

Gerenciamento de processos



Prof. Dr. João Paulo Lemos Escola
Copyright © 2022

Conteúdo

- lsusb
- lspci
- lshw
- jobs
- kill
- ps aux
- killall
- seq
- bg
- fg

lsusb e lspci

- Lista os dispositivos USB e os dispositivos PCI do seu computador:

```
ubuntu@ubuntu:~$ lsusb
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
ubuntu@ubuntu:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: VMware SVGA II Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
```

lshw

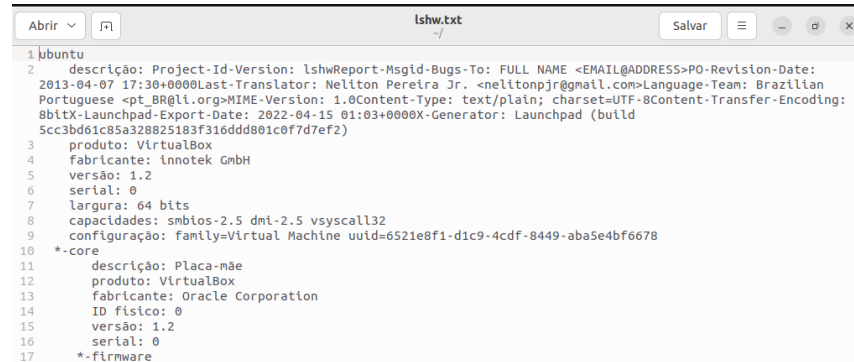
- Lista o hardware do seu computador;
- Utilize o prefixo *sudo*:

```
capacidades: 18042
*-input:4
    produto: ImExPS/2 Generic Explorer Mouse
    ID físico: 5
    nome lógico: input5
    nome lógico: /dev/input/event4
    nome lógico: /dev/input/mouse0
    capacidades: 18042
ubuntu@ubuntu:~$
```

Gerando um arquivo de detalhes

- Vamos criar um arquivo com todo o conteúdo retornado pelo comando *lshw*:
- Agora temos um relatório de hardware de nosso computador, para guardar e enviar aos amigos;
- Abra com o gedit:

```
ubuntu@ubuntu:~$ sudo lshw > lshw.txt
ubuntu@ubuntu:~$ gedit lshw.txt
```



```
1 ubuntu
2  descrição: Project-Id-Version: lshwReport-MsgId-Bugs-To: FULL NAME <EMAIL@ADDRESS>-PO-Revision-Date:
3 2013-04-07 17:30+0000Last-Translator: Neliton Pereira Jr. <nelitonpjr@gmail.com>Language-Team: Brazilian
4 Portuguese <pt_BR@li.org>MIME-Version: 1.0Content-Type: text/plain; charset=UTF-8Content-Transfer-Encoding:
5 8bitX-Launchpad-Export-Date: 2022-04-15 01:03+0000X-Generator: Launchpad (build
6 5cc3bd61c85a328825183f316ddd801c0f7d7ef2)
7 produto: VirtualBox
8 fabricante: Innotek GmbH
9 versão: 1.2
10 serial: 0
11 largura: 64 bits
12 capacidades: smbios-2.5 dmi-2.5 vsyscall32
13 configuração: family=Virtual Machine uuid=6521e8f1-d1c9-4cdf-8449-aba5e4bf6678
14
15 *-core
16  descrição: Placa-mãe
17  produto: VirtualBox
18  fabricante: Oracle Corporation
19  ID físico: 0
20  versão: 1.2
21  serial: 0
22 *-firmware
```

Primeiro e segundo plano

- Veja que o gedit está em primeiro plano, então não podemos digitar comandos se não fecharmos o gedit:

```
14      ID físico: 0
15      versão: 1.2
16      serial: 0
17      *-firmware
18         descrição: BIOS
19         fabricante: innotek GmbH
20         ID físico: 0
21         versão: VirtualBox
22         data: 12/01/2006
```

Texto sem formatação ▾ Largura da tabulação: 8 ▾ Lin

```
ubuntu@ubuntu:~$ gedit lshw.txt

(gedit:8304): dbind-WARNING **: 12:17:16.763: Couldn't connect to accessibility
bus: Failed to connect to socket /root/.cache/at-spi/bus_0: Permissão negada
ubuntu@ubuntu:~$ gedit lshw.txt

(gedit:8332): dbind-WARNING **: 12:18:16.192: Couldn't connect to accessibility
bus: Failed to connect to socket /root/.cache/at-spi/bus_0: Permissão negada
```

CTRL+C

- Nesse caso, utilizando o CTRL+C, estamos finalizando o processo;
- Agora podemos voltar a digitar comandos no terminal;
- Mas precisa ser sempre assim? Não.

