

Prácticas del Tema 1: Introducción al biocómputo en sistemas GNU/Linux

Pablo Vinuesa

2018-02-23

Contents

Tema 1. Introducción al biocómputo en sistemas GNU/Linux	2
Exploreemos el sistema en el que estamos trabajando	2
ssh establecer sesion remota encriptada (segura) via ssh al servidor ivory	2
hostname muestra el nombre del host (la máquina a la que estoy conectado) y la IP	2
uname muestra el sistema operativo del host	2
Variables de ambiente: \$HOME \$PATH \$USER	2
EXPLORACIÓN EL SISTEMA DE ARCHIVOS	3
pwd imprime la ruta absoluta del directorio actual	3
ls lista contenidos del directorio	3
Veamos el contenido del directorio raiz	3
Veamos el contenido del directorio /bin usando el comodín '*'	3
expansión de caracteres con * y ?	5
cd para cambiar de directorios usando rutas absolutas y relativas	6
de nuevo, donde estoy?	6
sube un directorio usando RUTA RELATIVA	6
donde estoy?	6
regresa a tu home	6
cd cambiar directorios con rutas absolutas (/ruta/completa/al/dir) y relativas ../../	6
Generación de directorios y permisos, copiado de archivos y directorios	6
mkdir genera nuevos directorios	6
ls -l lista los permisos de archivos y directorios	7
copiar, mover, renombrar y borrar archivos	7
generar un directorio, mover un archivo a un directorio, y borrar un directorio (rm -rf; recursively and force)	8
ln -s generación de una liga simbolica	8
mv para mover o renombrar una la liga o cualquier archivo o directorio	8
Visualización de contenidos de archivos	8
head y tail para desplegar la cabecera y cola de archivos	8
cat despliega uno o más archivos, concatenándolos	9
el paginador less despliega archivos página a página	9
Edición de archivos con pico vim o nedit	10
Uso de tuberías de herramientas UNIX/Linux para filtrado de texto con cut, grep, sort, uniq, wc	10
wc cuenta líneas, palabras y caracteres	10
cut , sort , uniq y pipes ' ' para procesar datos	10
man despliega el manual de cada comando	11
grep Filtra las líneas de un archivo que contienen (o no) caracteres o expresiones regulares	12
grep Filtra las líneas de un archivo que contienen (o no) caracteres o expresiones regulares	13
> redireccionado de la salida STDOUT a un archivo	15
Inicios de programación en Bash	15
asignación y uso de variables	15
condicionales if/fi if/else/fi	15
bucles for	15
alinear todos los archivo fasta en el directorio seq_data	16

Tema 1. Introducción al biocómputo en sistemas GNU/Linux

Este apunte fue creado para el módulo de Genómica del Curso de OMICAS UAEM por Pablo Vinuesa, CCG-UNAM

version: 2018-02-23

Una vez que domines los comandos básicos que se presentarán seguidamente, recomiendo revisar tutoriales mucho más detallados y completos como los siguientes:

- Bash Reference Manual
 - Advanced Bash Scripting Guide
-

Exploremos el sistema en el que estamos trabajando

ssh establecer sesion remota encriptada (segura) via ssh al servidor ivory

Abre un terminal y ejecuta el comando mostrado abajo

```
ssh -X $USER 132.248.220.19
```

hostname muestra el nombre del host (la máquina a la que estoy conectado) y la IP

```
hostname
hostname -i
```

uname muestra el sistema operativo del host

```
uname
uname -a
```

Variables de ambiente: \$HOME \$PATH \$USER

```
# imprime el contenido de la variable de ambiente $HOME
echo "$HOME"
```

```
## /home/vinuesa
```

```
# imprime el contenido de la variable de ambiente $USER
echo "$USER"
```

```
## vinuesa
```

```
# imprime el contenido de la variable de ambiente $PATH
echo "$PATH"
```

```
## /home/vinuesa/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local
```

EXPLORACIÓN EL SISTEMA DE ARCHIVOS

pwd imprime la ruta absoluta del directorio actual

```
# dónde me encuentro en el sistema?  
pwd
```

ls lista contenidos del directorio

```
# Qué contiene el directorio actual?  
ls  
  
# mostrar todos (-a all) los archivos, incluidos los ocultos  
ls -a
```

