UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA

Licenciatura en Sistemas Computacionales

Taller de Linux

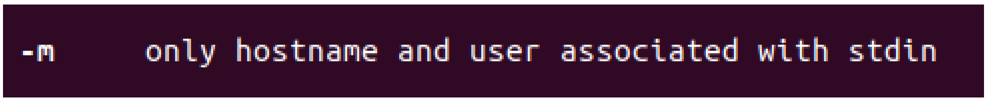
Practica #3

Grupo 103

Alumno: *Moreno Avendaño Erandi Sacbe*
Matrícula: *1190721*
Maestra: *Julia Corrales Espinoza*

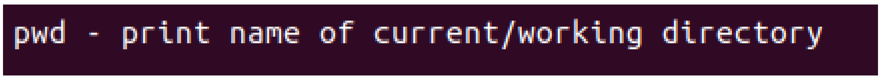
17 de agosto del 2022

1. Buscar la ayuda del mando who –m y who i.



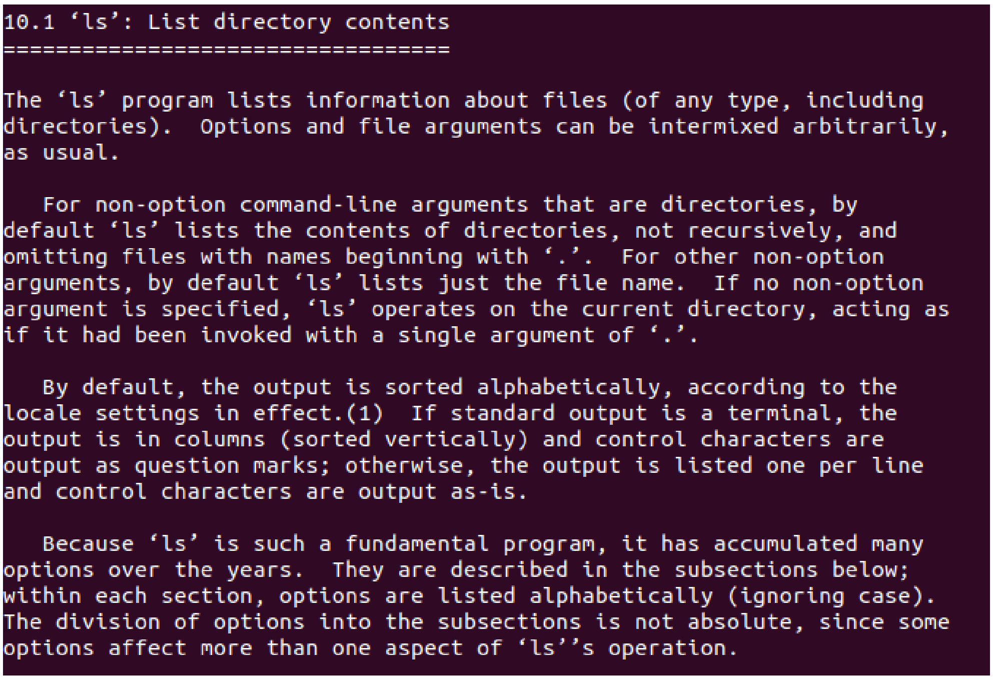
```
-m     only hostname and user associated with stdin
```