Caractere &

- Permite executar um comando em segundo plano;
- Agora temos o gedit em execução e ainda podemos digitar comandos;
- Veja que o [1] representa que há um processo em segundo plano (background) e seu número (PID) é 8359:

```
17      *-firmware
18          descrição: BIOS
19          fabricante: innotek GmbH
20          ID físico: 0
21          versão: VirtualBox
22          data: 12/01/2006
```

Texto sen

```
ubuntu@ubuntu:~$ gedit lshw.txt &
[1] 8359
ubuntu@ubuntu:~$
(gedit:8359): dbind-WARNING **: 12:19:33
bus: Failed to connect to socket /root/.
ubuntu@ubuntu:~$
```


jobs

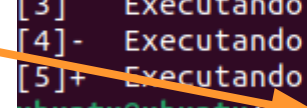
- Retorna a lista de processos em background;
- Quando um processo é concluído, a informação aparece no terminal, junto com o próximo comando ou Enter digitado:

```
ubuntu@ubuntu:~$ sleep 10 &
[1] 9794
ubuntu@ubuntu:~$ sleep 10 &
[2] 9795
ubuntu@ubuntu:~$ sleep 10 &
[3] 9796
ubuntu@ubuntu:~$ sleep 10 &
[4] 9797
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ jobs
[1]      Executando                sleep 10 &
[2]      Executando                sleep 10 &
[3]-    Executando                sleep 10 &
[4]+    Executando                sleep 10 &
ubuntu@ubuntu:~$
[1]      Concluído                 sleep 10
[2]      Concluído                 sleep 10
[3]-    Concluído                 sleep 10
[4]+    Concluído                 sleep 10
```

kill

- Mata o processo em background;
- Para isso, utilizamos o caractere % junto com o número do processo:

```
ubuntu@ubuntu:~$ sleep 1000 &
[1] 9800
ubuntu@ubuntu:~$ sleep 1000 &
[2] 9801
ubuntu@ubuntu:~$ sleep 1000 &
[3] 9802
ubuntu@ubuntu:~$ sleep 1000 &
[4] 9803
ubuntu@ubuntu:~$ sleep 1000 &
[5] 9804
ubuntu@ubuntu:~$ jobs
[1]      Executando                sleep 1000 &
[2]      Executando                sleep 1000 &
[3]      Executando                sleep 1000 &
[4]-    Executando                sleep 1000 &
[5]+    Executando                sleep 1000 &
ubuntu@ubuntu:~$ kill %4
ubuntu@ubuntu:~$ jobs
[1]      Executando                sleep 1000 &
[2]      Executando                sleep 1000 &
[3]      Executando                sleep 1000 &
[4]-    Terminado                sleep 1000
[5]+    Executando                sleep 1000 &
```



ps aux

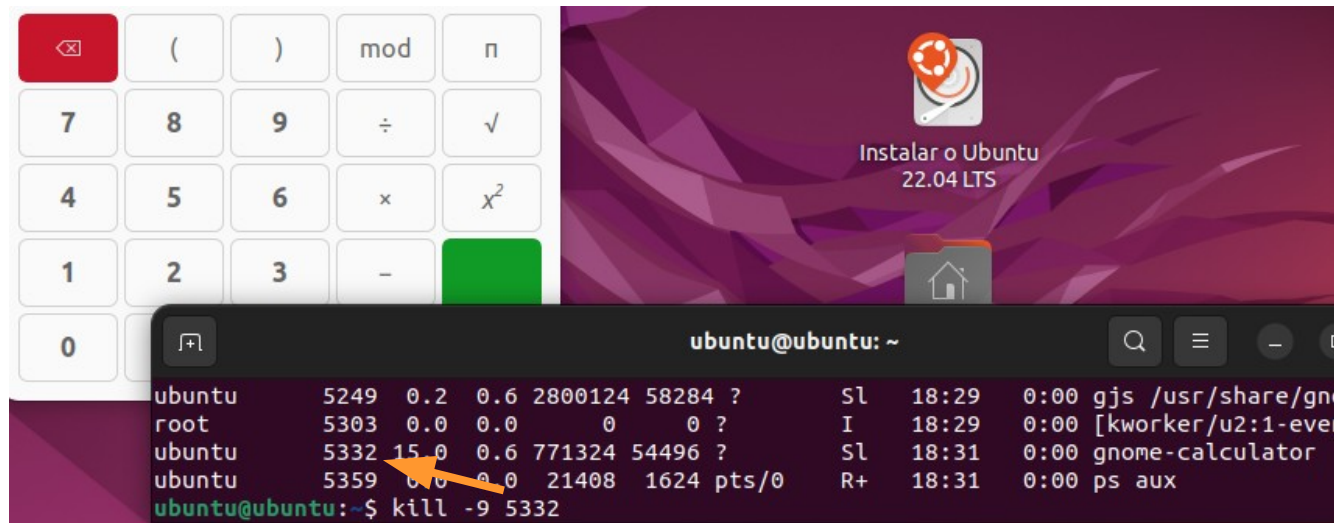
- Lista os processos em execução;
- Podemos visualizar os PIDs dos processos;

```
ubuntu@ubuntu:~$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.5	0.1	166592	12060	?	Ss	18:12	0:00	/sbin/init ma
root	2	0.0	0.0	0	0	?	S	18:12	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	18:12	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	18:12	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I	18:12	0:00	[kworker/0:0-
root	6	0.0	0.0	0	0	?	I<	18:12	0:00	[kworker/0:0H
root	7	0.0	0.0	0	0	?	I	18:12	0:00	[kworker/0:1-
root	8	0.1	0.0	0	0	?	I	18:12	0:00	[kworker/u2:0

kill -9 PID

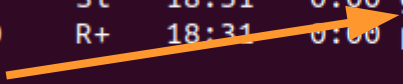
- Mata o processo pelo PID;
- O nível mais alto de requisição é o 9;



killall -9 <nome>

- Mata o processo pelo nome;
- Também elimina a árvore de processo, caso exista:

