

```
Ign http://us.archive.ubuntu.com trusty InRelease
Get:2 http://us.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:3 http://extras.ubuntu.com trusty Release.gpg [72 B]
Hit http://extras.ubuntu.com trusty Release
Hit http://us.archive.ubuntu.com trusty-backports InRelease
Hit http://us.archive.ubuntu.com trusty Release.gpg
Hit http://extras.ubuntu.com trusty/main Sources
Get:4 http://security.ubuntu.com trusty-security/main Sources [124 kB]
Get:5 http://us.archive.ubuntu.com trusty-updates/main Sources [390 kB]
Hit http://extras.ubuntu.com trusty/main i386 Packages
Get:6 http://security.ubuntu.com trusty-security/restricted Sources [4,637 B]
Get:7 http://security.ubuntu.com trusty-security/universe Sources [48.6 kB]
Get:8 http://security.ubuntu.com trusty-security/multiverse Sources [3,208 B]
Get:9 http://us.archive.ubuntu.com trusty-updates/restricted Sources [5,911 B]
Get:10 http://us.archive.ubuntu.com trusty-updates/universe Sources [173 kB]
Get:11 http://security.ubuntu.com trusty-security/main i386 Packages [539 kB]
Get:12 http://us.archive.ubuntu.com trusty-updates/multiverse Sources [7,535 B]
Get:13 http://us.archive.ubuntu.com trusty-updates/main i386 Packages [904 kB]
Ign http://extras.ubuntu.com trusty/main Translation-en_US
Ign http://extras.ubuntu.com trusty/main Translation-en
Get:14 http://security.ubuntu.com trusty-security/restricted i386 Packages [13.1 kB]
Get:15 http://security.ubuntu.com trusty-security/universe i386 Packages [151 kB]
Get:16 http://security.ubuntu.com trusty-security/multiverse i386 Packages [4,204 B]
Get:17 http://us.archive.ubuntu.com trusty-updates/restricted Translation-en [3,208 B]
Get:18 http://us.archive.ubuntu.com trusty-updates/universe Translation-en [173 kB]
Get:19 http://us.archive.ubuntu.com trusty-updates/multiverse Translation-en [3,208 B]
Get:20 http://security.ubuntu.com trusty-security/multiverse Translation-en [4,204 B]
Get:21 http://us.archive.ubuntu.com trusty-updates/main i386 Packages [904 kB]
Hit http://security.ubuntu.com trusty-security/restricted Translation-en [3,357 B]
Hit http://us.archive.ubuntu.com trusty-updates/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty-updates/restricted Translation-en
Get:23 http://security.ubuntu.com trusty-security/universe Translation-en [88.2 kB]
Hit http://us.archive.ubuntu.com trusty-updates/universe Translation-en
Hit http://us.archive.ubuntu.com trusty-updates/main i386 Packages [904 kB]
Hit http://us.archive.ubuntu.com trusty-updates/multiverse Translation-en
100% [18 Translation-en [ip2 6]] [Waiting for headers] 4.4 kB/s
```

# BASH TERMINAL LINUX PARA INICIANTE

```
2050 natanael 20 0 324924 105888 60676 S 2.0 10.3 0:09.82 compiz
1110 root 20 0 226592 43288 22864 S 1.3 4.2 0:04.23 Xorg
2383 natanael 20 0 128356 29644 22896 S 0.7 2.9 0:01.00 gnome-terminal
2425 natanael 20 0 5568 2812 2368 R 0.7 0.3 0:00.04 top
1826 natanael 20 0 46612 7624 6728 S 0.3 0.7 0:00.12 ibus-daemon
1 root 20 0 4464 3648 2576 S 0.0 0.4 0:03.05 init
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 20 0 0 0 0 S 0.0 0.0 0:00.12 ksoftirqd/0
5 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:0H
6 root 20 0 0 0 0 S 0.0 0.0 0:00.39 kworker/u16:0
7 root 20 0 0 0 0 S 0.0 0.0 0:00.38 rcu_sched
8 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_bh
9 root rt 0 0 0 0 S 0.0 0.0 0:00.00 migration/0
10 root rt 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/0
11 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 khelper
12 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
13 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 netns
14 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 perf
15 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd
16 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 writeback
17 root 25 5 0 0 0 S 0.0 0.0 0:00.00 ksmd
18 root 39 19 0 0 0 S 0.0 0.0 0:00.91 khugepaged
19 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 crypto
20 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kintegrityd
21 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 bioset
22 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kblockd
23 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 ata_sff
24 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 md
25 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 devfreq_wq
27 root 20 0 0 0 0 S 0.0 0.0 0:00.73 kworker/0:1
29 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kswapd0
30 root 20 0 0 0 0 S 0.0 0.0 0:00.00 fsnotify_mark
31 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ecryptfs-kthrea
42 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kthrotld
```

```
btcfllash (8) - firmware flash utility for BTC DRW1008 DVD+/-RW recorder.
dvd+rw-booktype (1) - format DVD+/-RW/-RAM disk with a logical format
dvd+rw-format (1) - format DVD+/-RW/-RAM disk
dvd+rw-mediainfo (1) - display information about dvd drive and disk
dvd-ram-control (1) - checks features of DVD-RAM discs
growisofs (1) - combined genisoimage frontend/DVD recording program.
rpl8 (8) - Firmware loader for DVD drives
natanael@ubuntu:~$ apropos media
dvd+rw-mediainfo (1) - display information about dvd drive and disk
eject (1) - eject removable media
gst-typefind-1.0 (1) - print Media type of file
mli-tool (8) - view, manipulate media-independent interface status
netscsid (1) - write data to optical disk media
URI::data (3pm) - URI that contains immediate data
wodim (1) - write data to optical disk media
```

```
natanael@ubuntu:~$ apropos music
rhythmbox (1) - music player and library for tagged files using GStreamer
natanael@ubuntu:~$ apropos text
awk (1) - pattern scanning and text processing language
bit-xtm (1) - xtm encoding of text files
bzip2 (1) - bzip2 file format and utilities
bzmore (1) - file perusal filter for crt viewing of bzip2 compressed...
chcon (1) - change file security context
csplit (1) - split a file into sections determined by context lines
dcgettext (3) - translate message
dcgettext (2) - translate message and choose plural form
```

