Introdução Computação

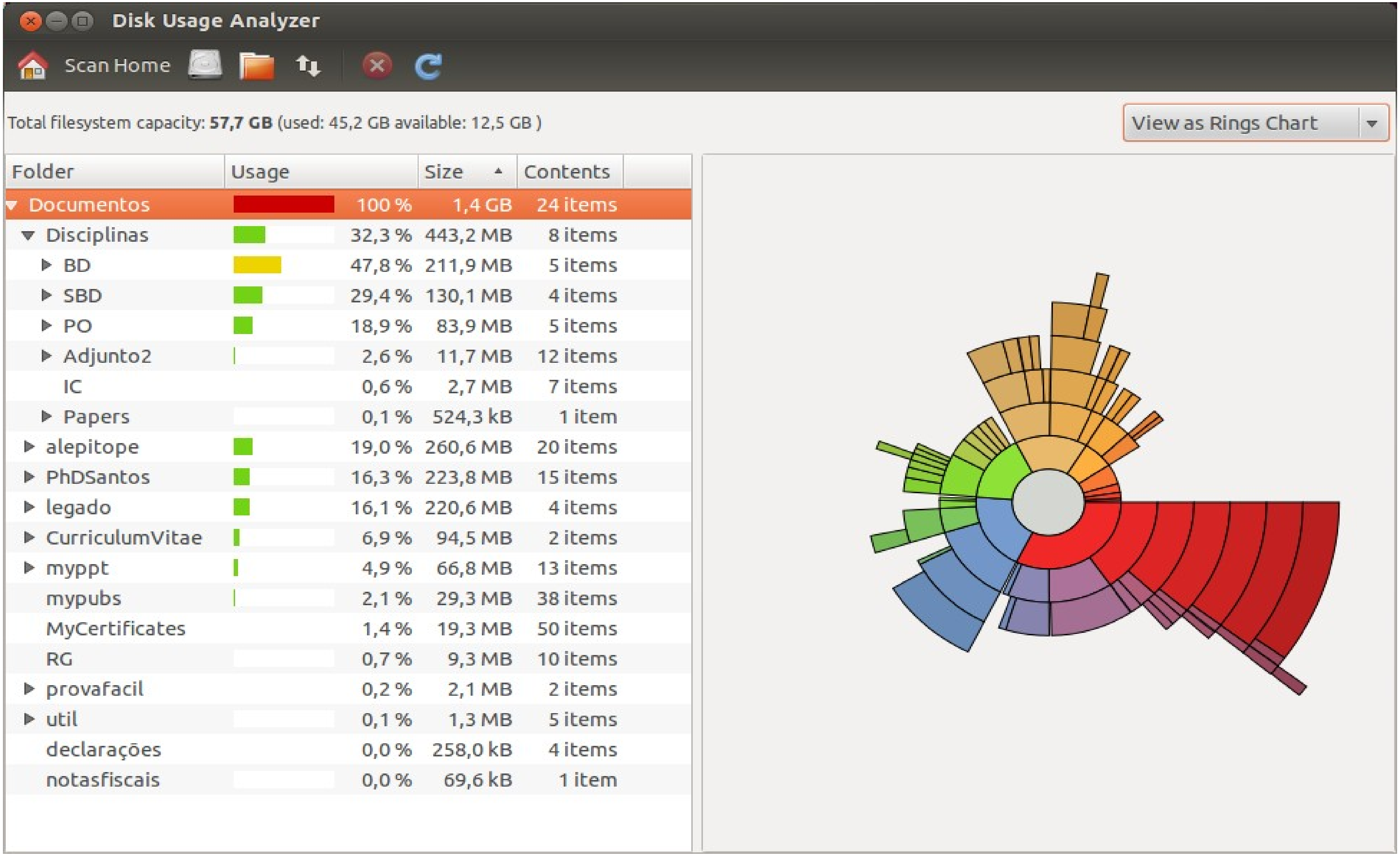
Comandos Linux

Professor Dr. Anderson Santos
santosardr@facom.ufu.br

Interface Gráfica

- Linux não é mais sinônimo de difícil;
- Interface baseadas na versão *Debian* são as mais “*user friendly*”;
- Grande variedade de software gratuito para fins científicos:
  - R e scilab competem com o caríssimo MatLab;

Interface Gráfica



Total filesystem capacity: **57,7 GB** (used: 45,2 GB available: 12,5 GB)