```
root      5303  0.0  0.0      0      0 ?        I    18:29   0:00 [kworker/u2:1-event
ubuntu    5332 15.0  0.6 771324 54496 ?        Sl   18:31   0:00 gnome-calculator
ubuntu    5359  0.0  0.0   21408   1624 pts/0    R+   18:31   0:00 ps aux
ubuntu@ubuntu:~$ killall -9 gnome-calculator
ubuntu@ubuntu:~$
```



seq

- Gera uma sequência, ou seja, uma contagem até o número especificado;
- Também é possível definir o número inicial da contagem:

```
ubuntu@ubuntu:~$ seq 10
1
2
3
4
5
6
7
8
9
10
ubuntu@ubuntu:~$ seq 2 10
2
3
4
5
6
7
8
9
10
```

Fila de processos

- ; = executa o processo e, ao termina, executa o próximo;
- & = executa o processo em background e já dispara o próximo;
- && = executa o processo e, caso não resulte em falha, executa o próximo;

```
ubuntu@ubuntu:~$ sleep 10; echo "obrigado"
obrigado
ubuntu@ubuntu:~$ sleep 10 & echo "obrigado"
[6] 9830
obrigado
ubuntu@ubuntu:~$ sleep 10 && echo "obrigado"
[6]+  Concluído                sleep 10
obrigado
ubuntu@ubuntu:~$
```

Exemplo de fila de processos

- ; = Aguarda 10 segundos e mostra uma mensagem em seguida;
- & = Aguarda 10 segundos em background e mostra a mensagem;
- && = Aguarda 10 segundos, e caso o processo não retorne erro, executa o próximo comando.

```
ubuntu@ubuntu:~$ sleep 10; echo "obrigado"
obrigado
ubuntu@ubuntu:~$ sleep 10 & echo "obrigado"
[6] 9830
obrigado
ubuntu@ubuntu:~$ sleep 10 && echo "obrigado"
[6]+  Concluído                sleep 10
obrigado
ubuntu@ubuntu:~$
```


Exemplo de laço com seq

```
ubuntu@ubuntu:~$ for i in `seq 10`; do echo "contando...$i"; sleep 1; done
contando...1
contando...2
contando...3
contando...4
contando...5
contando...6
contando...7
contando...8
contando...9
contando...10
ubuntu@ubuntu:~$
```

bg e fg

- Quando temos um processo ocupando o terminal, podemos pará-lo com o CTRL+Z;
- O comando *bg* retoma o processo em background (segundo plano);
- O comando *fg* retoma o processo em foreground (primeiro plano).

```
ubuntu@ubuntu:~$ sleep 1000
^Z
[7]+  Parado                  sleep 1000
ubuntu@ubuntu:~$ bg
[7]+  sleep 1000 &
ubuntu@ubuntu:~$ sleep 1000
^Z
[8]+  Parado                  sleep 1000
ubuntu@ubuntu:~$ fg
sleep 1000
█
```

O que vimos?

- lsusb
- lspci
- lshw
- jobs
- kill
- ps aux
- killall
- seq
- bg
- fg

Na próxima aula...

- Gerenciamento de usuários e grupos.