# NATANAEL ANTONIOLI



## 1. Introdução

Olá! Seja bem-vindo a mais um curso produzido pelo canal Fábrica de Noobs. Neste, apresentarei os conhecimentos básicos necessários para utilizar o terminal de qualquer distribuição Linux, de forma que você, ao final do curso, seja capaz de realizar a maioria dos procedimentos que um usuário iniciante pode necessitar.

Tais procedimentos vão desde de operações simples de arquivos (tal como criar diretórios e navegador por eles) até instalação e execução de novos programas e controle de contas de usuário.

Dominar esses conhecimentos é fundamental para qualquer usuário que queira se iniciar em Linux, principalmente em distribuições com interfaces não muito intuitivas e amigáveis.

O objetivo desta apostila é transmitir a você os comandos básicos para se operar uma interface Linux, em especial, em distribuições como o Kali Linux, que serão de uso fundamental no estudo de segurança da informação.

No entanto, realizaremos todos os procedimentos no Ubuntu, a fim de deixar este curso acessível também à usuários com outros objetivos.

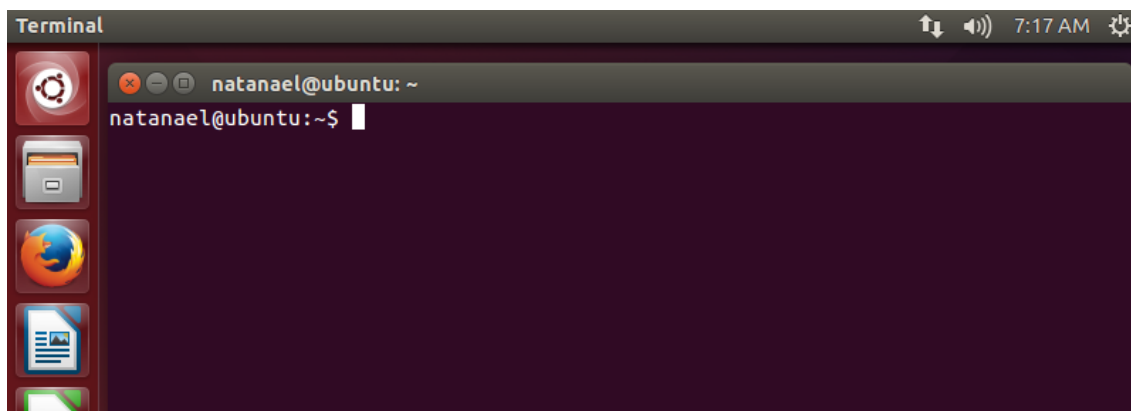
Você pode aprender os passos para a instalação do Ubuntu em <https://www.youtube.com/watch?v=kjFEHijKruw>, ou verificar o guia em PDF passo-a-passo para outras distribuições em <http://www.fabricadenoobs.com.br/apostilas/>. Você pode fazer a instalação como sistema operacional nativo ou em uma máquina virtual.

Sendo assim, prepare um ambiente com Linux da forma que preferir e prossiga para os próximos capítulos.

Bons estudos!

## 2. Familiarizando-se com o Terminal

Nosso objeto de estudo será o Terminal do Linux. Ele é acessível através da barra de busca do sistema operacional e tem a seguinte aparência.



Uma das operações mais elementares que podemos realizar em um Terminal é a navegação entre diretórios – as famosas “pastas” de um computador.

Podemos saber em que pasta estamos utilizando o comando *pwd*. Ele tem a função de informar em qual diretório nós estamos no momento.

```
natanael@ubuntu:~$ pwd
/home/natanael
```

Outra ferramenta interessante é o comando *who am i*, que permite sabermos qual conta de usuário estamos utilizando e a hora atual.

```
natanael@ubuntu:~$ who am i
natanael pts/1    2017-02-03 07:16 (:0)
```

O comando *date* também possui uma função parecida:

```
natanael@ubuntu:~$ date
Fri Feb 3 07:22:41 PST 2017
```

Caso queiramos “limpar” a tela, podemos fazê-la com o comando *clear*. Ele irá apagar quaisquer informações que estejam sendo exibidas em nosso terminal.