2. Cuál es la funcionalidad del mando pwd

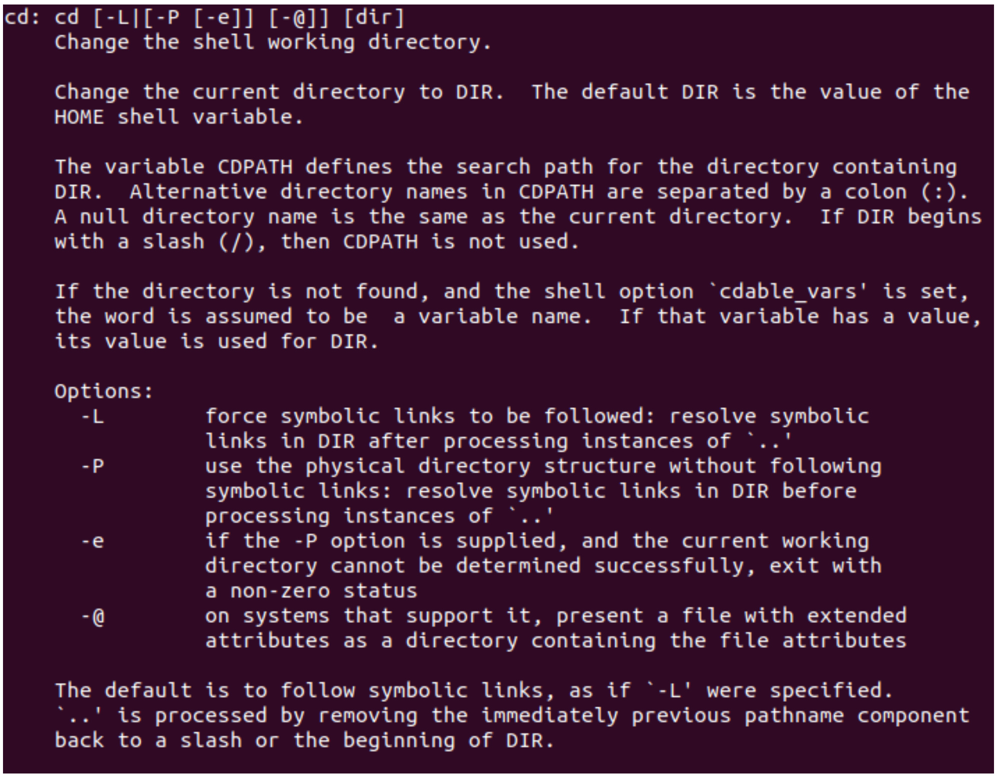


```
pwd - print name of current/working directory
```

3. Obtener la información del mando cd y ls y explicar si es adecuado utilizar los dos mandos uno seguido del otro.

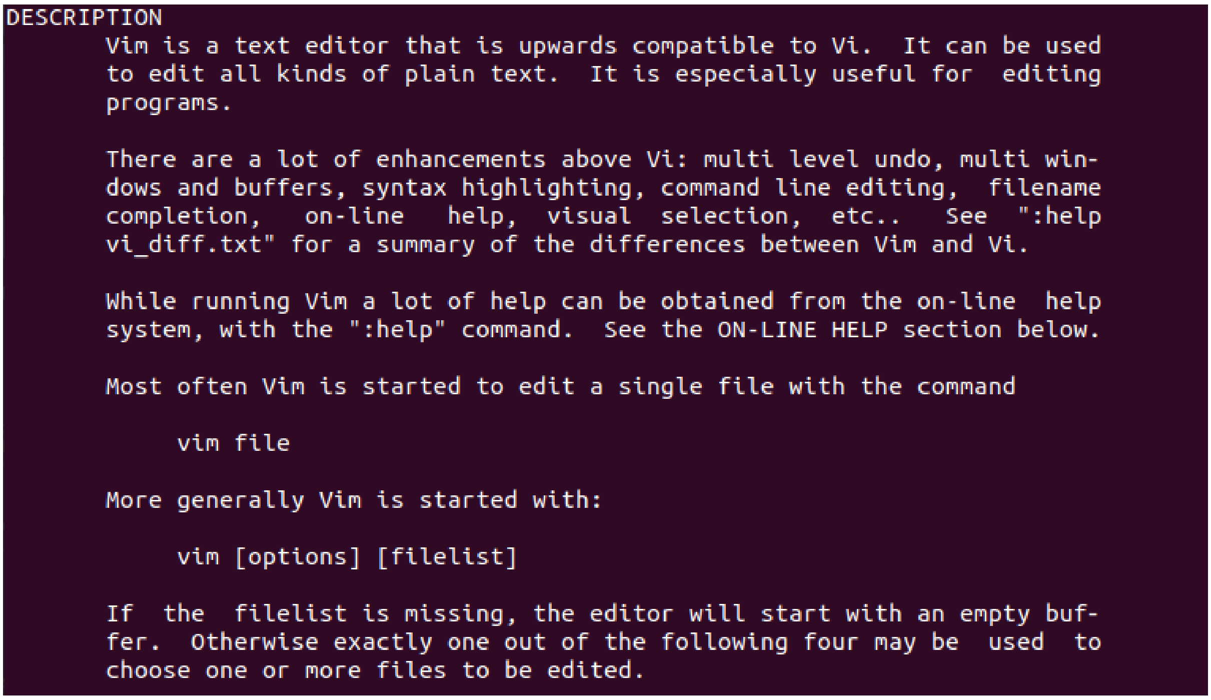


```
10.1 ‘ls’: List directory contents
==================================

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
and control characters are output as-is.

   Because ‘ls’ is such a fundamental program, it has accumulated many
options over the years.  They are described in the subsections below;
within each section, options are listed alphabetically (ignoring case).
The division of options into the subsections is not absolute, since some
options affect more than one aspect of ‘ls’’s operation.
```