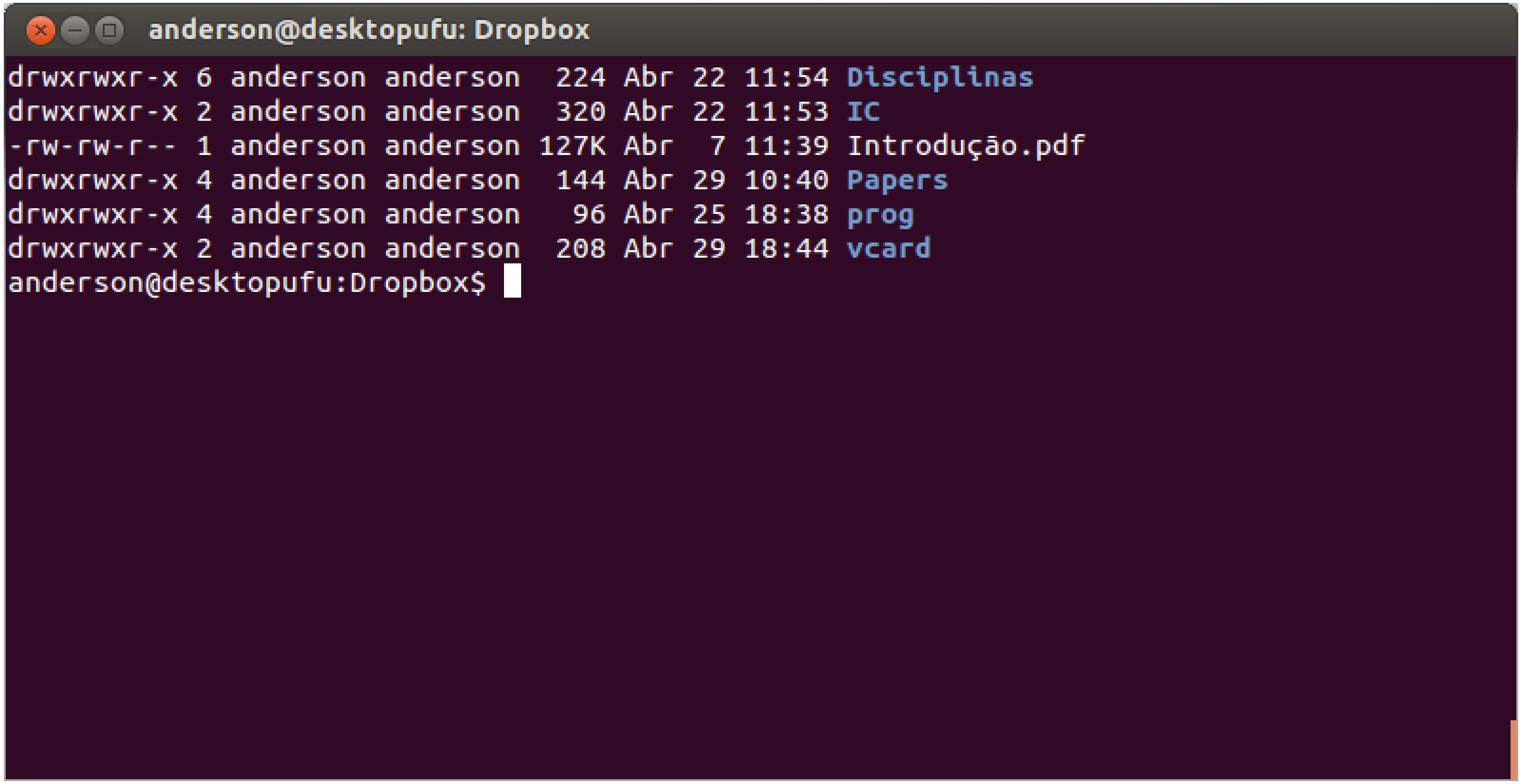
Folder	Usage	Size	Contents
▼ Documentos	100 %	1,4 GB	24 items
▼ Disciplinas	32,3 %	443,2 MB	8 items
▶ BD	47,8 %	211,9 MB	5 items
▶ SBD	29,4 %	130,1 MB	4 items
▶ PO	18,9 %	83,9 MB	5 items
▶ Adjunto2	2,6 %	11,7 MB	12 items
IC	0,6 %	2,7 MB	7 items
▶ Papers	0,1 %	524,3 kB	1 item
▶ alepitope	19,0 %	260,6 MB	20 items
▶ PhDSantos	16,3 %	223,8 MB	15 items
▶ legado	16,1 %	220,6 MB	4 items
▶ CurriculumVitae	6,9 %	94,5 MB	2 items
▶ myppt	4,9 %	66,8 MB	13 items
mypubs	2,1 %	29,3 MB	38 items
MyCertificates	1,4 %	19,3 MB	50 items
RG	0,7 %	9,3 MB	10 items
▶ provafacil	0,2 %	2,1 MB	2 items
▶ util	0,1 %	1,3 MB	5 items
declarações	0,0 %	258,0 kB	4 items
notasfiscais	0,0 %	69,6 kB	1 item

