilisis dolor ac, pharetra gravida arcu. Sed aliquet var
ius nunc convallis interdum. Donec vestibulum, ligula sit amet laoreet tempus, se
m metus placerat mi, eget faucibus leo eros at neque. Nullam porta, lectus id sod
ales hendrerit, metus eros venenatis sapien, eget faucibus enim velit ac eros. Ve
stibulum vitae gravida diam. Vivamus et mauris at mi placerat euismod quis id vel
```

1.4.b – As informações podem ser vistas nos campos **model_name**, **cpu_MHz** e **cache size**.

```
processor       : 3
vendor_id       : AuthenticAMD
cpu family      : 21
model           : 2
model name      : AMD FX-8320E Eight-Core Processor
stepping        : 0
microcode       : 0xffffffff
cpu MHz         : 3214.228
cache size      : 2048 KB
```

1.4.c –

```
nodev   ramfs
nodev   hugetlbfs
nodev   devpts
```

1.5 – Para mudar, deve subir 1 nível na hierarquia dos diretórios, o comando **cd ..** pode ser utilizado.

```
user1@ABS:~/Downloads$ pwd
/home/user1/Downloads
user1@ABS:~/Downloads$ cd ..
user1@ABS:~$ pwd
/home/user1
user1@ABS:~$
```

1.6.a – Os comandos **cp** e **mv** podem ser utilizados para copiar um arquivo de um diretório filho para outro diretório filho.

```
user1@ABS:~$ cp Downloads/teste.txt Documentos/teste_copia.txt
user1@ABS:~$ ls -l Documentos/
total 4
-rw-rw-r-- 1 user1 user1 2757 mar 15 16:12 teste_copia.txt
user1@ABS:~$
```

```
user1@ABS:~$ mv Downloads/teste.txt Modelos
user1@ABS:~$ ls -l Modelos/
total 4
-rw-rw-r-- 1 user1 user1 2757 mar 15 15:51 teste.txt
user1@ABS:~$
```

1.6.b – Neste exemplo, todos os usuários possuíam permissão de leitura e escrita sobre o arquivo. O comando **chmod** foi utilizado para alterar essas permissões. O parâmetro **u=rw** concede permissão de leitura e escrita para o usuário dono do arquivo, o parâmetro **g=r** muda as permissões do grupo para leitura apenas e o parâmetro **o-rw** remove as permissões de leitura e escrita para os demais usuários. É possível adicionar permissões novamente com o sinalizador **+**.

```
user1@ABS:~/Documentos$ ls -l
total 4
-rw-rw-rw- 1 user1 user1 2757 mar 15 16:12 arquivo.txt
user1@ABS:~/Documentos$ chmod u=rw,g=r,o-rw arquivo.txt
user1@ABS:~/Documentos$ ls -l
total 4
-rw-r----- 1 user1 user1 2757 mar 15 16:12 arquivo.txt
user1@ABS:~/Documentos$
```

1.7 –

```
user1@ABS:~$ ls
'Área de Trabalho'   Documentos   Downloads   Imagens   Modelos   Música   Público   Vídeos
user1@ABS:~$ mkdir /home/user1/SO
user1@ABS:~$ ls
'Área de Trabalho'   Documentos   Downloads   Imagens   Modelos   Música   Público   SO   Vídeos
user1@ABS:~$ rmdir /home/user1/SO
user1@ABS:~$ ls
'Área de Trabalho'   Documentos   Downloads   Imagens   Modelos   Música   Público   Vídeos
user1@ABS:~$
```

1.8 –

```
user1@ABS:~$ ls -l > list.txt
user1@ABS:~$ cat list.txt
total 32
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13 Área de Trabalho
drwxr-xr-x 2 user1 user1 4096 mar 15 16:40 Documentos
drwxr-xr-x 3 user1 user1 4096 mar 15 16:41 Downloads
drwxr-xr-x 2 user1 user1 4096 mar 15 16:43 Imagens
-rw-rw-r-- 1 user1 user1    0 mar 15 16:44 list.txt
drwxr-xr-x 2 user1 user1 4096 mar 15 16:27 Modelos
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13 Música
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13 Público
drwxr-xr-x 2 user1 user1 4096 mar 15 15:13 Vídeos
user1@ABS:~$
```

1.9 – O comando **find** pode ser utilizado para buscar arquivos com nomes idênticos à pesquisa, ou para buscar arquivos que contenham a pesquisa no nome. No exemplo a seguir, foram consultados todos os arquivos com nomes começando com palavar **teste,** de maneira recursiva em todos os subdiretórios. Esse comando pode ser utilizado com outros parâmetros, como buscar todos os arquivos que contenham determinado trecho de texto, palavra e afins.

```
user1@ABS:~$ find . -name "teste*"
./Documentos/teste_copia (cópia 1).txt
./Documentos/teste.txt
./.local/share/Trash/files/teste_copia.txt
./.local/share/Trash/info/teste_copia.txt.trashinfo
./teste.txt
./Downloads/teste.txt
user1@ABS:~$
```

1.10 – Utilizando o parâmetro **-a**, as informações completas com a versão do kernel, nome da distribuição, data de instalação e arquitetura de hardware são exibidas.