Em contrapartida, caso nosso desejo seja verificar todos os comandos que já foram inseridos na máquina, podemos usar o comando *history*, que tem essa função.

```
natanael@ubuntu:~$ history
1 pwd
```

```
2 who am i
3 date
4 add user pedro
5 adduser pedro
6 sudo adduser pedro
7 clear
8 history
9 pwd
10 cd /
11 pwd
12 ls
13 ls -l
```

Tal histórico pode ser deletado utilizando o argumento *-c* após o comando *history*. Argumentos são instruções adicionais que podemos utilizar após a execução de um comando.

```
natanael@ubuntu:~$ history -c
natanael@ubuntu:~$ history
1 history
```

Assim como no Windows, contas de usuário em Linux também possui determinados poderes. Em algumas distribuições, a conta padrão já possui todos os privilégios de administrador (aqui denominado root), enquanto que em outras – como o Ubuntu – não.

Nessas situações, a permissão de administrador pode ser obtida utilizando-se o comando *sudo* antes de qualquer outro comando que exija tal permissão.

Os privilégios de administrador podem ser utilizados, por exemplo, para adicionar um novo usuário na máquina. Isso pode ser feito através do comando *add user*, seguido do nome de usuário que desejamos. Por exemplo:

```
natanael@ubuntu:~$ sudo adduser natanaelnovo
[sudo] password for natanael:
Adding user `natanaelnovo' ...
```

```
Adding new group `natanaelnovo' (1003) ...
Adding new user `natanaelnovo' (1003) with group `natanaelnovo' ...
Creating home directory `/home/natanaelnovo' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for natanaelnovo
Enter the new value, or press ENTER for the default

  Full Name []: Natanael
  Room Number []: 16
  Work Phone []: 555555
  Home Phone []: 55555
  Other []: Teste

Is the information correct? [Y/n] Y
```

Na execução acima, o programa primeiramente solicitou a senha de usuário da nossa conta. Em seguida, solicitou a senha da nova conta a ser criada, junto com algumas informações padrão. Após a confirmação, a nova conta foi criada.

Podemos alternar entre contas de usuário utilizando o comando *su* seguido da conta que desejamos acessar.

```
natanael@ubuntu:~$ su natanaelnovo
Password:
natanaelnovo@ubuntu:/home/natanael$
```



### 3. Pesquisa por ajuda

É natural que, ao utilizar um sistema novo, precisemos de ajuda com algumas ferramentas. Justamente por essa razão, existem alguns comandos em Linux que lhe serão fiéis companheiros em situações do tipo.

O primeiro deles é o argumento `-h` ou `--help`, que pode ser inserido depois de quase todos os comandos existentes dentro do terminal. Ele irá apresentar todos os argumentos que um determinado programa contém, e é quase uma convenção que todo utilitário de linha de comando possua tal argumento.

```
natanael@ubuntu:~/Desktop$ file -h
```

```
Usage: file [-bchikLLNnprsvz0] [--apple] [--mime-encoding] [--mime-type]
        [-e testname] [-F separator] [-f namefile] [-m magicfiles] file ...
file -C [-m magicfiles]
file [--help]
```

```
natanael@ubuntu:~/Desktop$
```

Já o segundo retorna informações mais completas, é o *man* seguido do utilitário que desejamos obter ajuda. O terminal irá exibir uma espécie de manual detalhado e interativo do programa em questão.

```

natanael@ubuntu: ~/Desktop
FILE(1) BSD General Commands Manual FILE(1)
NAME
  file - determine file type
SYNOPSIS
  file [-bchikLLNnprsvz0] [--apple] [--mime-encoding] [--mime-type]
        [-e testname] [-F separator] [-f namefile] [-m magicfiles]
        [-R maxrecursion] file ...
  file -C [-m magicfiles]
  file [--help]
DESCRIPTION
  This manual page documents version 5.14 of the file command.

  file tests each argument in an attempt to classify it. There are three
  sets of tests, performed in this order: filesystem tests, magic tests,
  and language tests. The first test that succeeds causes the file type
  to be printed.

  The type printed will usually contain one of the words text (the file
  contains only printing characters and a few common control characters
  Manual page file(1) line 1 (press h for help or q to quit)

```

Caso quisermos saber exatamente onde os arquivos que compõem determinado utilitário estão localizados, podemos utilizar o comando *whereis*.

```
natanael@ubuntu:~$ whereis cp
```

```
cp: /bin/cp /usr/share/man/man1/cp.1.gz
```

## 4. Operações de diretório

Além do comando *pwd*, que informa em qual diretório estamos, podemos verificar quais pastas estão presentes no diretório atual utilizando o comando *ls*:

```
natanaelnovo@ubuntu:/home/natanael$ ls
Desktop Downloads Music Public Videos
Documents examples.desktop Pictures Templates
```