Interface Gráfica

- Entretanto quando é exigida eficiência no processamento uma interface gráfica não ajuda, pelo contrário: inviabiliza

O terminal ...

- … é o melhor amigo de quem lida com computação intensiva.



anderson@desktopufu: Dropbox

```
drwxrwxr-x 6 anderson anderson  224 Abr 22 11:54 Disciplinas
drwxrwxr-x 2 anderson anderson  320 Abr 22 11:53 IC
-rw-rw-r-- 1 anderson anderson 127K Abr  7 11:39 Introdução.pdf
drwxrwxr-x 4 anderson anderson  144 Abr 29 10:40 Papers
drwxrwxr-x 4 anderson anderson   96 Abr 25 18:38 prog
drwxrwxr-x 2 anderson anderson  208 Abr 29 18:44 vcard
anderson@desktopufu:Dropbox$
```

Comando *man*

```
MAN(1)                    Manual pager utils                    MAN(1)

NAME
     man - an interface to the on-line reference manuals

SYNOPSIS
     man  [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-m
     system[,...]] [-M path] [-S list] [-e extension] [-i|-I]  [--regex|--wildcard]
     [--names-only]  [-a]  [-u]  [--no-subpages]  [-P  pager]  [-r prompt] [-7] [-E
     encoding]   [--no-hyphenation]   [--no-justification]   [-p    string]    [-t]
     [-T[device]] [-H[browser]] [-X[dpi]] [-Z] [[section] page ...] ...
     man -k [apropos options] regexp ...
     man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
     man -f [whatis options] page ...
     man  -l  [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale]
     [-P pager] [-r prompt]  [-7]  [-E  encoding]  [-p  string]  [-t]  [-T[device]]
     [-H[browser]] [-X[dpi]] [-Z] file ...
     man -w|-W [-C file] [-d] [-D] page ...
     man -c [-C file] [-d] [-D] page ...
     man [-hV]
```

Comando *pwd*

```
PWD(1)                    User Commands                    PWD(1)

NAME
    pwd - print name of current/working directory

SYNOPSIS
    pwd [OPTION]...

DESCRIPTION
    Print the full filename of the current working directory.

    -L, --logical
        use PWD from environment, even if it contains symlinks

    -P, --physical
        avoid all symlinks

    --help display this help and exit

    --version
```

Comando *ls*

```
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 ..
```

Comando *cd*

- *cd = change directory*
- *cd ..*
- *cd ../*
- *cd ../../*
- *cd /*
- *cd ~*
- *cd sample*
- *cd sample/protein*

Comando *clear*

```
clear(1)                                            clear(1)

NAME
       clear - clear the terminal screen

SYNOPSIS
       clear

DESCRIPTION
       clear clears your screen if this is possible.  It looks in the environment for
       the terminal type and then in the terminfo database to figure out how to clear
       the screen.

       clear ignores any command-line parameters that may be present.

SEE ALSO
       tput(1), terminfo(5)

       This describes ncurses version 5.9 (patch 20110404).
```

Comando *alias*

- Permite criar outros comandos por meio de uma parametrização específica de um comando ou o acionamento de vários comandos
- alias lt='ls -lt'
- alias tudo='ls -al'
- alias cls='clear;ls -lt'

Comando *ping*

```
PING(8)          System Manager's Manual: iputils          PING(8)

NAME
    ping, ping6 - send ICMP ECHO_REQUEST to network hosts

SYNOPSIS
    ping  [-LRUbdfnqrvVaAB] [-c count] [-m mark] [-i interval] [-l preload]
    [-p pattern] [-s packetsize] [-t ttl] [-w deadline] [-F flowlabel]  [-I
    interface]  [-M  hint] [-N nioption] [-Q tos] [-S sndbuf] [-T timestamp
    option] [-W timeout] [hop ...] destination

DESCRIPTION
    ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit
    an  ICMP  ECHO_RESPONSE from a host or gateway.  ECHO_REQUEST datagrams
    (``pings'') have an IP and ICMP header, followed by  a  struct  timeval
    and  then  an  arbitrary  number  of ``pad'' bytes used to fill out the
    packet.

    ping6 can also send Node Information Queries (RFC4620).

OPTIONS
    -a     Audible ping.
```