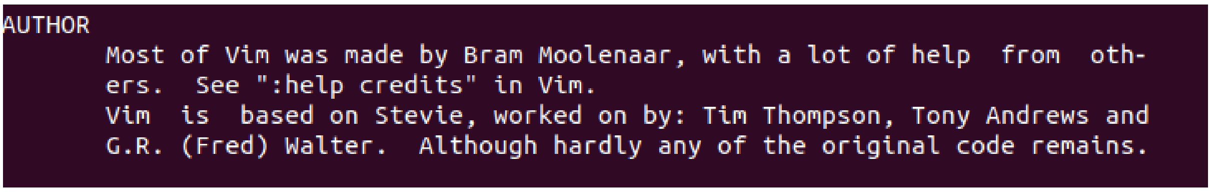


```
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default DIR is the value of the
    HOME shell variable.

    The variable CDPATH defines the search path for the directory containing
    DIR.  Alternative directory names in CDPATH are separated by a colon (:).
    A null directory name is the same as the current directory.  If DIR begins
    with a slash (/), then CDPATH is not used.

    If the directory is not found, and the shell option `cdable_vars' is set,
    the word is assumed to be  a variable name.  If that variable has a value,
    its value is used for DIR.

    Options:
      -L        force symbolic links to be followed: resolve symbolic
                links in DIR after processing instances of `..'
      -P        use the physical directory structure without following
                symbolic links: resolve symbolic links in DIR before
                processing instances of `..'
      -e        if the -P option is supplied, and the current working
                directory cannot be determined successfully, exit with
                a non-zero status
      -@        on systems that support it, present a file with extended
                attributes as a directory containing the file attributes

    The default is to follow symbolic links, as if `-L' were specified.
    `..' is processed by removing the immediately previous pathname component
    back to a slash or the beginning of DIR.
```

4. Obtener la ayuda del mando vim y explique su utilidad.



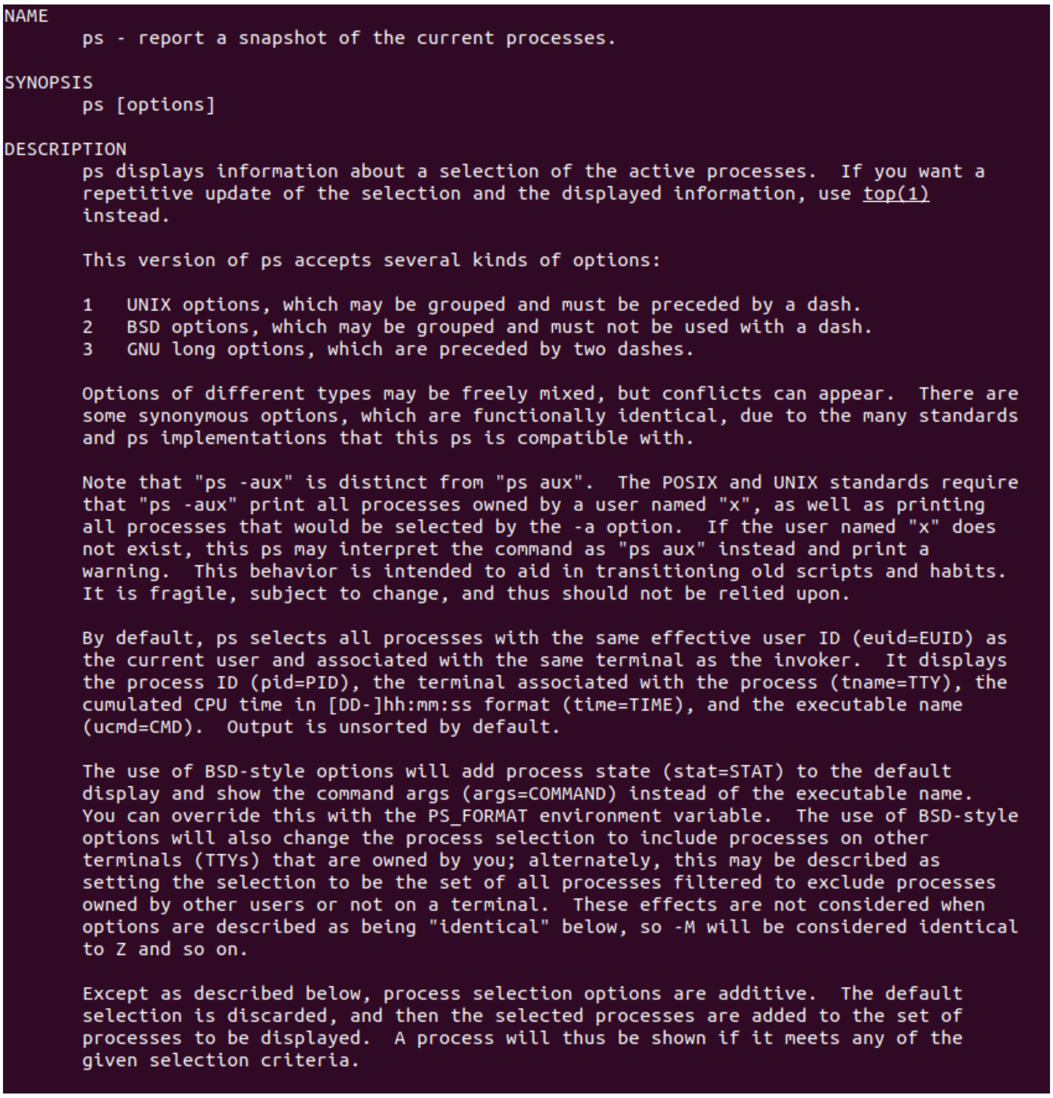
```
DESCRIPTION
       Vim is a text editor that is upwards compatible to Vi.  It can be used
       to edit all kinds of plain text.  It is especially useful for  editing
       programs.

       There are a lot of enhancements above Vi: multi level undo, multi win-
       dows and buffers, syntax highlighting, command line editing,  filename
       completion,   on-line   help,  visual  selection,  etc..   See  ":help
       vi_diff.txt" for a summary of the differences between Vim and Vi.

       While running Vim a lot of help can be obtained from the on-line  help
       system, with the ":help" command.  See the ON-LINE HELP section below.

       Most often Vim is started to edit a single file with the command

            vim file

       More generally Vim is started with:

            vim [options] [filelist]

       If  the  filelist is missing, the editor will start with an empty buf-
       fer.  Otherwise exactly one out of the following four may be  used  to
       choose one or more files to be edited.
```



```
AUTHOR
       Most of Vim was made by Bram Moolenaar, with a lot of help  from  oth-
       ers.  See ":help credits" in Vim.
       Vim  is  based on Stevie, worked on by: Tim Thompson, Tony Andrews and
       G.R. (Fred) Walter.  Although hardly any of the original code remains.
```

