

Table of Contents

- [1 Introduction to Unix, Part 1 - Intro to Unix and Terminal](#)
 - [1.1 Summary of the course today](#)
- [2 What is the command line and why should we use it?](#)
- [3 What is Unix?](#)
- [4 The Unix Approach](#)
- [5 Linux, MacOS, and Unix](#)
- [6 TAs and support](#)
- [7 Hands on Linux: let's login!](#)
 - [7.1 While the computer loads: make yourself comfortable!](#)
 - [7.2 This is how the desktop should look like, after login](#)
 - [7.3 The Linux desktop](#)
- [8 Let's open the terminal](#)
 - [8.1 How does a terminal looks like?](#)
 - [8.2 More definitions](#)
- [9 Launching applications from the Terminal](#)
 - [9.1 Running applications in the background](#)
 - [9.2 Killing an application](#)
- [10 Your first command: ls](#)
 - [10.1 Anatomy of a command](#)
 - [10.2 listing all files and their details \(ls -l\)](#)
 - [10.3 Listing hidden files \(ls -a; ls -la\)](#)
 - [10.4 Combining multiple ls parameters together](#)
 - [10.5 File permissions and output of ls -l](#)
 - [10.6 Arguments](#)
 - [10.6.1 Exercise - ls](#)
 - [10.7 Command-line tip: the Tab key and auto-completion](#)
- [11 How to get documentation of a command?](#)
 - [11.1 ls --help](#)
 - [11.2 The man page of a command](#)
 - [11.3 The man page of ls](#)
 - [11.4 Navigating a man page](#)
 - [11.5 Other useful sections in a man page](#)
 - [11.6 Getting documentation via info](#)
- [12 Exercise Time!](#)
- [13 Getting help from Internet](#)
- [14 How can I use a Unix system at home/lab?](#)
 - [14.1 The Linux distributions](#)
 - [14.2 What is the difference between Linux Distributions?](#)
- [15 Examples of Linux Distributions](#)
 - [15.0.1 Ubuntu](#)
- [16 What if I don't want to install a new operating system?](#)
 - [16.1 Windows](#)
 - [16.2 Mac](#)
 - [16.3 Linux](#)
- [17 Time for a break!](#)

Introduction to Unix, Part 1 - Intro to Unix and Terminal

Welcome to the **Programming for Evolutionary Biology workshop!!**

Giovanni M. Dall'Olio. Data Strategy and Design, GSK. March 2020.

Quick link to slides online: <https://tinyurl.com/evop-unix> (<https://tinyurl.com/evop-unix>)

All materials available here: https://github.com/dalloliogm/peb_unix_intro/archive/master.zip
(https://github.com/dalloliogm/peb_unix_intro/archive/master.zip).

How to use these slides: Press Space to get to the next slide. Use arrows to navigate the subsections.

Summary of the course today

- **Morning first half:** Basic Terminal commands, First Login to Linux
- Morning second half: Login to a Remote Unix server, browsing file contents
- Afternoon first half: Finding patterns in a file with grep; piping commands; cut, sort and uniq
- Afternoon second half: awk and sed; makefiles

What is the command line and why should we use it?

The Command Line allows to interact with a computing system, using text commands instead of a graphical system.

There is a steep learning curve at the beginning, but it is a very powerful approach.

Compared to Graphical interfaces, it allows to execute certain tasks quicker and using fewer resources.

What is Unix?

Unix is the name of an operating system created in the '80s, which became popular for a variety of reasons:

- it was free for academic use
- it ran on any computer hardware independently of the maker
- it introduced a novel approach to programming and file managing

The Unix Approach

Unix introduced some important principles to the design of an operating system:

- Make each program do one thing well.
- Expect the output of every program to become the input to another, as yet unknown, program.
- Work on file streams, reading one line at a time.

These principles have been written a long time ago, but learning them will make you a better programmer.

Linux, MacOS, and Unix

The original Unix system does not run on modern computers, but we can use Unix-like systems such as Linux and MacOSX.



TAs and support

Some practical information:

- There are six Teaching Assistants (TAs), to help you, approximately one every five students.
- Use sticky notes during the exercises: a green note when you finished the exercise, and a red note if you need help
- We will have a second projector where a TA will shows the commands to type

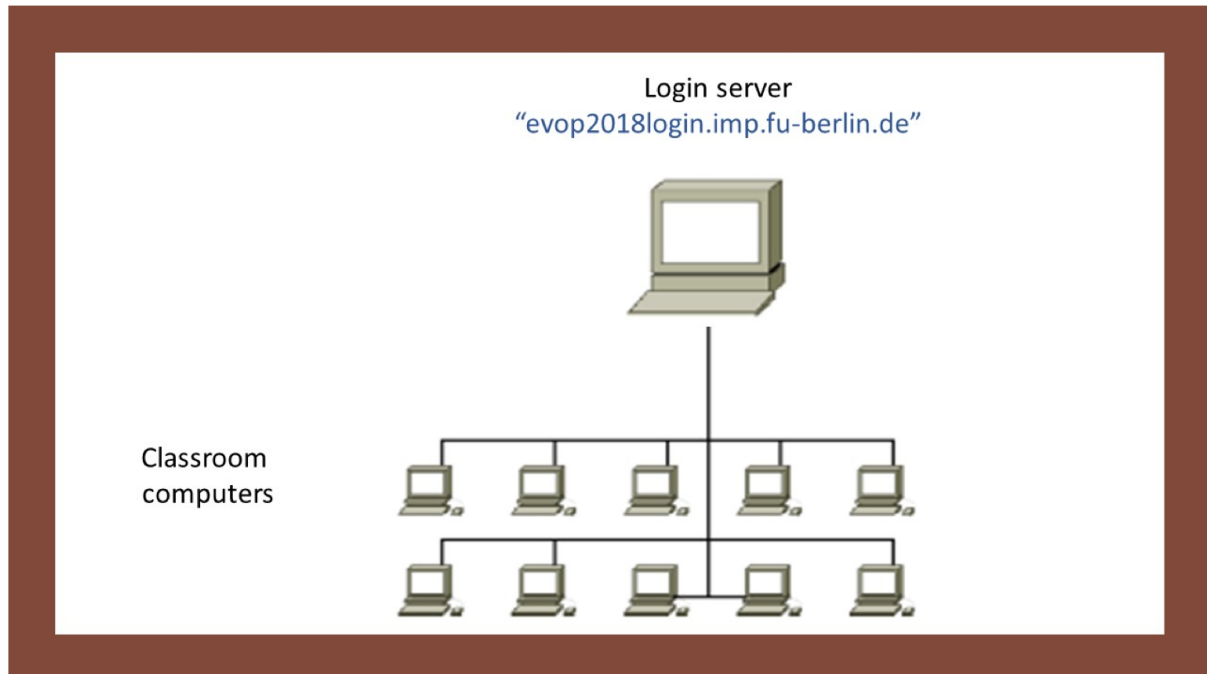
Hands on Linux: let's login!