Comando *wget*

```
WGET(1)                    GNU Wget                    WGET(1)

NAME
    Wget - The non-interactive network downloader.

SYNOPSIS
    wget [option]... [URL]...

DESCRIPTION
    GNU Wget is a free utility for non-interactive download of files from
    the Web.  It supports HTTP, HTTPS, and FTP protocols, as well as
    retrieval through HTTP proxies.

    Wget is non-interactive, meaning that it can work in the background,
    while the user is not logged on.  This allows you to start a retrieval
    and disconnect from the system, letting Wget finish the work.  By
    contrast, most of the Web browsers require constant user's presence,
    which can be a great hindrance when transferring a lot of data.

    Wget can follow links in HTML, XHTML, and CSS pages, to create local
    versions of remote web sites, fully recreating the directory structure
    of the original site.  This is sometimes referred to as "recursive
    downloading."  While doing that, Wget respects the Robot Exclusion
```

Comando *tar*

```
TAR(1)              BSD General Commands Manual             TAR(1)

NAME
   tar — The GNU version of the tar archiving utility

SYNOPSIS
   tar [-] A --catenate --concatenate | c --create | d --diff --compare |
      --delete | r --append | t --list | --test-label | u --update | x
      --extract --get [options] [pathname ...]

DESCRIPTION
   Tar stores and extracts files from a tape or disk archive.

   The first argument to tar should be a function; either one of the letters
   Acdrtux, or one of the long function names.  A function letter need not
   be prefixed with ``-'', and may be combined with other single-letter
   options.  A long function name must be prefixed with --.  Some options
   take a parameter; with the single-letter form these must be given as sep-
   arate arguments.  With the long form, they may be given by appending
   =value to the option.
```

Comando *cat*

```
CAT(1)                    User Commands                    CAT(1)

NAME
    cat - concatenate files and print on the standard output

SYNOPSIS
    cat [OPTION]... [FILE]...

DESCRIPTION
    Concatenate FILE(s), or standard input, to standard output.

    -A, --show-all
        equivalent to -vET

    -b, --number-nonblank
        number nonempty output lines, overrides -n

    -e     equivalent to -vE

    -E, --show-ends
        display $ at end of each line

    -n, --number
```