Veamos el contenido del directorio raiz

```
ls /  
  
## bin  
## boot  
## cdrom  
## dev  
## etc  
## home  
## initrd.img  
## initrd.img.old  
## lib  
## lib64  
## lost+found  
## media  
## mnt  
## opt  
## proc  
## root  
## run  
## sbin  
## snap  
## srv  
## sys  
## tmp  
## usr  
## var  
## vmlinuz  
## vmlinuz.old
```

Veamos el contenido del directorio /bin usando el comodín '*'

```
ls /bin/* | head -20  
  
## /bin/bash  
## /bin/bunzip2
```

```
## /bin/busybox
## /bin/bzcat
## /bin/bzcmp
## /bin/bzdiff
## /bin/bzegrep
## /bin/bzexe
## /bin/bzfgrep
## /bin/bzgrep
## /bin/bzip2
## /bin/bzip2recover
## /bin/bzless
## /bin/bzmore
## /bin/cat
## /bin/chacl
## /bin/chgrp
## /bin/chmod
## /bin/chown
## /bin/chvt
```

idem, pero con detalles de permisos etc de los archivos contenidos en /bin

```
ls -l /bin/* | head -20
```

```
## -rwxr-xr-x 1 root root 1037528 may 16 2017 /bin/bash
## -rwxr-xr-x 1 root root 31288 may 20 2015 /bin/bunzip2
## -rwxr-xr-x 1 root root 1964536 ago 19 2015 /bin/busybox
## -rwxr-xr-x 1 root root 31288 may 20 2015 /bin/bzcat
## lrwxrwxrwx 1 root root 6 jun 5 2017 /bin/bzcmp -> bzdiff
## -rwxr-xr-x 1 root root 2140 may 20 2015 /bin/bzdiff
## lrwxrwxrwx 1 root root 6 jun 5 2017 /bin/bzegrep -> bzgrep
## -rwxr-xr-x 1 root root 4877 may 20 2015 /bin/bzexe
## lrwxrwxrwx 1 root root 6 jun 5 2017 /bin/bzfgrep -> bzgrep
## -rwxr-xr-x 1 root root 3642 may 20 2015 /bin/bzgrep
## -rwxr-xr-x 1 root root 31288 may 20 2015 /bin/bzip2
## -rwxr-xr-x 1 root root 14616 may 20 2015 /bin/bzip2recover
## lrwxrwxrwx 1 root root 6 jun 5 2017 /bin/bzless -> bzmore
## -rwxr-xr-x 1 root root 1297 may 20 2015 /bin/bzmore
## -rwxr-xr-x 1 root root 52080 mar 2 2017 /bin/cat
## -rwxr-xr-x 1 root root 14752 feb 7 2016 /bin/chacl
## -rwxr-xr-x 1 root root 60272 mar 2 2017 /bin/chgrp
## -rwxr-xr-x 1 root root 56112 mar 2 2017 /bin/chmod
## -rwxr-xr-x 1 root root 64368 mar 2 2017 /bin/chown
## -rwxr-xr-x 1 root root 10536 sep 22 2016 /bin/chvt
```

idem, pero ordenando los archivos por fechas de modificacion (-t), listando los mas recientes al final

```
ls -ltr /bin/* | head -20
```

```
## -rwxr-xr-x 1 root root 31248 dic 3 2012 /bin/nc.openbsd
## -rwxr-xr-x 1 root root 89 abr 26 2014 /bin/red
## -rwxr-xr-x 1 root root 51512 abr 26 2014 /bin/ed
## -rwsr-xr-x 1 root root 44680 may 7 2014 /bin/ping6
## -rwsr-xr-x 1 root root 44168 may 7 2014 /bin/ping
## -rwxr-xr-x 1 root root 119624 jun 30 2014 /bin/netstat
## -rwxr-xr-x 1 root root 5047 oct 27 2014 /bin/znew
## -rwxr-xr-x 1 root root 1910 oct 27 2014 /bin/zmore
## -rwxr-xr-x 1 root root 2037 oct 27 2014 /bin/zless
```

```
## -rwxr-xr-x 1 root root 5938 oct 27 2014 /bin/zgrep
## -rwxr-xr-x 1 root root 2131 oct 27 2014 /bin/zforce
## -rwxr-xr-x 1 root root 140 oct 27 2014 /bin/zfgrep
## -rwxr-xr-x 1 root root 140 oct 27 2014 /bin/zegrep
## -rwxr-xr-x 1 root root 5764 oct 27 2014 /bin/zdiff
## -rwxr-xr-x 1 root root 1777 oct 27 2014 /bin/zcmp
## -rwxr-xr-x 1 root root 1937 oct 27 2014 /bin/zcat
## -rwxr-xr-x 1 root root 2301 oct 27 2014 /bin/uncompress
## -rwxr-xr-x 1 root root 5927 oct 27 2014 /bin/gzexe
## -rwxr-xr-x 1 root root 2301 oct 27 2014 /bin/gunzip
## -rwxr-xr-x 1 root root 98240 oct 27 2014 /bin/gzip
```