Sirve como un editor de texto con atajos de teclado específicos para cada función, lo que hace la edición de texto muy rápida sin siquiera utilizar el mouse.

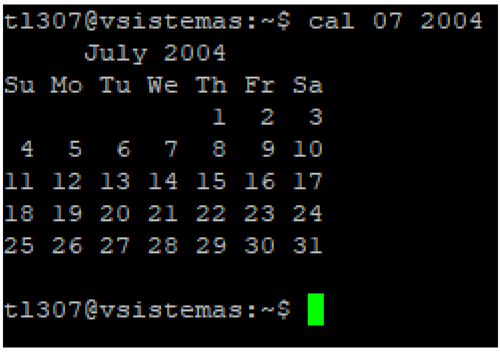
5. Obtener la ayuda del mando ps cual es su funcionalidad, ¿Cómo puede finalizar un proceso?



```
NAME
       ps - report a snapshot of the current processes.

SYNOPSIS
       ps [options]

DESCRIPTION
       ps displays information about a selection of the active processes.  If you want a
       repetitive update of the selection and the displayed information, use top(1)
       instead.

       This version of ps accepts several kinds of options:

       1   UNIX options, which may be grouped and must be preceded by a dash.
       2   BSD options, which may be grouped and must not be used with a dash.
       3   GNU long options, which are preceded by two dashes.

       Options of different types may be freely mixed, but conflicts can appear.  There are
       some synonymous options, which are functionally identical, due to the many standards
       and ps implementations that this ps is compatible with.

       Note that "ps -aux" is distinct from "ps aux".  The POSIX and UNIX standards require
       that "ps -aux" print all processes owned by a user named "x", as well as printing
       all processes that would be selected by the -a option.  If the user named "x" does
       not exist, this ps may interpret the command as "ps aux" instead and print a
       warning.  This behavior is intended to aid in transitioning old scripts and habits.
       It is fragile, subject to change, and thus should not be relied upon.

       By default, ps selects all processes with the same effective user ID (euid=EUID) as
       the current user and associated with the same terminal as the invoker.  It displays
       the process ID (pid=PID), the terminal associated with the process (tname=TTY), the
       cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and the executable name
       (ucmd=CMD).  Output is unsorted by default.

       The use of BSD-style options will add process state (stat=STAT) to the default
       display and show the command args (args=COMMAND) instead of the executable name.
       You can override this with the PS_FORMAT environment variable.  The use of BSD-style
       options will also change the process selection to include processes on other
       terminals (TTYs) that are owned by you; alternately, this may be described as
       setting the selection to be the set of all processes filtered to exclude processes
       owned by other users or not on a terminal.  These effects are not considered when
       options are described as being "identical" below, so -M will be considered identical
       to Z and so on.

       Except as described below, process selection options are additive.  The default
       selection is discarded, and then the selected processes are added to the set of
       processes to be displayed.  A process will thus be shown if it meets any of the
       given selection criteria.
```