Comandos *more* ou *less*

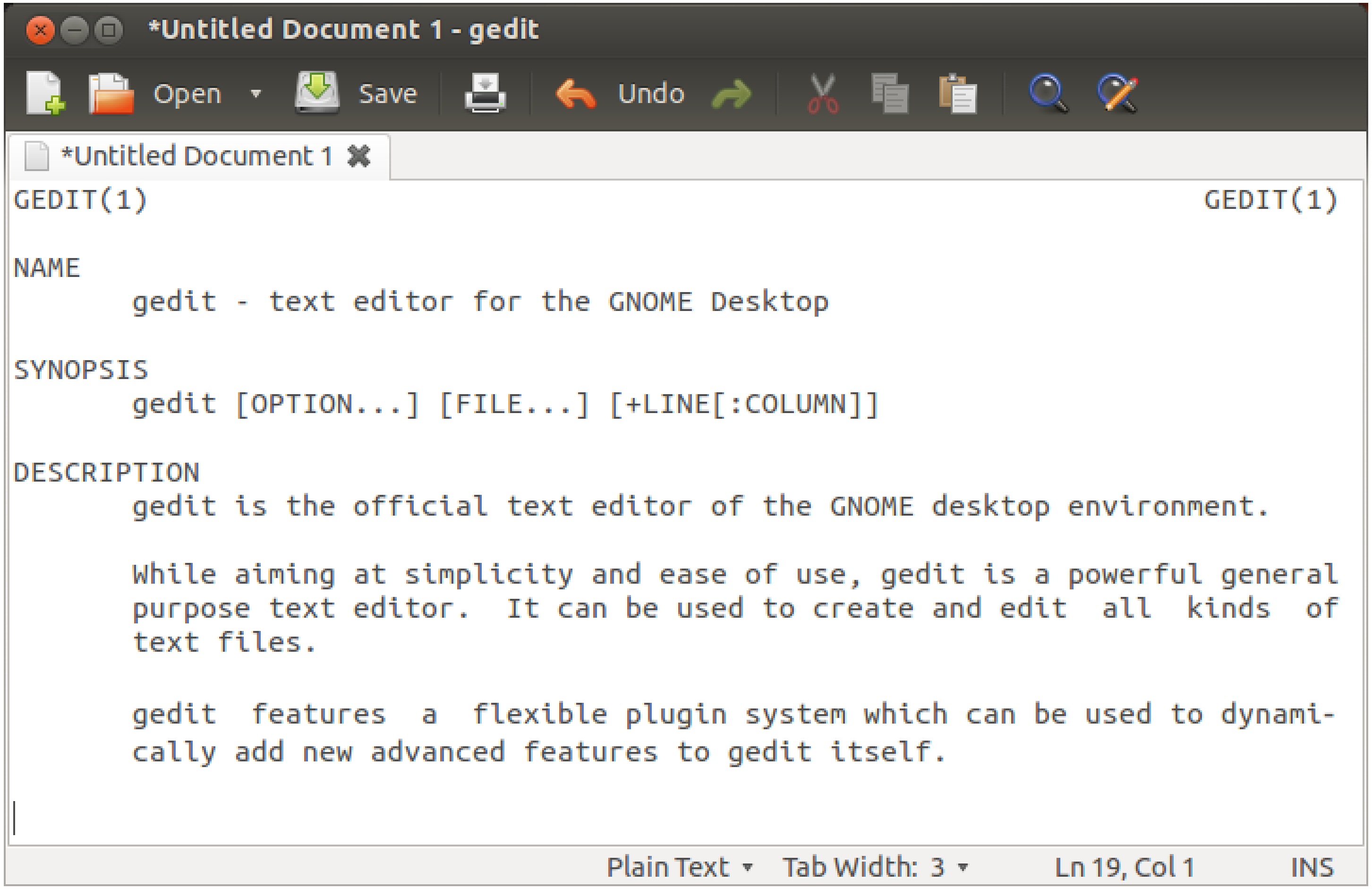
```
LESS(1)                                   LESS(1)

NAME
    less - opposite of more

SYNOPSIS
    less -?
    less --help
    less -V
    less --version
    less [-[+]aABcCdeEfFgGiIJKLmMnNqQrRsSuUVwWX~]
       [-b space] [-h lines] [-j line] [-k keyfile]
       [-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
       [-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
       [-# shift] [+[+]cmd] [--] [filename]...
    (See  the  OPTIONS section for alternate option syntax with long option
    names.)

DESCRIPTION
    Less is a program similar to more (1), but which allows backward  move-
    ment in the file as well as forward movement.  Also, less does not have
    to read the entire input file before  starting,  so  with  large  input
    files  it  starts  up  faster than text editors like vi (1).
```

Comando *gedit*



```
GEDIT(1)                                                            GEDIT(1)

NAME
       gedit - text editor for the GNOME Desktop

SYNOPSIS
       gedit [OPTION...] [FILE...] [+LINE[:COLUMN]]

DESCRIPTION
       gedit is the official text editor of the GNOME desktop environment.

       While aiming at simplicity and ease of use, gedit is a powerful general
       purpose text editor.  It can be used to create and edit  all  kinds  of
       text files.

       gedit  features  a  flexible plugin system which can be used to dynami-
       cally add new advanced features to gedit itself.
```

Plain Text ▾ Tab Width: 3 ▾ Ln 19, Col 1 INS

Comandos *head* e *tail*

```
HEAD(1)                    User Commands                   HEAD(1)

NAME
       head - output the first part of files

SYNOPSIS
       head [OPTION]... [FILE]...

DESCRIPTION
       Print  the  first  10 lines of each FILE to standard output.  With more
       than one FILE, precede each with a header giving the file  name.   With
       no FILE, or when FILE is -, read standard input.

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

       -c, --bytes=[-]K
              print the first K bytes of each  file;  with  the  leading  `-',
              print all but the last K bytes of each file

       -n, --lines=[-]K
              print  the first K lines instead of the first 10; with the lead-
              ing `-', print all but the last K lines of each file
```