expansión de caracteres con * y ?

```
# lista los archivos en /bin que empiezan por las letras b y c
ls /bin/b*
ls /bin/c*
```

```
## /bin/bash
## /bin/bunzip2
## /bin/busybox
## /bin/bzcat
## /bin/bzcmp
## /bin/bzdiff
## /bin/bzegrep
## /bin/bzexe
## /bin/bzfgrep
## /bin/bzgrep
## /bin/bzip2
## /bin/bzip2recover
## /bin/bzless
## /bin/bzmore
## /bin/cat
## /bin/chacl
## /bin/chgrp
## /bin/chmod
## /bin/chown
## /bin/chvt
## /bin/cp
## /bin/cpio
```

```
# lista los archivos en /bin que empiezan por la letra c seguida de uno o dos caracteres más
ls /bin/c?
ls /bin/c??
```

```
## /bin/cp
## /bin/cat
```

cd para cambiar de directorios usando rutas absolutas y relativas

de nuevo, donde estoy?

```
pwd
```

```
## /home/vinuesa/Dropbox/move/OMICAS_UAEM_genomica/clase1/intro2linux
```

sube un directorio usando RUTA RELATIVA

```
cd ..
```

```
# o cd ../
```

donde estoy?

```
pwd
```

```
## /home/vinuesa/Dropbox/move/OMICAS_UAEM_genomica/clase1/intro2linux
```

regresa a tu home

```
cd $HOME
```

```
# que es equivalente a:
```

```
cd
```

cd cambiar directorios con rutas absolutas (/ruta/completa/al/dir) y relativas ../../

```
# a dónde nos lleva este comando?
```

```
cd /
```

- cambia de nuevo a tu home

```
cd
```

```
# o tambien usando la variable de ambiente $HOME
```

```
cd $HOME
```

Generación de directorios y permisos, copiado de archivos y directorios

mkdir genera nuevos directorios

```
# vamos a $HOME y generamos el directorio intro2genomics
```

```
cd
```

```
if [ -d intro2genomics ]; then
```

```
    echo "found dir intro2genomics"
```

```
else
```

```
mkdir intro2genomics
fi
```

```
## found dir intro2genomics
```

ls -l lista los permisos de archivos y directorios

- comprueba los **permisos** del nuevo directorio

```
ls -l
```

```
## total 2804
## -rw-rw-r-- 1 vinuesa vinuesa      78 feb 23 11:18 administration.cmds
## -rw-rw-r-- 1 vinuesa vinuesa    492 feb 23 11:18 Batch.cmds
## -rw-rw-r-- 1 vinuesa vinuesa      75 feb 23 11:18 Category.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   2835 feb 23 11:18 C.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   1814 feb 23 11:18 Filesystem.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   1814 feb 23 12:12 Filesystem_commands.tab
## -rw-rw-r-- 1 vinuesa vinuesa      54 feb 23 11:18 FORTRAN77.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   2346 feb 23 11:18 grep
## drwxrwxr-x 3 vinuesa vinuesa   4096 feb 23 11:20 intro2genomics
## drwxrwxr-x 2 vinuesa vinuesa   4096 feb 23 12:00 intro2linux
## -rw-rw-r-- 1 vinuesa vinuesa 1598266 feb 23 11:18 intro_biocomputo_Linux_pt1.odp
## -rw-rw-r-- 1 vinuesa vinuesa   10193 feb 23 11:18 linux_commands.tab
## -rw-rw-r-- 1 vinuesa vinuesa   1705 feb 23 11:18 linux_very_basic_commands_table.csv
## -rw-rw-r-- 1 vinuesa vinuesa   1042 feb 23 11:18 management.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   2133 feb 23 11:18 Misc.cmds
## -rw-rw-r-- 1 vinuesa vinuesa    215 feb 23 11:18 Network.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   1138 feb 23 11:18 Process.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   2000 feb 23 11:18 processing.cmds
## -rw-rw-r-- 1 vinuesa vinuesa     81 feb 23 11:18 programming (Case Conflict).cmds
## -rw-rw-r-- 1 vinuesa vinuesa   1735 feb 23 11:18 programming.cmds
## -rw-rw-r-- 1 vinuesa vinuesa    528 feb 23 11:18 SCCS.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   1038 feb 23 11:18 Shell.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   1394 feb 23 11:18 System.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   2000 feb 23 11:18 Text.cmds
## -rw-rw-r-- 1 vinuesa vinuesa    567 feb 23 11:18 utilities.cmds
## -rw-rw-r-- 1 vinuesa vinuesa   6047 feb 23 11:18 working_with_linux_commands.code
## -rw-rw-r-- 1 vinuesa vinuesa  781253 feb 23 12:05 working_with_linux_commands.html
## -rw-rw-r-- 1 vinuesa vinuesa  241335 feb 23 12:12 working_with_linux_commands.pdf
## -rw-rw-r-- 1 vinuesa vinuesa   11782 feb 23 12:15 working_with_linux_commands.Rmd
```