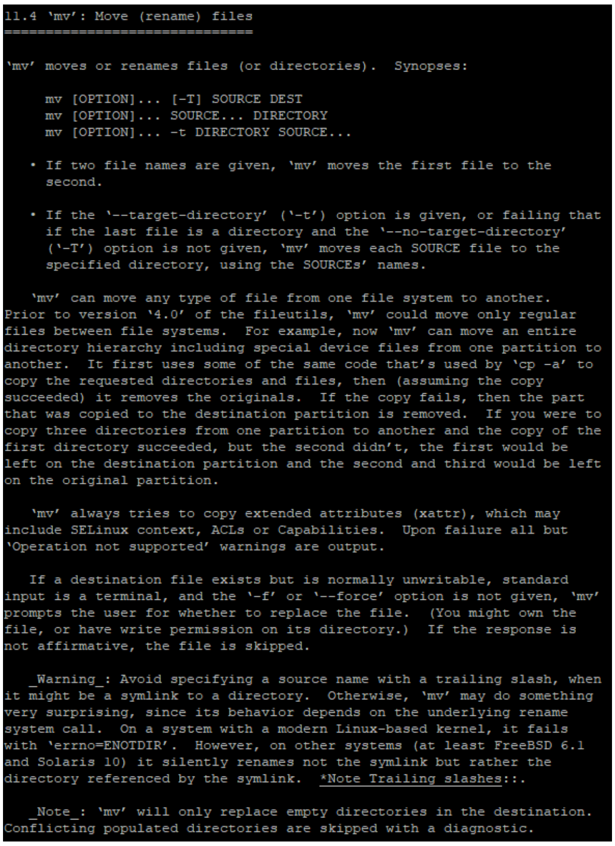
Con el comando ps puedes ver los procesos activos de los usuarios del sistema, con el se puede especificar con ciertos parámetros que muestre el ID de los procesos activos y con el comando kill se puede finalizar un proceso dada su ID.

6. Utilice el mando cal para buscar día de su cumpleaños
Yo nací el 23 de Julio del 2004



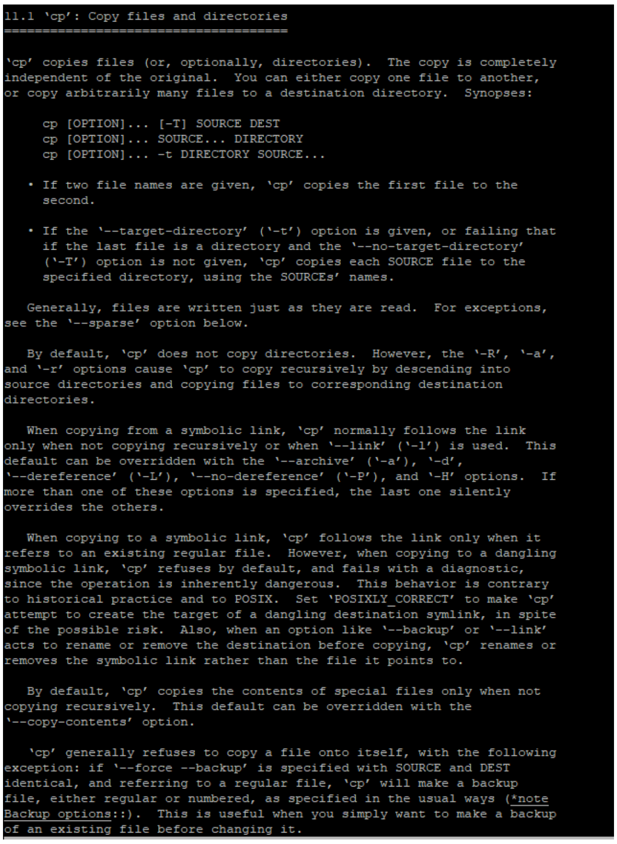
```
t1307@vsistemas:~$ cal 07 2004
     July 2004
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

t1307@vsistemas:~$
```