During this course we will use two sets of computers:

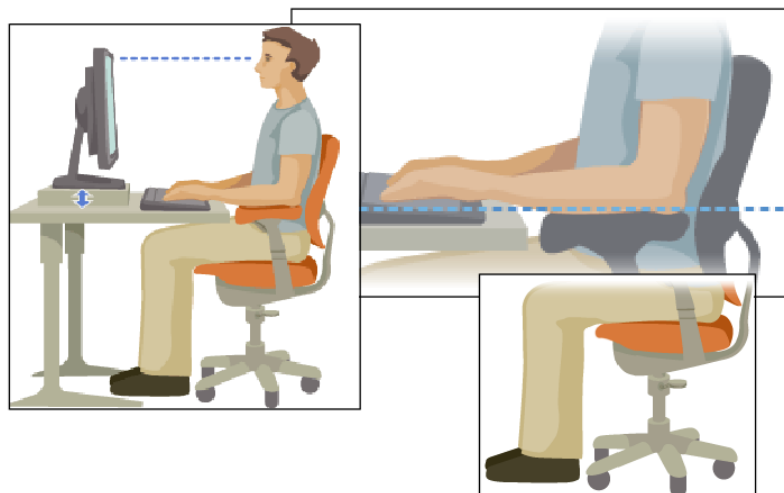
- the classroom computers (in front of you)
- A remote login server: `'evop-login.imp.fu-berlin.de'`.

Let's focus on the classroom computers for now.

Check the **login information: Password and Computer infrastructure** that has been provided to you.

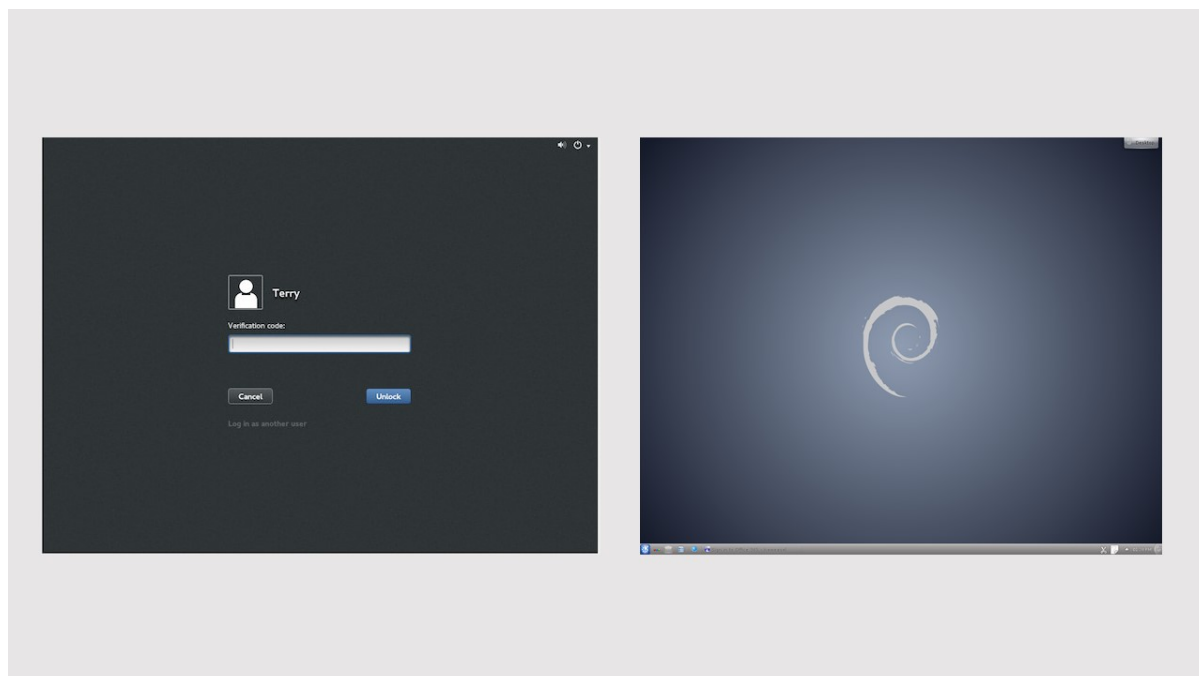


While the computer loads: make yourself comfortable!