- generemos un subdirectorio por debajo del que acabamos de crear:

```
mkdir -p intro2genomics/sesion1_linux && cd intro2genomics/sesion1_linux
```

- cambiamos a /home/vinuesa e intenta crear estos mismos directorios ahí

```
cd /home/vinuesa && mkdir -p intro2genomics/sesion1_linux
```

copiar, mover, renombrar y borrar archivos

Vamos a copiar los archivos que necesitamos para lo que sigue de las prácticas a subdirectorios de nuestro \$HOME

```
# cambia a tu home, y luego a intro2genomics/sesion1_linux
cd && cd intro2genomics/sesion1_linux
```

- copia el archivo /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab al directorio actual

Noten el punto al final del comando, significa: copia X en el dir actual

```
cp /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab .
```

- otra manera, usando rutas absolutas y la variable de ambiente \$HOME

```
cp /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab $HOME/intro2genomics
```

- copiar el directorio /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/ a tu dir actual

Noten el punto '.' y cp -r (recursively), necesario para copiar directorios completos

```
cp -r /home/vinuesa/cursos/intro2genomics/sesion1_linux/data .
```

generar un directorio, mover un archivo a un directorio, y borrar un directorio (rm -rf; recursively and force)

```
mkdir borrame
```

```
mv linux_basic_commands.tab borrame
```

```
ls borrame
```

```
rm -rf borrame
```

```
## mv: cannot stat 'linux_basic_commands.tab': No such file or directory
```

ln -s generación de una liga simbolica

Esto es muy importante, ya que permite ahorrar mucho espacio en disco al evitar la multiplicación de copias físicas en el disco duro del mismo archivo de gran tamaño (no es el caso de linux_basic_commands.tab)

generamos la liga simbólica en nuestro directorio de trabajo

```
ln -s /home/vinuesa/cursos/intro2genomics/sesion1_linux/data/linux_basic_commands.tab comandos_de_linux
```

confirmamos que se genero la liga

```
ls -l
```

mv para mover o renombrar una la liga o cualquier archivo o directorio

```
mv comandos_de_linux.tab linux_commands.tab
```

Visualización de contenidos de archivos

head y tail para desplegar la cabecera y cola de archivos

```
head linux_commands.tab
```



```
## IEEE Std 1003.1-2008 utilities Name Category Description First appeared
## admin SCCS Create and administer SCCS files PWB UNIX
## alias Misc Define or display aliases
## ar Misc Create and maintain library archives Version 1 AT&T UNIX
## asa Text processing Interpret carriage-control characters System V
## at Process management Execute commands at a later time Version 7 AT&T UNIX
## awk Text processing Pattern scanning and processing language Version 7 AT&T UNIX
## basename Filesystem Return non-directory portion of a pathname; see also dirname Version 7 A
## batch Process management Schedule commands to be executed in a batch queue
## bc Misc Arbitrary-precision arithmetic language Version 6 AT&T UNIX
```

```
tail linux_commands.tab
```

```
## val SCCS Validate SCCS files System III
## vi Text processing Screen-oriented (visual) display editor 1BSD
## wait Process management Await process completion Version 4 AT&T UNIX
## wc Text processing Line, word and byte or character count Version 1 AT&T UNIX
## what SCCS Identify SCCS files PWB UNIX
## who System administration Display who is on the system Version 1 AT&T UNIX
## write Misc Write to another user's terminal Version 1 AT&T UNIX
## xargs Shell programming Construct argument lists and invoke utility PWB UNIX
## yacc C programming Yet another compiler compiler PWB UNIX
## zcat Text processing Expand and concatenate data 4.3BSD
```