Existem também alguns argumentos que facilitam o uso deste comando. O uso do *-lh*, por exemplo, nos exhibe as pastas com seus respectivos tamanhos, datas de modificação e permissões.

```
natanaelnovo@ubuntu:/home/natanael$ ls -lh
total 44K
drwxr-xr-x 2 natanael natanael 4.0K Feb  3 07:16 Desktop
drwxr-xr-x 2 natanael natanael 4.0K Feb  2 15:58 Documents
drwxr-xr-x 2 natanael natanael 4.0K Feb  2 16:42 Downloads
-rw-r--r-- 1 natanael natanael 8.8K Feb  2 15:52 examples.desktop
drwxr-xr-x 2 natanael natanael 4.0K Feb  2 15:58 Music
drwxr-xr-x 2 natanael natanael 4.0K Feb  2 15:58 Pictures
drwxr-xr-x 2 natanael natanael 4.0K Feb  2 15:58 Public
drwxr-xr-x 2 natanael natanael 4.0K Feb  2 15:58 Templates
drwxr-xr-x 2 natanael natanael 4.0K Feb  2 15:58 Videos
```

Para acessar uma dessas pastas, podemos utilizar o comando *cd* seguido da pasta que desejarmos acessar – assim como no Windows.

```
natanaelnovo@ubuntu:/home/natanael$ cd Desktop
natanaelnovo@ubuntu:/home/natanael/Desktop$
```

Se quisermos voltar uma pasta, podemos utilizar *cd ..*, ao passo que podemos voltar ao diretório raiz utilizando *cd /*.

```
natanaelnovo@ubuntu:/home/natanael/Desktop$ cd ..
natanaelnovo@ubuntu:/home/natanael$ cd /
natanaelnovo@ubuntu:/$
```



Para criar novas pastas, podemos usar o comando `mkdir` seguido do nome que desejamos atribuir para a pasta. Note que caso usarmos um espaço entre cada nome, podemos criar várias pastas. Por exemplo:

```
natanael@ubuntu:~/Desktop$ mkdir pasta
natanael@ubuntu:~/Desktop$ ls
pasta
```

Em contrapartida, podemos deletar pastas vazias utilizando o comando *rmdir*.

```
natanael@ubuntu:~/Desktop$ rmdir pasta
natanael@ubuntu:~/Desktop$ ls
natanael@ubuntu:~/Desktop$
```

Para deletar uma pasta com arquivos, basta utilizar o comando `rm` com o atributo `-rf`. No exemplo abaixo, criamos uma pasta na área de trabalho, verificamos que ela possui arquivos e em seguida prosseguimos com sua exclusão.

```
natanael@ubuntu:~/Desktop$ ls
pastafiles
natanael@ubuntu:~/Desktop$ cd pastafiles
natanael@ubuntu:~/Desktop/pastafiles$ ls
Untitled Document  Untitled Document 2
natanael@ubuntu:~/Desktop/pastafiles$ cd ..
natanael@ubuntu:~/Desktop$ rm pastafiles -rf
natanael@ubuntu:~/Desktop$ ls
natanael@ubuntu:~/Desktop$
```

Caso quisermos criar novos arquivos vazios, podemos fazê-los utilizando o comando `touch` seguido pelo nome do arquivo. Note que, se não especificarmos a extensão, o arquivo será criado na forma vazia.

```
natanael@ubuntu:~/Desktop$ touch arquivo
natanael@ubuntu:~/Desktop$ touch arquivo.html
natanael@ubuntu:~/Desktop$ touch arquivo.jpg
natanael@ubuntu:~/Desktop$ ls
arquivo arquivo.html arquivo.jpg
```

A cópia de arquivos pode ser feita utilizando o comando *cp*, seguido do arquivo original e do nome de destino do arquivo que queremos copiar.

```
natanael@ubuntu:~/Desktop$ cp arquivo.html copia.html
```

```
natanael@ubuntu:~/Desktop$ ls
```

```
arquivo arquivo.html arquivo.jpg copia.html
```

Ainda é possível obter informações sobre determinado arquivo utilizando o comando *file*.

```
natanael@ubuntu:~/Desktop$ file copia.html
```

```
copia.html: ASCII text
```

## 5. Programas

Assim como no Windows, é possível realizar a instalação de novos programas em uma distribuição Linux. Porém, tais programas dificilmente serão obtidos na forma de instaladores.

Inicialmente, eles eram instalados a partir da compilação dos próprios arquivos do código fonte (note a presença da ideologia do software livre). No entanto, tal processo pode ser trabalhoso, em especial para novos usuários.

Por essa razão, foi criado o *apt*, utilitário que permite a instalação quase automática de tais programas. Ao solicitar uma instalação, o programa procura o local para realizar o download entre vários repositórios na Internet e realiza a instalação.