This is how the desktop should look like, after login

Once logged in, you will see a desktop very similar to a Windows or Mac environment:



(The wallpaper and desktop may be different)

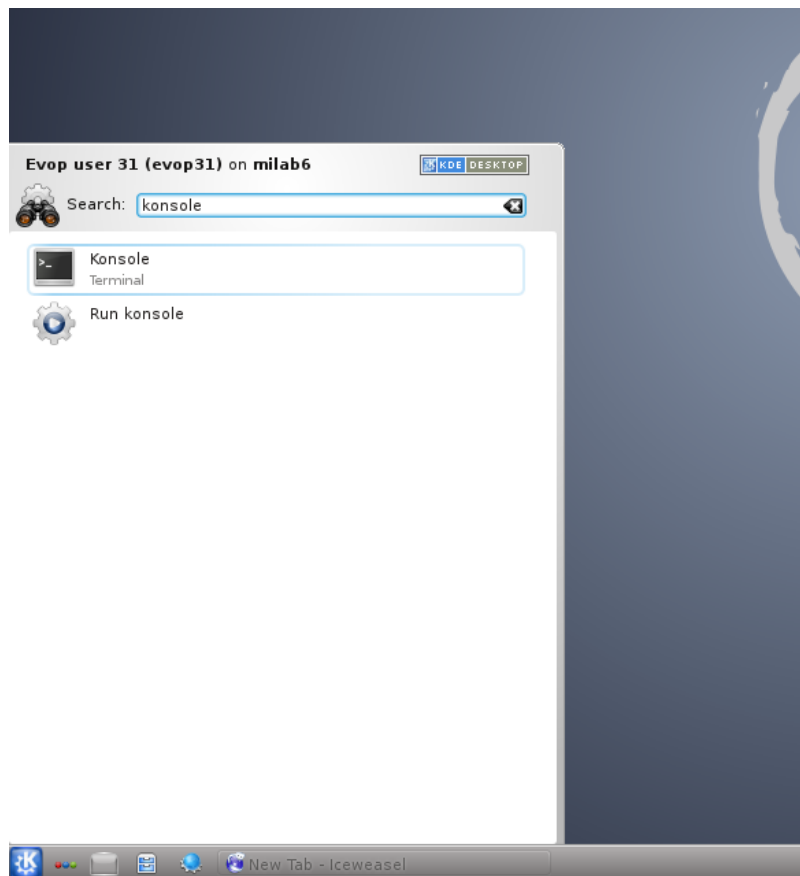
The Linux desktop

The Linux interface may be different to Windows and Mac, but it should not be difficult to use:

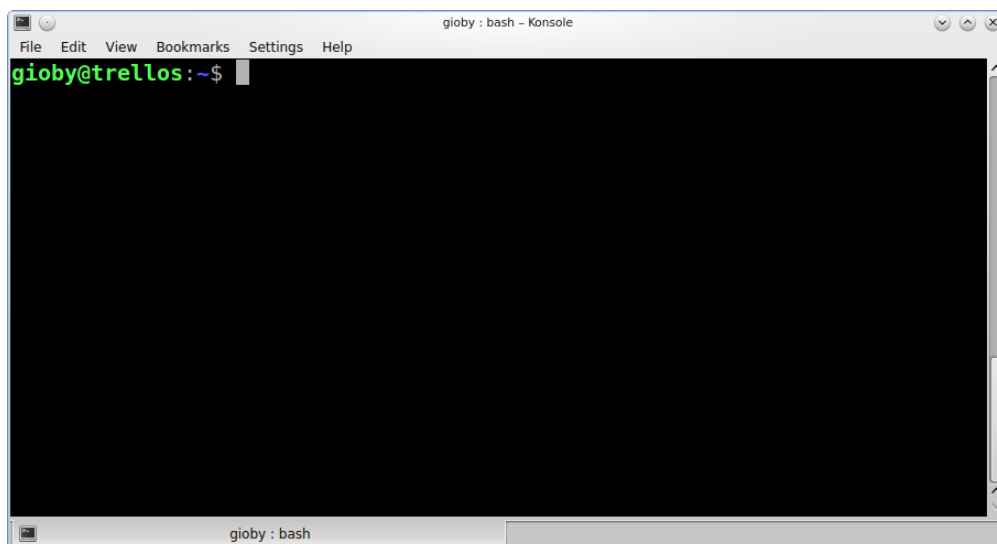
- you have a Launch menu, from where you can access all the applications installed
- most applications are similar:
 - `firefox` or `chrome` for web browsing
 - `kate` , `gedit` for editing text
 - `libreoffice` for documents, presentations, spreadsheets

Let's open the terminal

Open the Launch menu and search for `terminal` or `konsole`



How does a terminal looks like?



More definitions

Linux :