le podemos indicar el numero de lineas a desplegar

```
head -3 linux_commands.tab
```

```
## IEEE Std 1003.1-2008 utilities Name Category Description First appeared
## admin SCCS Create and administer SCCS files PWB UNIX
## alias Misc Define or display aliases
```

```
tail -1 linux_commands.tab
```

```
## zcat Text processing Expand and concatenate data 4.3BSD
```

cat despliega uno o más archivos, concatenándolos

```
cat linux_commands.tab | head
```

```
## IEEE Std 1003.1-2008 utilities Name Category Description First appeared
## admin SCCS Create and administer SCCS files PWB UNIX
## alias Misc Define or display aliases
## ar Misc Create and maintain library archives Version 1 AT&T UNIX
## asa Text processing Interpret carriage-control characters System V
## at Process management Execute commands at a later time Version 7 AT&T UNIX
## awk Text processing Pattern scanning and processing language Version 7 AT&T UNIX
## basename Filesystem Return non-directory portion of a pathname; see also dirname Version 7 A
## batch Process management Schedule commands to be executed in a batch queue
## bc Misc Arbitrary-precision arithmetic language Version 6 AT&T UNIX
```

el paginador less despliega archivos página a página

```
less linux_commands.tab | head
```

```
## IEEE Std 1003.1-2008 utilities Name Category Description First appeared
```

```
## admin    SCCS      Create and administer SCCS files      PWB UNIX
## alias    Misc      Define or display aliases
## ar       Misc      Create and maintain library archives   Version 1 AT&T UNIX
## asa      Text processing  Interpret carriage-control characters   System V
## at       Process management  Execute commands at a later time       Version 7 AT&T UNIX
## awk      Text processing  Pattern scanning and processing language Version 7 AT&T UNIX
## basename Filesystem      Return non-directory portion of a pathname; see also dirname Version 7 AT&T UNIX
## batch    Process management  Schedule commands to be executed in a batch queue
## bc       Misc      Arbitrary-precision arithmetic language   Version 6 AT&T UNIX
```

Edición de archivos con pico vim o nedit

```
pico linux_commands.tab
# usa ctrl-X para salir
```

```
# vim es un editor muy potente, usado por defecto en los sistemas Linux.
# Es muy conveniente aprender a usarlo, pero requiere tiempo del que no disponemos aquí
# pueden usar el comando 'vimtutor' para estudiar el tutorial integrado de vim
```

```
# 'nedit' es un editor con ambiente gráfico, que usaremos rutinariamente en nuestras prácticas
# Noten el uso de & al final de la sentencia para enviar el proceso al fondo
# para evitar que bloquee la terminal
nedit linux_commands.tab &
```

Uso de tuberías de herramientas UNIX/Linux para filtrado de texto con cut, grep, sort, uniq, wc

wc cuenta líneas, palabras y caracteres

```
# dame el número de líneas, palabras y caracteres del archivo
wc linux_commands.tab

# Cuántas líneas tiene el archivo linux_commands.tab?
wc -l linux_commands.tab
```

```
## 161 1553 10193 linux_commands.tab
## 161 linux_commands.tab
```