O primeiro passo é rodar o comando *apt-get update*, que faz com que o programa verifique todos os repositórios disponíveis e baixe a lista dos pacotes disponíveis em cada um, criando um banco de dados no qual cada solicitação pode ser relacionada com um endereço, que contém a versão mais recente. Recomendo executar o comando periodicamente, com a permissão de administrador.

```
natanael@ubuntu:~$ sudo apt-get update
Get:1 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Ign http://us.archive.ubuntu.com trusty InRelease
...
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Fetched 2,928 kB in 23s (123 kB/s)
Reading package lists... Done
```

Em seguida, podemos realizar a instalação de programas utilizando o comando *apt-get install*, seguido do programa que desejamos. Por exemplo, para instalar o GIMP – famoso editor de imagens, devemos rodar:

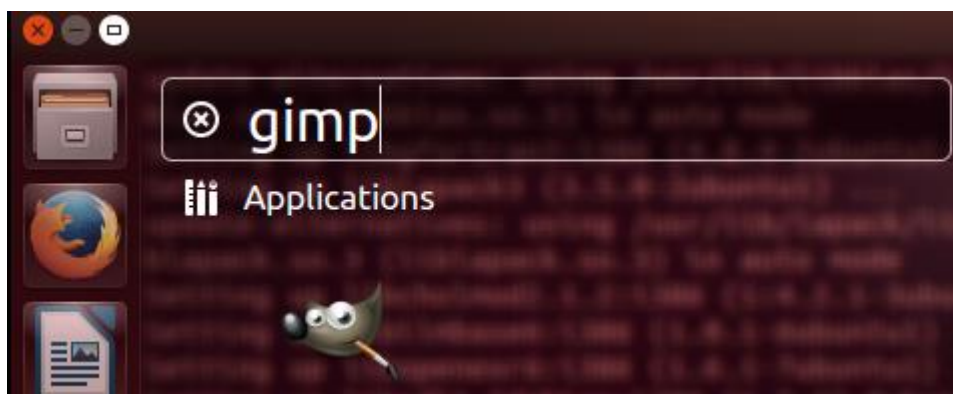
```
natanael@ubuntu:~$ sudo apt-get install gimp
```

A execução do comando fará com que o programa seja baixado, exibindo todo o processo de download e instalação na tela, além dos repositórios escolhidos.

Uma vez instalado, a maioria dos programas pode ser chamada apenas por um comando de terminal. Por exemplo, o comando abaixo fará com que o GIMP seja iniciado.

```
natanael@ubuntu:~$ gimp
```

Em caso negativo, é ainda possível buscar pelo programa na aba de busca, assim como no Windows.



É ainda possível localizar o programa instalado (ou qualquer outro pacote) com o já estudado comando *whereis*. Isso permite que você execute os arquivos direto do diretório em questão.

```
natanael@ubuntu:~$ whereis gimp
```

```
gimp: /usr/bin/gimp /etc/gimp /usr/lib/gimp /usr/bin/X11/gimp /usr/share/gimp  
/usr/share/man/man1/gimp.1.gz
```

Se quisermos desinstalar um programa, podemos fazê-lo com o comando *apt-get remove*. Ele irá realizar a mesma operação de instalação, mas de forma inversa.

```
natanael@ubuntu:~$ sudo apt-get remove gimp
```

Nem sempre os programas que queremos instalar estão nos repositórios oficiais de nossa distribuição. Isso não é um problema. Justamente por essa razão, existem os chamados PPAs, ou Personal Package Archives, que são servidores na internet contendo programas de terceiros.

Vamos tomar como exemplo a ferramenta Hollywood, um programa que exibe uma tela semelhante às utilizadas por hackers em filmes. ~~Você dificilmente encontrará outra utilidade para ele além de impressionar seus amigos, mas não deixa de ser algo legal.~~

Como o Hollywood não está presente nos repositórios oficiais, precisamos adicionar seu repositório, que é *ppa:hollywood/ppa*. Faremos

isso utilizado o comando `sudo apt-add-repository` seguido do repositório que queremos adicionar, no caso:

```
natanael@ubuntu:~/Desktop$ sudo apt-add-repository ppa:hollywood/ppa
```

O Terminal irá confirmar se você realmente deseja realizar tal ação e, em caso afirmativo, adicionará o repositório.

Em seguida, devemos atualizar a lista de repositórios com o `apt-get update`. Daí o programa poderá ser instalado normalmente utilizando o `apt-get install`.

```
natanael@ubuntu:~/Desktop$ sudo apt-get install byobu hollywood
```