7. Buscar el mando mv y cp y ver la diferencia entre cada uno de ellos



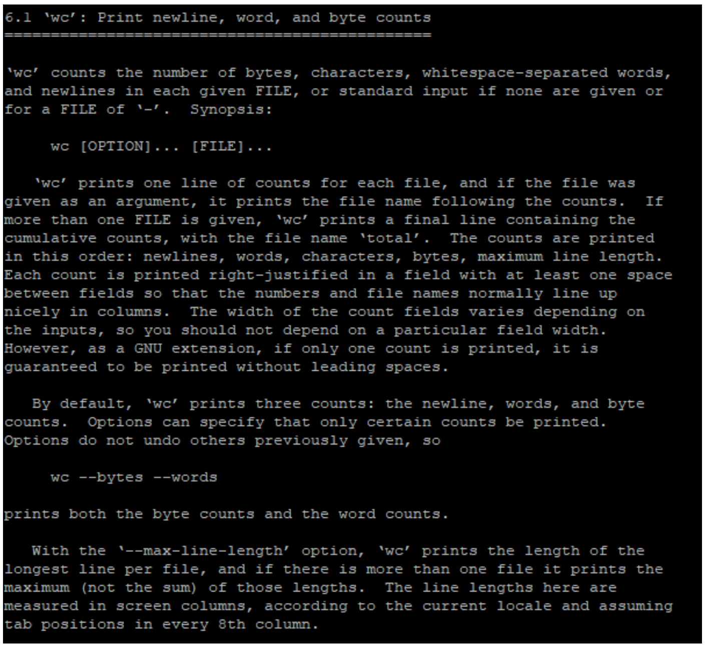
```
11.4 'mv': Move (rename) files
==============================

'mv' moves or renames files (or directories).  Synopses:

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

   • If two file names are given, 'mv' moves the first file to the
     second.

   • If the '--target-directory' ('-t') option is given, or failing that
     if the last file is a directory and the '--no-target-directory'
     ('-T') option is not given, 'mv' moves each SOURCE file to the
     specified directory, using the SOURCEs' names.

   'mv' can move any type of file from one file system to another.
Prior to version '4.0' of the fileutils, 'mv' could move only regular
files between file systems.  For example, now 'mv' can move an entire
directory hierarchy including special device files from one partition to
another.  It first uses some of the same code that's used by 'cp -a' to
copy the requested directories and files, then (assuming the copy
succeeded) it removes the originals.  If the copy fails, then the part
that was copied to the destination partition is removed.  If you were to
copy three directories from one partition to another and the copy of the
first directory succeeded, but the second didn't, the first would be
left on the destination partition and the second and third would be left
on the original partition.

   'mv' always tries to copy extended attributes (xattr), which may
include SELinux context, ACLs or Capabilities.  Upon failure all but
'Operation not supported' warnings are output.

   If a destination file exists but is normally unwritable, standard
input is a terminal, and the '-f' or '--force' option is not given, 'mv'
prompts the user for whether to replace the file.  (You might own the
file, or have write permission on its directory.)  If the response is
not affirmative, the file is skipped.

   _Warning_: Avoid specifying a source name with a trailing slash, when
it might be a symlink to a directory.  Otherwise, 'mv' may do something
very surprising, since its behavior depends on the underlying rename
system call.  On a system with a modern Linux-based kernel, it fails
with 'errno=ENOTDIR'.  However, on other systems (at least FreeBSD 6.1
and Solaris 10) it silently renames not the symlink but rather the
directory referenced by the symlink.  *Note Trailing slashes::.

   _Note_: 'mv' will only replace empty directories in the destination.
Conflicting populated directories are skipped with a diagnostic.
```

El comando mv se utiliza para mover archivos a un directorio especificado
Mientras que cp se utiliza para copiar archivos en un directorio especificado