```
user1@ABS:~$ uname -a
Linux ABS 5.4.0-66-generic #74-Ubuntu SMP Wed Jan 27 22:54:38 UTC 2021 x86_64
 x86_64 x86_64 GNU/Linux
```

2. Praticando Comandos

2.1 –

```
top - 17:00:57 up  1:55,  1 user,  load average: 0,22, 0,40, 0,48
Tarefas: 238 total,   1 em exec., 237 dormindo,   0 parado,   0 zumbi
%CPU(s):  3,8 us,  4,7 sis,  0,0 ni, 91,2 oc,  0,0 ag,  0,0 ih,  0,3 is  0,0 tr
MB mem :   1987,2 total,    186,4 livre,   1288,2 usados,    512,5 buff/cache
MB swap:    662,2 total,    482,5 livre,    179,8 usados,    453,0 mem dispon.

    PID USUARIO   PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TEMPO+ COMANDO
  10619 user1     20   0 2799484 287184 162512 S  20,3  14,1   3:10.28 Web Content
   8430 root      20   0 1541468 283856  69804 S  17,3  13,9   3:13.41 Xorg
   9926 user1     20   0 3052220 270292 133504 S  13,0  13,3   3:32.89 firefox
   8713 user1     20   0 1269112  75844  68704 S   9,6   3,7   0:42.07 xfwm4
  10951 user1     20   0  234956  30280  24488 S   9,3   1,5   0:00.93 xfce4-screensho
```

2.2 – Os processos são exibidos de maneira hierárquica utilizando estrutura de árvores.

```
 8430 tty7     00:03:36  \_ Xorg
 8549 ?        00:00:00  \_ lightdm
 8566 ?        00:00:01      \_ xfce4-session
 8679 ?        00:00:00          \_ ssh-agent
 8713 ?        00:00:47          \_ xfwm4
 8730 ?        00:00:05          \_ xfce4-panel
 8741 ?        00:00:01          |   \_ panel-1-whisker
 8742 ?        00:00:00          |   \_ panel-5-systray
 8743 ?        00:00:00          |   \_ panel-6-notific
 8744 ?        00:00:00          |   \_ panel-7-indicat
 8746 ?        00:00:00          |   \_ panel-8-statusn
 8747 ?        00:00:00          |   \_ panel-9-power-m
 8749 ?        00:00:06          |   \_ panel-10-pulsea
 8736 ?        00:00:10          \_ Thunar
 8745 ?        00:00:02          \_ xfdesktop
 8787 ?        00:00:00          \_ xiccd
```

2.3 – O programa entra em segundo plano, mas continuam executando. Os mesmos podem ser vistos com o uso do comando **ps**.

```
user1@ABS:~$ sleep 300
^Z
[3]+  Parado                 sleep 300
user1@ABS:~$ ps
    PID TTY          TIME CMD
  11638 pts/0    00:00:00 bash
  11644 pts/0    00:00:00 sleep
  11645 pts/0    00:00:00 sleep
  11646 pts/0    00:00:00 sleep
  11647 pts/0    00:00:00 ps
```

2.4 –

```
user1@ABS:~$ jobs -l
[1]  12794 Executando              sleep 20 &
[2]- 12795 Executando              sleep 100 &
[3]+ 12796 Executando              sleep 150 &
user1@ABS:~$ fg 1
sleep 20
user1@ABS:~$ jobs -l
[2]- 12795 Executando              sleep 100 &
[3]+ 12796 Executando              sleep 150 &
user1@ABS:~$
```

3. Makefile

3.1 –

```
user1@ABS:~/hellomake$ make -f makefile1
gcc -o hellomake hellomake.c hellofunc.c -I.
user1@ABS:~/hellomake$ ./hellomake
Olá arquivo makefile!
user1@ABS:~/hellomake$ make -f makefile2
gcc -I.   -c -o hellomake.o hellomake.c
gcc -I.   -c -o hellofunc.o hellofunc.c
gcc -o hellomake hellomake.o hellofunc.o -I.
user1@ABS:~/hellomake$ ./hellomake
Olá arquivo makefile!
user1@ABS:~/hellomake$ make -f makefile3
make: 'hellomake' está atualizado.
user1@ABS:~/hellomake$ ./hellomake
Olá arquivo makefile!
user1@ABS:~/hellomake$
```

3.2 – Atualizações para o **makefile** funciona com os códigos. Basta adicionar os nomes de todos os arquivos (sem as extensões) na variável **PROGRAMS**. Após isso, criar a seção com o comando do **gcc**, utilizando a variável **all**.

```
CC = gcc
CFLAGS = -g

PROGRAMS = pilha loop-malloc


all: $(PROGRAMS)

make: $(all)
        $(CC) -o $(all) $(CFLAGS)

clean:
        rm -f *~ $(PROGRAMS)
```

```
user1@ABS:~/Downloads/exercicio1$ ls
loop-malloc.c  makefile1  pilha.c
user1@ABS:~/Downloads/exercicio1$ make -f makefile1
gcc -g     pilha.c   -o pilha
gcc -g     loop-malloc.c   -o loop-malloc
user1@ABS:~/Downloads/exercicio1$ ls
loop-malloc  loop-malloc.c  makefile1  pilha  pilha.c
user1@ABS:~/Downloads/exercicio1$
```