O resultado, após sua execução, será uma tela como essa. Leve seu notebook para um shopping, rode o programa, comece a digitar no seu melhor estilo “Matrix” e observe a mágica acontecer.

```

File Edit View Search Terminal Help
:0-part5 -> .././sda5
  by-uuid
  | e067da25-351d-40f7-a3ee-4f3
077feaccf -> .././sda5
  | f52b9b00-2065-4aac-926d-cd6
396881d81 -> .././sda1
3 directories, 7 files

d libaudio2:i386 1.9.4-1
2016-02-17 23:18:11 configure libxt
st6:i386 2:1.2.2-1 <none>
2016-02-17 23:18:11 status unpacked
libxtst6:i386 2:1.2.2-1
2016-02-17 23:18:11 status half-con
figured libxtst6:i386 2:1.2.2-1
2016-02-17 23:18:11 status installe
d libxtst6:i386 2:1.2.2-1

Q V
" j f |
E * 0 0 | X B |
; I K [ | 8 0 |
/ [ Y | \ : W r |
D 2 n . | r : A H 2 |
n 4 8 . 8 . r ( T ) 0 W |
+ J # H & ( Y u S B [

32MiB Speedometer 2.8
/s
TX: eth0 0 B/s 0 B/s
172 B/s
32MiB
/s
RX: eth0 0 B/s 0 B/s
2.23 KiB/s

EREMOTEIO 121 Remote I/O error
EDEADLOCK 35 Resource deadlock avoid
ed
ENODEV 19 No such device
ERFKILL 132 Operation not possible d
ue to RF-kill
OVERFLOW 75 Value too large for def
ined data type

A: 127.4 V: 209.9 A-V:-82.443 ct:-

CPU[|100.0%] Tasks: 147, 180
Mem[649/1000] Load average: 5.
Swp[ 0/1021M] Uptime: 01:17:15

PID USER PRI NI VIRT RES
2084 natanael 20 0 263M 56984
10695 natanael 39 19 3020 2280
F1help F2setup F3Search F4Filter F5D

wanfitkuct (wan-fit-kuct) whiskey-alfa-november-foxtrot-india-tango-kilo-uniform-charlie-tango
lvdonJeb (lv-don-Jeb) india-victor-delta-oscar-november-Juliett-echo-bravo
EndIdnuch, (End-ld-nuch-COMMA) Echo-november-delta-india-delta-november-uniform-charlie-hotel-COMMA
letckJor (Letck-Jor) Lima-echo-india-charlie-kilo-Juliett-oscar-romeo
tytctot8 (Tyt-ct-ot-EIGHT) Tango-yankee-tango-tango-oscar-charlie-tango-oscar-tango-EIGHT
lyborEip (lyb-ort-Eip) Lima-yankee-bravo-oscar-romeo-tango-Echo-india-papa
natVoghYew (nat-Vogh-yew) mike-alfa-tango-Victor-oscar-golf-hotel-yankee-echo-whiskey
ralesanFu (ral-es-an-Fu) romeo-alfa-lima-echo-sierra-alfa-mike-Foxtrot-uniform
Cityyvtu (Cit-yuv-tu) Charlie-India-tango-yankee-uniform-victor-tango-uniform

u* 14.04 0:--: 1:hollywood* 261!! 1h17m 8.12 3.1GHz 1000M61% 2017-02-04 20:01:02

```

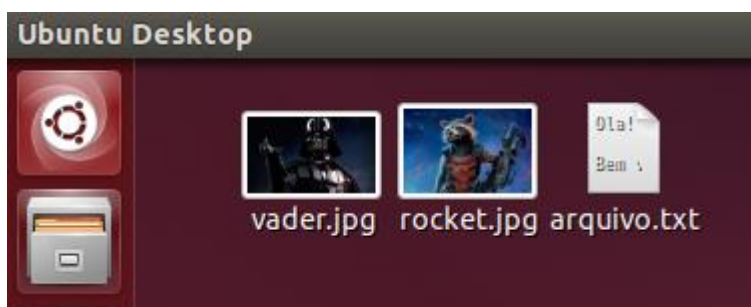
## 6. Compactação de arquivos

Ao usar uma distribuição Linux, é comum nos depararmos com arquivos unidos ou compactados, apresentando uma mesma extensão, tal como `.tar`, `.bz2` ou `.gz`.

No Linux, todos os utilitários que permitem compactar a descompactar arquivos já são nativos na maioria das distribuições, não sendo necessário nenhum processo adicional de instalação.

Antes de prosseguir para a compactação, estudaremos o processo de arquivamento, isto é, de agrupar vários arquivos em um só. O formato mais popular para tal processo é o `.tar`, e o arquivamento pode ser realizado com o comando `tar -cf`, seguido do arquivo de destino e dos arquivos que serão agrupados.

Para demonstrar essas ferramentas, usaremos um conjunto contendo três arquivos, sendo um de texto e duas imagens.



Iremos agrupá-los em um novo arquivo, denominado `final.tar`. O comando utilizado é:

```
natanael@ubuntu:~/Desktop$ tar -cf final.tar vader.jpg rocket.jpg arquivo.txt
```

O qual imediatamente irá retornar o arquivo que desejamos. Note que ele não está compactado, mas apenas agrupado.



Se quisermos novamente extrair estes arquivos, podemos realizar a operação com o comando `tar -xf`, dessa forma:

```
natanael@ubuntu:~/Desktop$ tar -xf final.tar
```

Já para a compactação, estudaremos como realiza-la de três formas distintas: com o *bzip2* e o *gzip* para Linux e com o *zip* caso queiramos compartilhar nosso arquivo com alguém que não usa Linux.

É importante salientar que, no processo de compactação, o arquivo original será substituído pelo arquivo compactado. Caso queira realizar a compressão em mais de um arquivo, una-os em um *.tar*.

O *bzip2* é capaz de comprimir mais que o *gzip*, porém consome mais memória no processo. Para compactar com este utilitário, devemos utilizar o comando *bzip2* seguido do nome do arquivo:

```
natanael@ubuntu:~/Desktop$ bzip2 rocket.jpg
```

Já para descompactar, podemos utilizar o mesmo comando, desta vez com o argumento *-d* seguido do nome do arquivo.

```
natanael@ubuntu:~/Desktop$ bzip2 -d rocket.jpg.bz2
```

Note que esse processo apenas compacta os arquivos, mas não realiza o arquivamento destes. Caso quiser compactar vários arquivos de uma única vez, primeiro una-os com o *tar*. O processo para extração de um *.tar* será mostrado adiante.