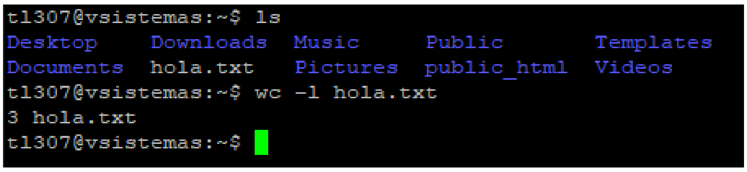
```
11.1 'cp': Copy files and directories
=====================================

'cp' copies files (or, optionally, directories).  The copy is completely
independent of the original.  You can either copy one file to another,
or copy arbitrarily many files to a destination directory.  Synopses:

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

   • If two file names are given, 'cp' copies the first file to the
     second.

   • If the '--target-directory' ('-t') option is given, or failing that
     if the last file is a directory and the '--no-target-directory'
     ('-T') option is not given, 'cp' copies each SOURCE file to the
     specified directory, using the SOURCEs' names.

   Generally, files are written just as they are read.  For exceptions,
see the '--sparse' option below.

   By default, 'cp' does not copy directories.  However, the '-R', '-a',
and '-r' options cause 'cp' to copy recursively by descending into
source directories and copying files to corresponding destination
directories.

   When copying from a symbolic link, 'cp' normally follows the link
only when not copying recursively or when '--link' ('-l') is used.  This
default can be overridden with the '--archive' ('-a'), '-d',
'--dereference' ('-L'), '--no-dereference' ('-P'), and '-H' options.  If
more than one of these options is specified, the last one silently
overrides the others.

   When copying to a symbolic link, 'cp' follows the link only when it
refers to an existing regular file.  However, when copying to a dangling
symbolic link, 'cp' refuses by default, and fails with a diagnostic,
since the operation is inherently dangerous.  This behavior is contrary
to historical practice and to POSIX.  Set 'POSIXLY_CORRECT' to make 'cp'
attempt to create the target of a dangling destination symlink, in spite
of the possible risk.  Also, when an option like '--backup' or '--link'
acts to rename or remove the destination before copying, 'cp' renames or
removes the symbolic link rather than the file it points to.

   By default, 'cp' copies the contents of special files only when not
copying recursively.  This default can be overridden with the
'--copy-contents' option.

   'cp' generally refuses to copy a file onto itself, with the following
exception: if '--force --backup' is specified with SOURCE and DEST
identical, and referring to a regular file, 'cp' will make a backup
file, either regular or numbered, as specified in the usual ways (*note
Backup options::).  This is useful when you simply want to make a backup
of an existing file before changing it.
```

8. Buscar el mando wc y visualizar cómo funciona el número de línea y para que nos es conveniente utilizarlo.



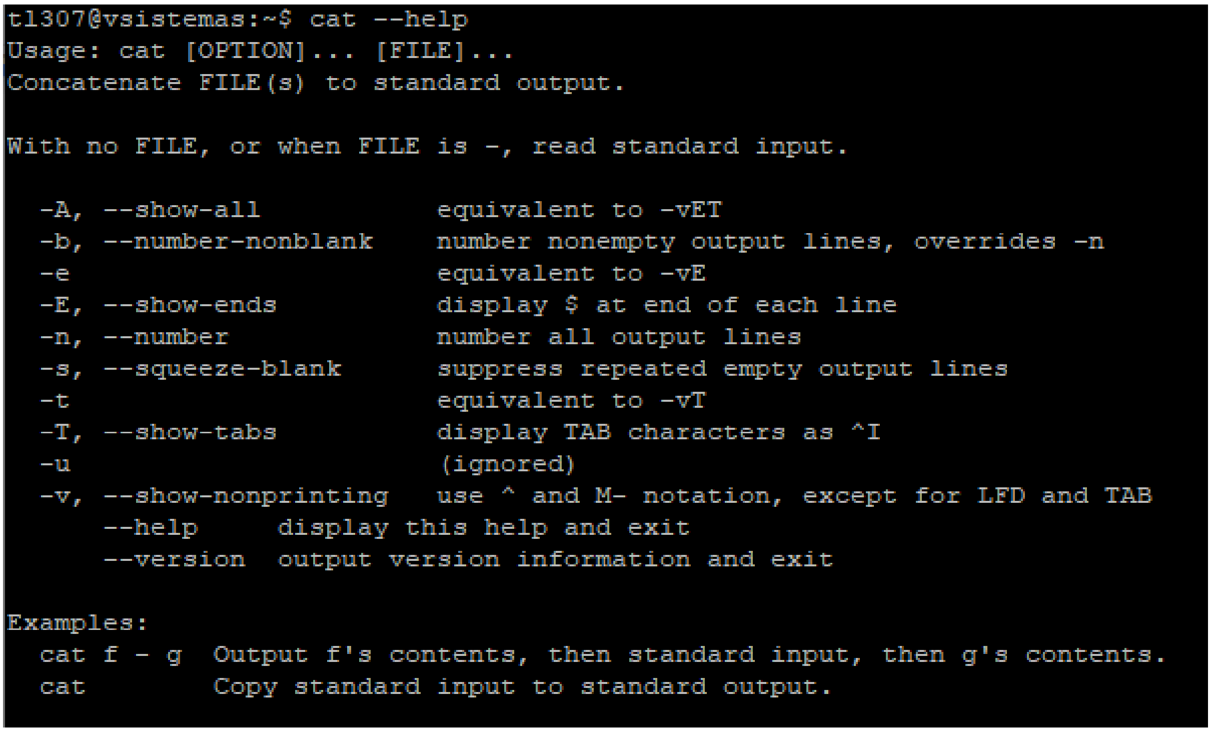
```
6.1 'wc': Print newline, word, and byte counts
==============================================

'wc' counts the number of bytes, characters, whitespace-separated words,
and newlines in each given FILE, or standard input if none are given or
for a FILE of '-'.  Synopsis:

     wc [OPTION]... [FILE]...

   'wc' prints one line of counts for each file, and if the file was
given as an argument, it prints the file name following the counts.  If
more than one FILE is given, 'wc' prints a final line containing the
cumulative counts, with the file name 'total'.  The counts are printed
in this order: newlines, words, characters, bytes, maximum line length.
Each count is printed right-justified in a field with at least one space
between fields so that the numbers and file names normally line up
nicely in columns.  The width of the count fields varies depending on
the inputs, so you should not depend on a particular field width.
However, as a GNU extension, if only one count is printed, it is
guaranteed to be printed without leading spaces.

   By default, 'wc' prints three counts: the newline, words, and byte
counts.  Options can specify that only certain counts be printed.
Options do not undo others previously given, so

     wc --bytes --words

prints both the byte counts and the word counts.

   With the '--max-line-length' option, 'wc' prints the length of the
longest line per file, and if there is more than one file it prints the
maximum (not the sum) of those lengths.  The line lengths here are
measured in screen columns, according to the current locale and assuming
tab positions in every 8th column.
```