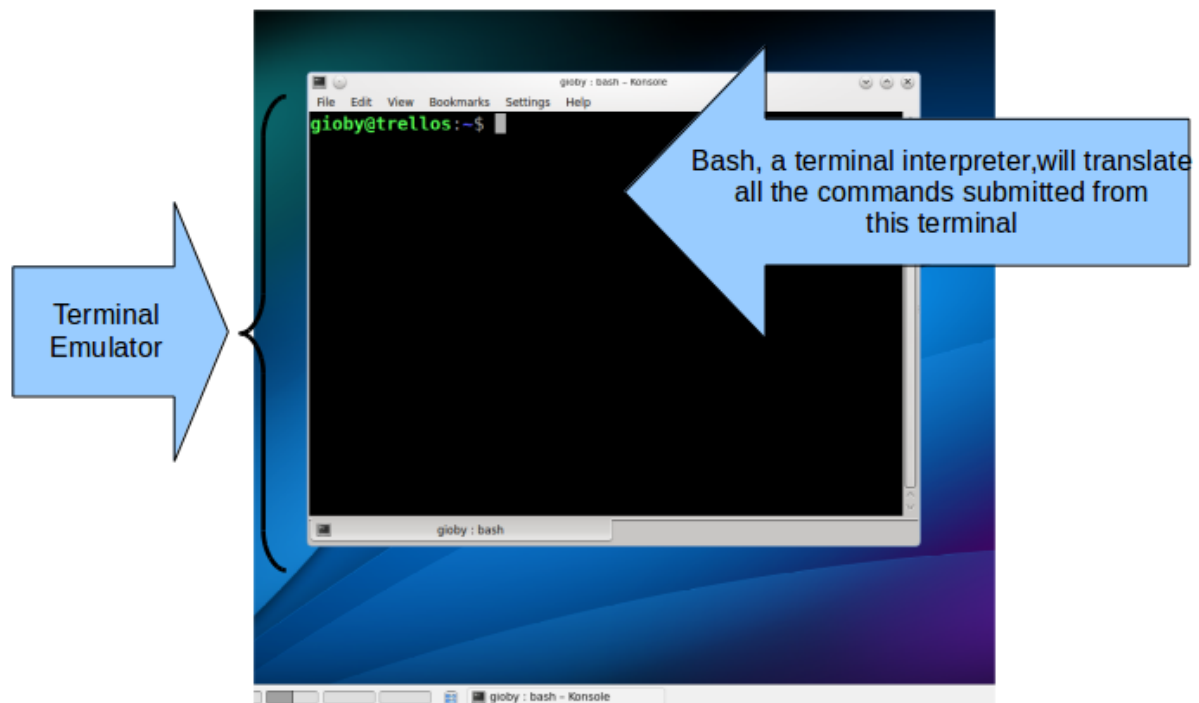
A "descendant" and free modern version of Unix, developed by an Open Source Community

Terminal :

A software that allows to input commands to the computer, by typing them rather than point-and-click

Bash :

A command-line interpreter, e.g. a software that interprets the commands given from the terminal, and execute them.



Launching applications from the Terminal

The terminal can be used any application installed in the computer, by typing its name instead of clicking on it.

For example, type `firefox` to launch a web browser:

```
$: firefox
```

* if this doesn't work, try: `google-chrome`

Running applications in the background

Some applications "freeze" the terminal when executed. For example, let's launch `kate`, a text editor

```
$: kate
```

* if this doesn't work, try `gedit`

How to get back control:

- after launching the application, press `CTRL-Z`, then run the `bg` command
- for future cases, before launching the application, add an `&` after the command (`kate &` or `gedit &`)

You can also type `jobs` for a list of all the application running in background.

Killing an application

Sometimes it is useful to `Force-Quit` an application, when it doesn't respond to inputs, or we need to abort it.

If the application is running in the foreground, we can press `CTRL-C` to abort its execution.

If the application is in the background (previous slide), we can use:

- `jobs` (to list all the jobs in background)
- `kill %1` (to kill the first job in background)

Your first command: ls

The most basic command is `ls`, which allows to list all the files in the current folder.

The name is an abbreviation for "list"

Typing `ls` will show all the files in the current directory.

```
evop2018login.imp.fu-berlin.de - PuTTY
Sun Mar 04 16:36:45 102 k00319234@evop2018login:~ $ ls
Desktop Downloads MyDocuments testfile2.txt testfile.txt
Sun Mar 04 16:36:47 103 k00319234@evop2018login:~ $ █
```

Press space or the down key to continue.

Anatomy of a command

A Unix command-line statement is composed of three parts:

- the `command` itself
- `parameters` (to alter the default behaviour of the command)
- `arguments` (target of the command)

listing all files and their details (`ls -l`)

Parameters alter the behaviour of the command.

For example adding `-l` parameter (lower case L) to `ls` will print more details:

```
-bash-4.3$ ls -l
total 28
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Desktop
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Documents
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Downloads
-rw-r--r-- 1 alvaro staff    1 Feb 26 10:58 file.txt
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Images
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Templates
-rw-r--r-- 1 alvaro staff    1 Feb 26 10:58 test1.txt
-bash-4.3$
```

Listing hidden files (ls -a; ls -la)

By convention, hidden files in Unix systems are prefixed by `.`

The `-a` option of `ls` shows all the files, including the hidden ones.

In general home folders contain several hidden configuration files (e.g. `.bashrc` , `.conf` , `.cache`). We can ignore them for the moment.

```
-bash-4.3$ ls -la
total 36
drwxr-xr-x  7 alvaro staff 4096 Feb 26 10:58 .
drwxr-xr-x 85 alvaro staff 4096 Feb 26 15:18 ..
drwxr-xr-x  2 alvaro staff 4096 Feb 26 10:57 Desktop
drwxr-xr-x  2 alvaro staff 4096 Feb 26 10:57 Documents
drwxr-xr-x  2 alvaro staff 4096 Feb 26 10:57 Downloads
-rw-r--r--  1 alvaro staff   1 Feb 26 10:58 file.txt
drwxr-xr-x  2 alvaro staff 4096 Feb 26 10:57 Images
drwxr-xr-x  2 alvaro staff 4096 Feb 26 10:57 Templates
-rw-r--r--  1 alvaro staff   1 Feb 26 10:58 test1.txt
-bash-4.3$
```

Combining multiple ls parameters together

Multiple parameters can be combined together.

For example, `ls -lt`, will print the list of files in the long format (`-l`), and sorted by modification date (`-t`)

```
-bash-4.3$ ls -lt
total 28
-rw-r--r-- 1 alvaro staff    1 Feb 26 10:58 test1.txt
-rw-r--r-- 1 alvaro staff    1 Feb 26 10:58 file.txt
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Templates
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Images
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Downloads
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Desktop
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Documents
-bash-4.3$
```

File permissions and output of ls -l

The first column in the ls -l output describes the permissions for the file (e.g. drwxr-xr-x.)