A sequência de comandos para o *gzip* é exatamente a mesma, apenas deve-se trocar o comando, da seguinte forma:

```
natanael@ubuntu:~/Desktop$ gzip rocket.jpg
```

```
natanael@ubuntu:~/Desktop$ gzip -d rocket.jpg.gz
```

O comando *zip*, além de garantir a portabilidade para Windows, também é um arquivador. Para compactarmos, basta utilizar o comando *zip* seguido do arquivo final, por sua vez seguido do nome dos arquivos que se deseja compactar.

```
natanael@ubuntu:~/Desktop$ zip final.zip arquivo.txt vader.jpg rocket.jpg
```

Já para descompactar, utilizamos o comando *unzip*:

```
natanael@ubuntu:~/Desktop$ unzip final.zip
```

Para compactar um arquivo *.tar*, devemos utilizar um dos comandos já estudados (*bzip2* ou *gzip*) da forma já vista.

```
natanael@ubuntu:~/Desktop$ bzip2 final.tar
```

```
natanael@ubuntu:~/Desktop$ gzip final.tar
```



Caso realizarmos a descompactação com o argumento `-d`, obteremos apenas o arquivo `.tar`, ainda agrupado. Para evitar que isso ocorra, devemos utilizar novamente o comando `tar` seguido dos argumentos `-jxvf` para arquivos `.bz2` e `-zxvf` para arquivos `.gz`.

```
nataael@ubuntu:~/Desktop$ tar -jxvf final.tar.bz2
```

```
nataael@ubuntu:~/Desktop$ tar -zxvf final.tar.gz
```

## 7. Editor de texto

Ao contrário dos sistemas da Microsoft, que utilizam arquivos binários, os arquivos de configuração do Linux são salvos em formato de texto editáveis. Isso possui várias vantagens, entre elas a facilidade de edição destes.

Caso seja necessário alterar estes arquivos, você pode usar desde editores de texto complexos, com interface gráfica, até ferramentas simples de linha de comando que realizam funções simples – mas que podem ser úteis em uma emergência.

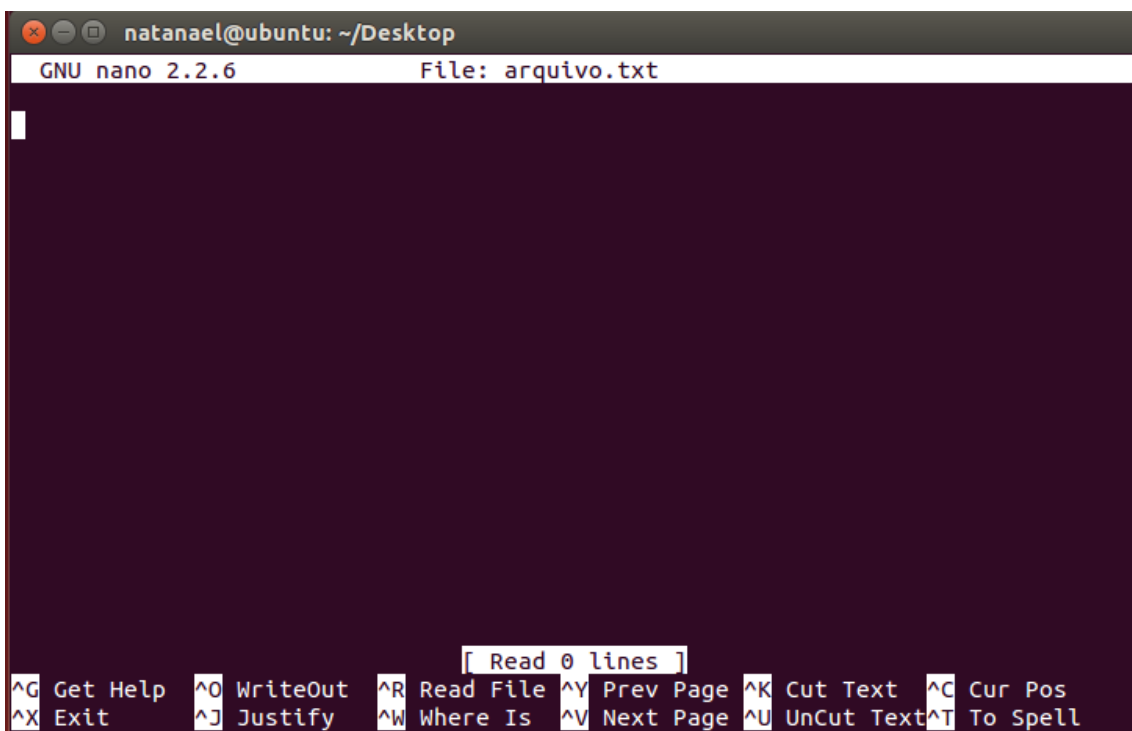
Para demonstrarmos tal uso, primeiramente criaremos um arquivo de texto em branco com comandos que já dominamos:

```
natanael@ubuntu:~/Desktop$ touch arquivo.txt
```