cut, sort, uniq y pipes '|' para procesar datos

- cut corta líneas de texto/tablas por delimitadores de campo específicos extrayendo campos particulares
- sort ordena
- uniq regresa listas de valores únicos
- el pipe '|' conecta la salida de un comando con la entrada de otro

```
# saca la segunda columna del archivo y ordenalos
cut -f2 linux_commands.tab | head
```

```
## Category
## SCCS
## Misc
## Misc
```

```
## Text processing
## Process management
## Text processing
## Filesystem
## Process management
## Misc
```

```
# cuantos elementos unicos tiene?
cut -f2 linux_commands.tab | sort -u | wc -l
```

```
## 13
```

```
# cuenta las ocurrencias de las diferentes palabras contenidas en la segunda columna del archivo linux_
cut -f2 linux_commands.tab | sort | uniq -c
```

```
##      11 Batch utilities
##       1 Category
##       9 C programming
##      28 Filesystem
##       1 FORTRAN77 programming
##      38 Misc
##       4 Network
##      14 Process management
##       1 Programming
##      10 SCCS
##      14 Shell programming
##       1 System administration
##      29 Text processing
```

man despliega el manual de cada comando

```
# mira las opciones de cut y sort en la manpage
man cut | head -30
```

```
## CUT(1)
##
## NAME
##      cut - remove sections from each line of files
##
## SYNOPSIS
##      cut OPTION... [FILE]...
##
## DESCRIPTION
##      Print selected parts of lines from each FILE to standard output.
##
##      With no FILE, or when FILE is -, read standard input.
##
##      Mandatory arguments to long options are mandatory for short options too.
##
##      -b, --bytes=LIST
##              select only these bytes
##
##      -c, --characters=LIST
##              select only these characters
##
```

User Commands

```
##      -d, --delimiter=DELIM
##          use DELIM instead of TAB for field delimiter
##
##      -f, --fields=LIST
##          select only these fields;  also print any line that contains no delimiter character, u
##
##      -n      (ignored)
##
##      --complement
man sort | head -20
```

```
## SORT(1) User Commands
##
## NAME
##      sort - sort lines of text files
##
## SYNOPSIS
##      sort [OPTION]... [FILE]...
##      sort [OPTION]... --files0-from=F
##
## DESCRIPTION
##      Write sorted concatenation of all FILE(s) to standard output.
##
##      With no FILE, or when FILE is -, read standard input.
##
##      Mandatory arguments to long options are mandatory for short options too.  Ordering options:
##
##      -b, --ignore-leading-blanks
##          ignore leading blanks
##
##      -d, --dictionary-order
```

grep Filtra las líneas de un archivo que contienen (o no) caracteres o expresiones regulares

```
grep Filesystem linux_commands.tab | head
```

```
## basename      Filesystem Return non-directory portion of a pathname; see also dirname      Version 7 A
## cat   Filesystem Concatenate and print files      Version 1 AT&T UNIX
## cd    Filesystem Change the working directory      Version 6 AT&T UNIX
## chgrp  Filesystem Change the file group ownership      PWB UNIX
## chmod  Filesystem Change the file modes/attributes/permissions      Version 1 AT&T UNIX
## chown  Filesystem Change the file ownership      Version 1 AT&T UNIX
## cksum  Filesystem Write file checksums and sizes      4.4BSD
## cmp    Filesystem Compare two files; see also diff      Version 1 AT&T UNIX
## compress      Filesystem Compress data      4.3BSD
## cp    Filesystem Copy files      Version 1 AT&T UNIX
```

ojo: Linux es sensible a mayúsculas y minúsculas: prueba este comando para comprobarlo

```
grep filesystem linux_commands.tab
```

grep -i lo hace insensible a la fuente

```
grep filesystem linux_commands.tab | head
```

```
### expansión de caracteres con * y ?
```

```
# lista los archivos en /bin que empiezan por las letras b y c
```

```
ls /bin/b*  
ls /bin/c*
```

```
## /bin/bash  
## /bin/bunzip2  
## /bin/busybox  
## /bin/bzcat  
## /bin/bzcmp  
## /bin/bzdiff  
## /bin/bzegrep  
## /bin/bzexe  
## /bin/bzfgrep  
## /bin/bzgrep  
## /bin/bzip2  
## /bin/bzip2recover  
## /bin/bzless  
## /bin/bzmore  
## /bin/cat  
## /bin/chacl  
## /bin/chgrp  
## /bin/chmod  
## /bin/chown  
## /bin/chvt  
## /bin/cp  
## /bin/cpio
```

```
# lista los archivos en /bin que empiezan por la letra c seguida de uno o dos caracteres más
```

```
ls /bin/c?  
ls /bin/c??
```

```
## /bin/cp  
## /bin/cat
```