- character 1 : whether the element is a directory (d) or a file (-)
- characters 2–4 : whether the file is readable (r), writeable (w), or executable (x) by the user who owns it (fedora)
- characters 5–7 : the same, but for the user group (also called fedora)
- characters 8–10 : the same, for any user
- character 11 : security settings (not in all systems)

See `man chmod` to modify file permissions.

```
-bash-4.3$ ls -l
total 28
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Desktop
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Documents
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Downloads
-rw-r--r-- 1 alvaro staff    1 Feb 26 10:58 file.txt
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Images
drwxr-xr-x 2 alvaro staff 4096 Feb 26 10:57 Templates
-rw-r--r-- 1 alvaro staff    1 Feb 26 10:58 test1.txt
-bash-4.3$
```

Arguments

The other component of a command is the argument, e.g. what is the target of the command.

By default `ls` shows the contents of the current directory (also referred as `.`)

To see the contents of another directory, just add its name to the command invocation:

```
ls -l Desktop
```

Exercise - ls

- use `ls` to print the contents of the current folder
- compare this with `ls .`
- Type `ls -l Desktop` to show the contents of the Desktop folder

Command-line tip: the Tab key and auto-completion

We can use the Tab key on the keyboard (usually above Caps Lock) to automatically complete a command or a directory name.

For example, try the following:

```
ls Desk<press Tab key>
```

How to get documentation of a command?

How can we know all the possible options of a command?

There are at least three options:

- `--help`
- `man`
- `info`

ls --help

Most command support a `--help` option (or `-h`, `-help`), which prints a summary of the most common parameters and options for the command:

```
$: ls --help
```

```
Usage: ls [OPTION]... [FILE]...
```

```
List information about the FILES (the current directory by default).
```

```
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
```

```
Mandatory arguments to long options are mandatory for short options to o.
```

```

-a, --all                do not ignore entries starting with .
-A, --almost-all        do not list implied . and ..
    --author              with -l, print the author of each file
-b, --escape             print C-style escapes for nongraphic characters
    --block-size=SIZE     scale sizes by SIZE before printing them;
e.g.,
```

The man page of a command

The **man** command allows to access the manual of a command.

Let's try it:

```
man ls
```

The man page of ls

Most man pages contain at least the following sections:

- NAME: name of the command
- SYNOPSIS: how to use it
- DESCRIPTION/OPTIONS: what the command does, and

```
LS(1)                                User Commands                                LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort.

  Mandatory arguments to long options are mandatory for short options
  too.

  -a, --all
      do not ignore entries starting with .
  -A, --almost-all
      do not list implied . and ..
  --author
      with -l, print the author of each file
  -b, --escape
      print C-style escapes for nongraphic characters
  --block-size=SIZE
      use SIZE-byte blocks. See SIZE format below
  -B, --ignore-backups
      do not list implied entries ending with ~
  -c
      with -lt: sort by, and show, ctime (time of last modification of
      file status information) with -l: show ctime and sort by name
      otherwise: sort by ctime
  -C
      list entries by columns
```

Name of the command

Synopsis (how to use it)
Options in square
brackets are optional

Parameters & arguments

In [2]:

```
man ls | head -n 30
```

LS(1) User Commands
LS(1)

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILES (the current directory by default).

Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
do not ignore entries starting with .

-A, --almost-all
do not list implied . and ..

--author
with -l, print the author of each file

-b, --escape
print C-style escapes for nongraphic characters

Navigating a man page

- Use the arrows keys or PageUp / PageDown to scroll the man page
- Use / followed by a keyword to search text
 - e.g. press /sort<Enter> to search for the word "sort"
- press q to exit

Other useful sections in a man page

- SEE ALSO: references to similar commands
- EXAMPLES: examples of how to use the command

```
File Edit View Search Terminal Help

SEE ALSO
Regular Manual Pages
awk(1), cmp(1), diff(1), find(1), gzip(1), perl(1), sed(1),
sort(1), xargs(1), zgrep(1), mmap(2), read(2), pcre(3),
pcresyntax(3), pcrepattern(3), terminfo(5), glob(7), regex(7).

POSIX Programmer's Manual Page
grep(1p).

TeXinfo Documentation
The full documentation for grep is maintained as a TeXinfo manual.
If the info and grep programs are properly installed at your site,
the command

    info grep

should give you access to the complete manual.

NOTES
GNU's not Unix, but Unix is a beast; its plural form is Unixen.

Manual page grep(1) line 640/664 100% (press h for help or q to quit)
```

```
File Modifica Visualizza Cerca Terminale Aiuto

--version
    output version information and exit

Note, comparisons honor the rules specified by `LC_COLLATE'.

EXAMPLES
comm -12 file1 file2
    Print only lines present in both file1 and file2.

comm -3
    file1 file2  Print lines in file1 not in file2, and vice versa.

AUTHOR
    Written by Richard M. Stallman and David MacKenzie.

REPORTING BUGS
    Report comm bugs to bug-coreutils@gnu.org
-- MOST: *stdin* (33,1) 43%
Press `Q' to quit, `H' for help, and SPACE to scroll.
```


Getting documentation via info

The command `info` allows to access a more complete description of a command.

```
File Edit View Search Terminal Help
File: coreutils.info, Node: ls invocation, Next: dir invocation, Up: Directory listing

10.1 `ls': List directory contents
=====

The `ls' program lists information about files (of any type, including
directories).  Options and file arguments can be intermixed
arbitrarily, as usual.

For non-option command-line arguments that are directories, by
default `ls' lists the contents of directories, not recursively, and
omitting files with names beginning with `.'.  For other non-option
arguments, by default `ls' lists just the file name.  If no non-option
argument is specified, `ls' operates on the current directory, acting
as if it had been invoked with a single argument of `.'.

By default, the output is sorted alphabetically, according to the
locale settings in effect. (1) If standard output is a terminal, the
output is in columns (sorted vertically) and control characters are
output as question marks; otherwise, the output is listed one per line
--zz-Info: (coreutils.info.gz)ls invocation, 58 lines --Top-----
```

Not all commands have an info page. though.