Para contar el número de líneas se utiliza



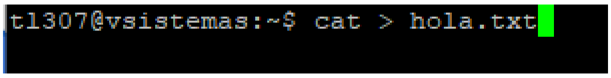
```
tl307@vsistemas:~$ ls
Desktop    Downloads  Music     Public       Templates
Documents  hola.txt   Pictures  public_html  Videos
tl307@vsistemas:~$ wc -l hola.txt
3 hola.txt
tl307@vsistemas:~$
```

9. Buscar la información del mando cat y escribir cómo puedo crear un archivo.



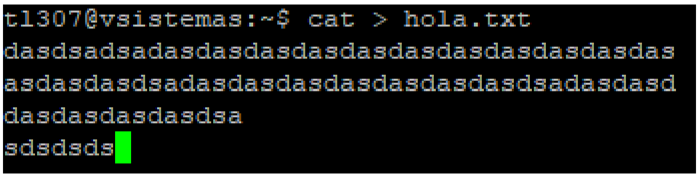
```
tl307@vsistemas:~$ cat --help
Usage: cat [OPTION]... [FILE]...
Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

  -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             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
  -u                       (ignored)
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
      --help     display this help and exit
      --version  output version information and exit

Examples:
  cat f - g  Output f's contents, then standard input, then g's contents.
  cat        Copy standard input to standard output.
```

Se puede crear un archivo utilizando cat y el operador de > y el nombre del archivo junto con su extensión, ejemplo:



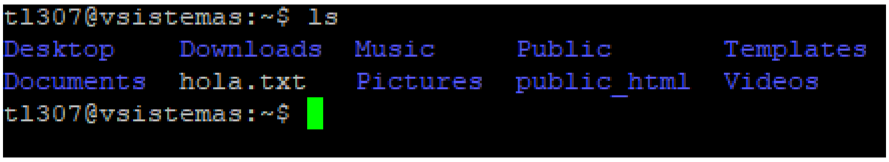
```
tl307@vsistemas:~$ cat > hola.txt
```

Después podrás añadir texto y para terminar presiona CTRL + D



```
tl307@vsistemas:~$ cat > hola.txt
dasdsadsadasdasdasdasdasdasdasdasdasdasdas
asdasdasdsadasdasdasdasdasdasdasdsadasdasd
dasdasdasdasdsa
sdsdsds
```

Ahí se puede ver el archivo creado



```
tl307@vsistemas:~$ ls
Desktop     Downloads  Music     Public       Templates
Documents   hola.txt   Pictures  public_html  Videos
tl307@vsistemas:~$
```