Comando *cp*

```
CP(1)                 User Commands                CP(1)

NAME
    cp - copy files and directories

SYNOPSIS
    cp [OPTION]... [-T] SOURCE DEST
    cp [OPTION]... SOURCE... DIRECTORY
    cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
    Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

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

    -a, --archive
        same as -dR --preserve=all

    --attributes-only
        don't copy the file data, just the attributes
```

Comando *mv*

```
MV(1)                    User Commands                    MV(1)

NAME
    mv - move (rename) files

SYNOPSIS
    mv [OPTION]... [-T] SOURCE DEST
    mv [OPTION]... SOURCE... DIRECTORY
    mv [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
    Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

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

    --backup[=CONTROL]
        make a backup of each existing destination file

    -b    like --backup but does not accept an argument

    -f, --force
        do not prompt before overwriting
```

Comando *rm*

```
RM(1)                    User Commands                    RM(1)

NAME
    rm - remove files or directories

SYNOPSIS
    rm [OPTION]... FILE...

DESCRIPTION
    This  manual  page  documents  the  GNU version of rm.  rm removes each
    specified file.  By default, it does not remove directories.

    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 whether to proceed with the entire operation.   If
    the response is not affirmative, the entire command is aborted.

    Otherwise,  if  a file is unwritable, standard input is a terminal, and
    the -f or --force  option  is  not  given,  or  the  -i  or  --interac-
    tive=always  option is given, rm prompts the user for whether to remove
    the file.  If the response is not affirmative, the file is skipped.
```

Comandos *mkdir* e *rmdir*

```
RMDIR(1)                User Commands                RMDIR(1)

NAME
    rmdir - remove empty directories

SYNOPSIS
    rmdir [OPTION]... DIRECTORY...

DESCRIPTION
    Remove the DIRECTORY(ies), if they are empty.

    --ignore-fail-on-non-empty

        ignore each failure that is solely because a directory

        is non-empty

    -p, --parents
        remove  DIRECTORY  and  its ancestors; e.g., `rmdir -p a/b/c' is
        similar to `rmdir a/b/c a/b a'

    -v, --verbose
        output a diagnostic for every directory processed
```

Comando *which*

```
WHICH(1)                                    WHICH(1)

NAME
    which - locate a command

SYNOPSIS
    which [-a] filename ...

DESCRIPTION
    which returns the pathnames of the files (or links) which would be exe-
    cuted in the current environment, had its arguments been given as  com-
    mands  in a strictly POSIX-conformant shell.  It does this by searching
    the PATH for executable files matching the names of the  arguments.  It
    does not follow symbolic links.

OPTIONS
    -a     print all matching pathnames of each argument

EXIT STATUS
    0      if all specified commands are found and executable

    1      if  one  or  more  specified commands is nonexistent or not exe-
           cutable
```

Comando *history*

```
HISTORY(3)                                      HISTORY(3)

NAME
    history - GNU History Library

COPYRIGHT
    The GNU History Library is Copyright (C) 1989-2011 by the Free Software
    Foundation, Inc.

DESCRIPTION
    Many programs read input from the user a line at a time.  The GNU  His-
    tory  library is able to keep track of those lines, associate arbitrary
    data with each line, and utilize information  from  previous  lines  in
    composing new ones.

HISTORY EXPANSION
    The  history library supports a history expansion feature that is iden-
    tical to the history expansion in bash.  This  section  describes  what
    syntax features are available.

    History expansions introduce words from the history list into the input
    stream, making it easy to repeat commands, insert the  arguments  to  a
    previous command into the current input line, or fix errors in previous
```

Comando *exit*

```
EXIT(3)                    Linux Programmer's Manual                    EXIT(3)

NAME
       exit - cause normal process termination

SYNOPSIS
       #include <stdlib.h>

       void exit(int status);

DESCRIPTION
       The  exit() function causes normal process termination and the value of
       status & 0377 is returned to the parent (see wait(2)).

       All functions registered with atexit(3) and on_exit(3) are  called,  in
       the  reverse  order  of their registration.  (It is possible for one of
       these functions to use atexit(3) or on_exit(3)  to  register  an  addi-
       tional  function  to be executed during exit processing; the new regis-
       tration is added to the front of the list of functions that  remain  to
       be  called.)  If one of these functions does not return (e.g., it calls
       _exit(2), or kills itself with a signal), then none  of  the  remaining
       functions is called, and further exit processing (in particular, flush-
       ing of stdio(3) streams) is abandoned.
```