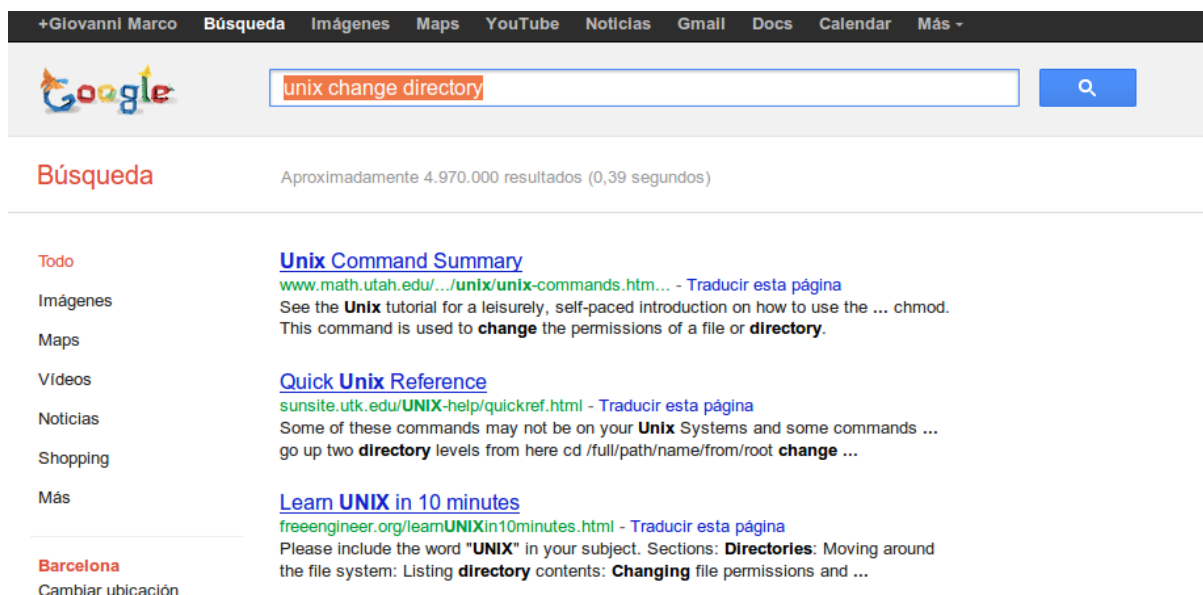
Excercise Time!

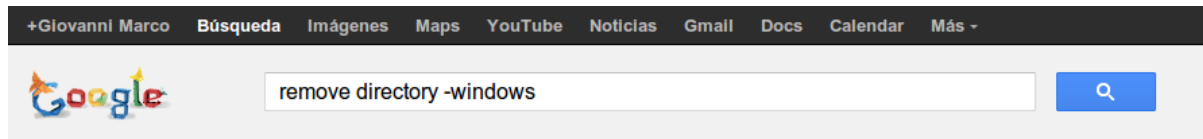
- Open the documentation page for `ls`
- Which parameter can be passed to `ls` to sort files by size?
- How to show the contents of a directory recursively with `ls`?

Getting help from Internet

Some tips to get better results when searching the documentation of a command on Internet:

- add keywords such as `unix`, `bash`, `fedora`
- Use the `-` operator to remove junk results
- if you get an error with a softawre or during an installation, copy and paste the message on google





Búsqueda

Aproximadamente 419.000.000 resultados (0,29 segundos)

Todo

Sugerencia: [Buscar solo resultados en español](#). Puedes especificar el idioma de búsqueda en [Preferencias](#)

Imágenes

Maps

Vídeos

Noticias

Shopping

Más

[Linux and UNIX rm command help](#)

[www.computerhope.com/unix/urm.htm](#) - Traducir esta página

-f, Remove all files (whether write-protected or not) in a directory without prompting ... -r, Recursively **remove directories** and subdirectories in the argument list.

[How to remove directory from \\$PATH](#)

[www.linuxquestions.org > ... > Linux - Newbie](#) - Traducir esta página

19 Mar 2011 – I accidentally added a wrong **directory** to \$PATH. How do I **remove** it?

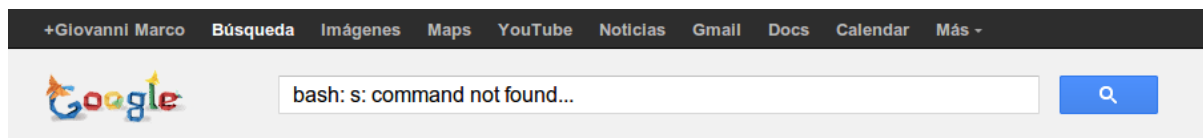
[rm\(1\): remove files/directories - Linux man page](#)

[linux.die.net/man/1/rm](#) - Traducir esta página

If the -I or --interactive=once option is given, and there are more than three files or the -r, -R, or --recursive are given, then rm prompts the user for ...