Em seguida, iniciaremos o editor utilizando o comando *nano*, seguido do nome do arquivo.

```
natanael@ubuntu:~/Desktop$ nano arquivo.txt
```

Essa é a interface do programa.



```
natanael@ubuntu: ~/Desktop
GNU nano 2.2.6      File: arquivo.txt
[ Read 0 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Ele funciona basicamente através de atalhos do teclado. Podemos editar o texto do arquivo livremente (através da parte superior) e realizar operações simples – tais como salvar, renomear e copiar – através de

comandos com a tecla CTRL, os quais estão descritos no rodapé do programa e são:

- CTRL-X: sai do editor.
- CTRL-R: lê um arquivo, permitindo adicioná-lo ao arquivo atual.
- CTRL-C: mostra a posição atual do cursor.
- CTRL-K: recorta o texto.
- CTRL-U: cola o texto.
- CTRL-S: salva o arquivo.
- CTRL-T: verifica a ortografia do texto.
- CTRL-W: realiza buscas no texto.
- CTRL-A: posiciona o cursor no início da linha.
- CTRL-E: posiciona o cursor no fim da linha.
- CTRL-G: exhibe as informações de ajuda.

Dessa forma basta escrever ou alterar o arquivo de texto e salvá-lo utilizando os comandos do editor.

## 8. Monitoramento de processos

Em Linux, processo é o nome dado à uma instância de um software em execução, ou seja, um programa que está sendo utilizado. No Windows, são as conhecidas “tarefas”.

Podemos obter a listagem de todos os processos em execução através do comando *top*. Ele exibirá uma interface semelhante ao conhecido gerenciador de tarefas do Windows, na qual podemos verificar a quantidade de memória e CPU que cada processo está consumindo, assim como seus respectivos usuários, entre outras informações.

```

natanael@ubuntu:~$ top
top - 18:28:28 up 2:08, 2 users, load average: 0.05, 0.12, 0.17
Tasks: 217 total, 3 running, 214 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.7 us, 1.7 sy, 0.0 ni, 95.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 1024228 total, 814240 used, 209988 free, 11472 buffers
KiB Swap: 1046524 total, 10548 used, 1035976 free. 307684 cached Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1289 root        20   0 155536 40984 20308 S   1.7   4.0   0:38.97 Xorg
 2098 natanael    20   0 330484 94340 59320 R   1.7   9.2   0:38.27 compiz
 3389 natanael    20   0 136028 28404 22056 S   1.3   2.8   0:00.63 gnome-termi+
   7 root        20   0   0      0      0 R   0.3   0.0   0:01.49 rcu_sched
  849 root        20   0  19176  6312  5952 S   0.3   0.6   0:00.86 thermald
 1989 natanael    20   0 116352 25520 18228 S   0.3   2.5   0:01.34 ibus-ui-gtk3
 3411 natanael    20   0   5540  2720  2312 R   0.3   0.3   0:00.05 top
   1 root        20   0   4452  3160  2576 S   0.0   0.3   0:03.06 init
   2 root        20   0   0      0      0 S   0.0   0.0   0:00.01 kthreadd
   3 root        20   0   0      0      0 S   0.0   0.0   0:00.60 ksoftirqd/0
   5 root         0 -20   0      0      0 S   0.0   0.0   0:00.00 kworker/0:0H
   8 root        20   0   0      0      0 S   0.0   0.0   0:00.00 rcu_bh
   9 root        rt    0   0      0      0 S   0.0   0.0   0:00.00 migration/0
  10 root        rt    0   0      0      0 S   0.0   0.0   0:00.06 watchdog/0
  11 root         0 -20   0      0      0 S   0.0   0.0   0:00.00 khelper

```

A interpretação dos dados contidos nessa tabela envolve o conhecimento de algumas variáveis. Cada coluna da tabela apresenta uma informação sobre o processo em questão. São elas:

- PID: identificador do processo.
- USER: login do proprietário do processo.
- PR: prioridade do processo.
- NI: valor “NICE” do processo.
- VIRT: quantidade de memória virtual utilizada pelo processo.
- RES: quantidade de memória física utilizada pelo processo.
- SHR: quantidade de memória compartilhada pelo processo.
- S: indica o status do processo, sendo S para “sleep”, R para “running” e Z para “zombie”.
- TIME+: tempo total de atividade do processo

- %CPU: porcentagem de CPU utilizada pelo processo.
- %MEM: porcentagem de memória virtual utilizada pelo processo.
- COMMAND: comando utilizado para iniciar o processo.

Caso quisermos forçar o encerramento de algum processo, podemos usar o comando *kill* seguido do PID do processo em questão. Por exemplo, se sabemos que o PID do Mozilla Firefox é 3441, podemos encerrá-lo com:

```
natanael@ubuntu:~$ kill 3441
```

É ainda possível realizar o encerramento manual de tais processos utilizando o comando *xkill* e logo em seguida clicando na janela cujo processo queremos encerrar.