Barcelona

Cambiar ubicación



Búsqueda

Aproximadamente 2.020.000 resultados (0,44 segundos)

Todo

Sugerencia: [Buscar solo resultados en español](#). Puedes especificar el idioma de búsqueda en [Preferencias](#)

Imágenes

Maps

Vídeos

Noticias

Shopping

Foros de debate

Más

[Bash: command not found!](#)

[www.linuxquestions.org > ... > Linux - General](#) - Traducir esta página

26 Oct 2005 – Hi I'm having a very weird problem. whatever command I type I keep getting "bash: Is: **command not found**". That go for all the command ...

[command not found error while executing a shell script](#) - 25 Jul 2010

[bash: g++: command not found](#) - 8 Jul 2010

[bash script variables](#) - 17 Nov 2004

[bash: <command name> command not found](#) - 14 Jul 2001

[Más resultados de linuxquestions.org >](#)

[Variables](#)

[tldp.org/LDP/Bash-Beginners.../sect_03_02.html](#) - Traducir esta página

Xdefaults XFILESEARCHPATH=/usr/X11R6/lib/X11/%L/%T/%N%C%S:/usr/X11R6/ ...

command not found franky ~> MYVAR1= "2" **bash: 2: command not found** ...

Barcelona

Cambiar ubicación

La Web

Páginas en español

Páginas de España

Páginas extranjeras

traducidas

[cygwin - cywin bash script command not found when called from ...](#)

[stackoverflow.com/.../cywin-bash-script-comma...](#) - Traducir esta página

Una respuesta

Mejor respuesta: `#!/bin/bash echo "Testing" cd "/cygdrive/x/Internal`

`Communications/Riccardo/" filename=htdocs-date +%A`.tar.gz tar cvzf $filename ...`

How can I use a Unix system at home/lab?

The Linux distributions

One option to access a Unix system is to install a "GNU/Linux" operating system on your computer.

However there is not a single "GNU/Linux" operating system, as different groups of people have created different distributions according to different needs or contexts

What is the difference between Linux Distributions?

- The software included by default when you install the system
 - e.g. firefox or chrome, gedit or kate
- Some distributions include only free software, other are less strict
- Some distributions are aimed for new users, while other are designed for "nerds, geeks, specialists....."

Behind the scenes, distributions can differ for:

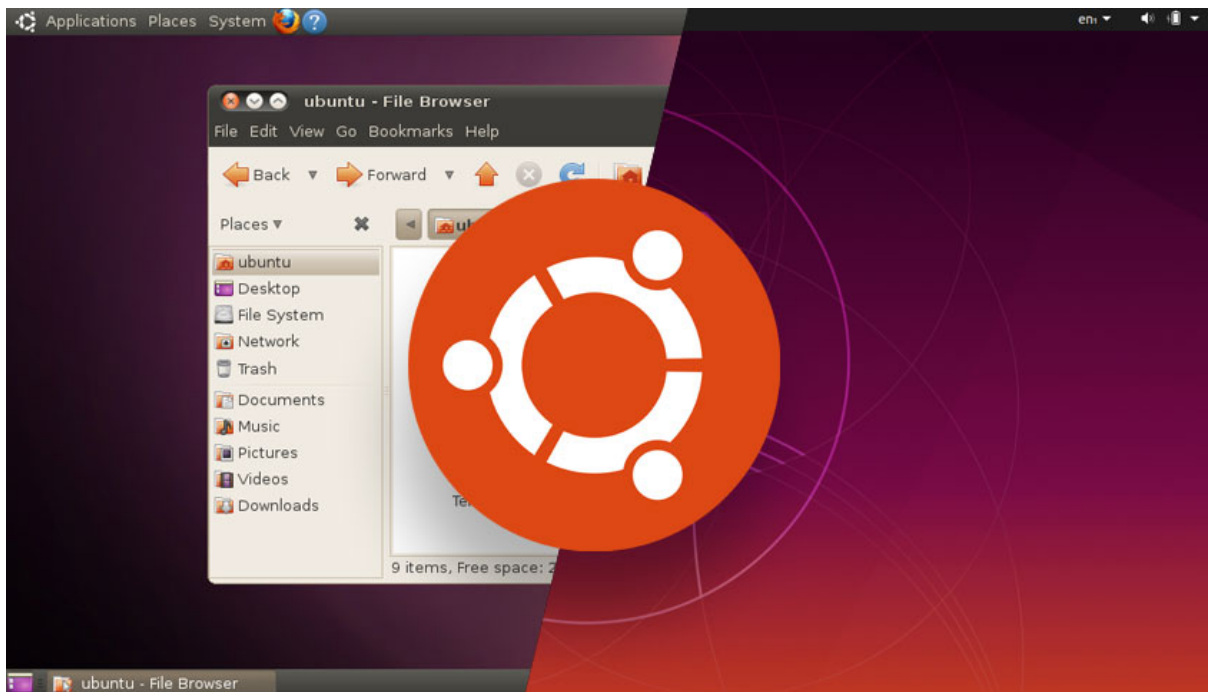
- the system used to install new software
- technical details such as libraries used to compile the software

Examples of Linux Distributions

- Ubuntu : a popular distribution aimed at beginners
- Fedora : another general use distribution

Ubuntu

Ubuntu is one of the most user-friendly Linux distribution. Good for beginners, but also for most users.



<https://ubuntu.com/> (<https://ubuntu.com/>)

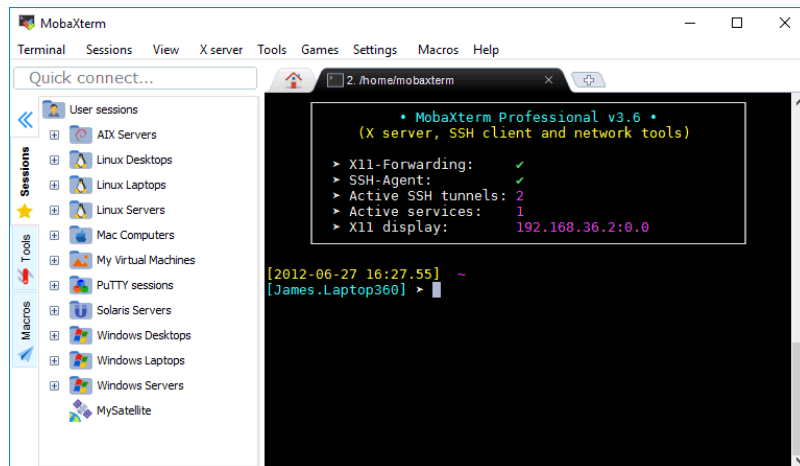
What if I don't want to install a new operating system?

There are options to access and use a command line even without having to install a new operating system in your computer. Let's see some of them.

Windows

MS Windows doesn't have a terminal emulator (except the latest versions), but there are a few options available.

- **Putty** (<http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>) is the simplest and most lightweight option to connect to a remote Unix server.
- **Putty Manager** (<http://puttymanager.sourceforge.net/>) extends putty and provides multiple tabs and options.
- **Cygwin** (<https://www.cygwin.com/>) emulates a whole unix environment. You can execute commands on your computer, or connect to a remote server
- **MobaXTerm** (<http://mobaxterm.mobatek.net/>) is a commercial solution (but the free version is usually enough) providing a Unix environment and some more options.



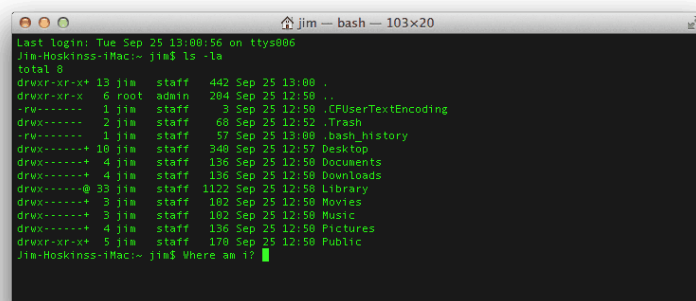
(<http://mobaxterm.mobatek.net/>)

To install new software in Cygwin or MobaXterm:, use (e.g. make):

```
apt-cyg install make
```

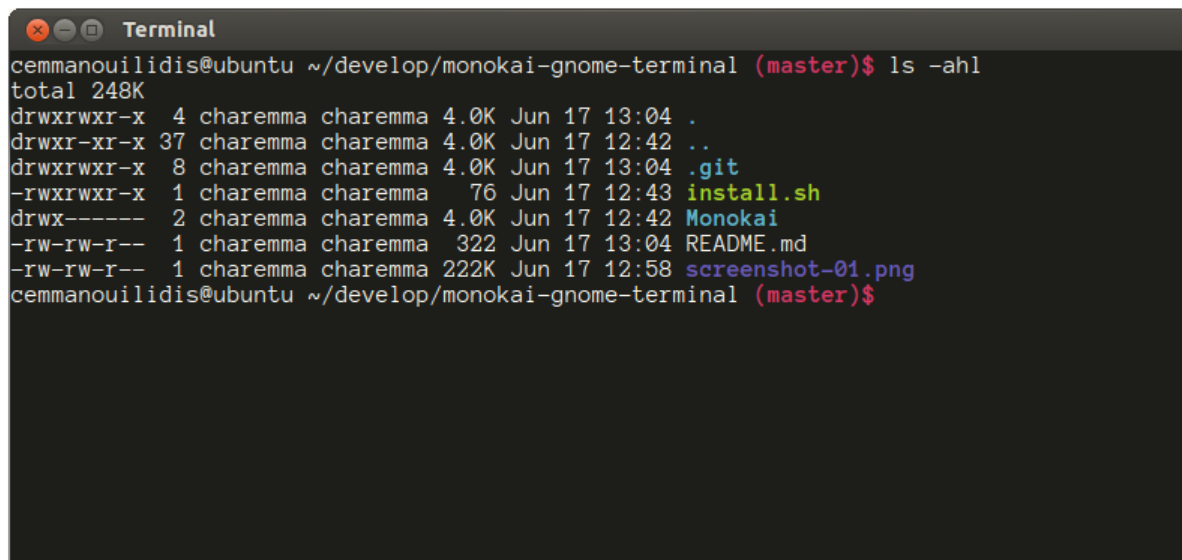
Mac

MacOs is also a Unix system, and it comes with a teminal emulator already installed. You should be able to use the Console App in Mac.



Linux

Congratulations on having Linux installed! There are several applications that you can use, from gnome-terminal to konsole.

A terminal window titled "Terminal" with a dark background and light-colored text. The prompt is "cemmanouilidis@ubuntu ~/develop/monokai-gnome-terminal (master)\$". The command "ls -ahl" has been executed, showing the following output:

```
total 248K
drwxrwxr-x  4 charemma charemma 4.0K Jun 17 13:04 .
drwxr-xr-x 37 charemma charemma 4.0K Jun 17 12:42 ..
drwxrwxr-x  8 charemma charemma 4.0K Jun 17 13:04 .git
-rwxrwxr-x  1 charemma charemma  76 Jun 17 12:43 install.sh
drwx----- 2 charemma charemma 4.0K Jun 17 12:42 Monokai
-rw-rw-r--  1 charemma charemma  32 Jun 17 13:04 README.md
-rw-rw-r--  1 charemma charemma 222K Jun 17 12:58 screenshot-01.png
```

The prompt is now "cemmanouilidis@ubuntu ~/develop/monokai-gnome-terminal (master)\$".

Time for a break!