grep Filtra las líneas de un archivo que contienen (o no) caracteres o expresiones regulares

```
grep Filesystem linux_commands.tab | head
```

```
## basename      Filesystem  Return non-directory portion of a pathname; see also dirname      Version 7 AT&T UNIX  
## cat           Filesystem  Concatenate and print files      Version 1 AT&T UNIX  
## cd           Filesystem  Change the working directory     Version 6 AT&T UNIX  
## chgrp        Filesystem  Change the file group ownership  PWB UNIX  
## chmod        Filesystem  Change the file modes/attributes/permissions      Version 1 AT&T UNIX  
## chown        Filesystem  Change the file ownership        Version 1 AT&T UNIX  
## cksum        Filesystem  Write file checksums and sizes   4.4BSD  
## cmp          Filesystem  Compare two files; see also diff  Version 1 AT&T UNIX  
## compress     Filesystem  Compress data                    4.3BSD  
## cp           Filesystem  Copy files                       Version 1 AT&T UNIX
```

```
# ojo: Linux es sensible a mayúsculas y minúsculas: prueba este comando para comprobarlo  
grep filesystem linux_commands.tab
```

```
# grep -i lo hace insensible a la fuente
grep filesystem linux_commands.tab | head
```

```
# filtra las lineas que contienen Filesystem o Text processing
egrep 'Filesystem|Text processing' linux_commands.tab
```

```
## asa Text processing Interpret carriage-control characters System V
## awk Text processing Pattern scanning and processing language Version 7 AT&T UNIX
## basename Filesystem Return non-directory portion of a pathname; see also dirname Version 7 AT&T UNIX
## cat Filesystem Concatenate and print files Version 1 AT&T UNIX
## cd Filesystem Change the working directory Version 6 AT&T UNIX
## chgrp Filesystem Change the file group ownership PWB UNIX
## chmod Filesystem Change the file modes/attributes/permissions Version 1 AT&T UNIX
## chown Filesystem Change the file ownership Version 1 AT&T UNIX
## cksum Filesystem Write file checksums and sizes 4.4BSD
## cmp Filesystem Compare two files; see also diff Version 1 AT&T UNIX
## comm Text processing Select or reject lines common to two files Version 4 AT&T UNIX
## compress Filesystem Compress data 4.3BSD
## cp Filesystem Copy files Version 1 AT&T UNIX
## csplit Text processing Split files based on context PWB UNIX
## cut Text processing Cut out selected fields of each line of a file System III
## dd Filesystem Convert and copy a file Version 5 AT&T UNIX
## df Filesystem Report free disk space Version 1 AT&T UNIX
## diff Text processing Compare two files; see also cmp Version 5 AT&T UNIX
## dirname Filesystem Return the directory portion of a pathname; see also basename System III
## du Filesystem Estimate file space usage Version 1 AT&T UNIX
## ed Text processing The standard text editor Version 1 AT&T UNIX
## ex Text processing Text editor 1BSD
## expand Text processing Convert tabs to spaces 3BSD
## file Filesystem Determine file type Version 4 AT&T UNIX
## find Filesystem Find files Version 1 AT&T UNIX
## fold Text processing Filter for folding lines 1BSD
## head Text processing Copy the first part of files PWB UNIX[citation needed]
## iconv Text processing Codeset conversion HP-UX
## join Text processing Merges two sorted text files based on the presence of a common field
## link Filesystem Create a hard link to a file Version 1 AT&T UNIX
## ln Filesystem Link files Version 1 AT&T UNIX
## lp Text processing Send files to a printer System V
## ls Filesystem List directory contents Version 1 AT&T UNIX
## mkdir Filesystem Make directories Version 1 AT&T UNIX
## mkfifo Filesystem Make FIFO special files 4.4BSD[dubious - discuss]
## more Text processing Display files on a page-by-page basis 3BSD
## mv Filesystem Move files Version 1 AT&T UNIX
## nl Text processing Line numbering filter System III
## paste Text processing Merge corresponding or subsequent lines of files Version 32V AT&T UNIX
## patch Text processing Apply changes to files 4.3BSD
## pathchk Filesystem Check pathnames
## pr Text processing Print files Version 1 AT&T UNIX
## pwd Filesystem print working directory - Return working directory name Version 5 AT&T UNIX
## rm Filesystem Remove directory entries Version 1 AT&T UNIX
## rmdir Filesystem Remove directories Version 1 AT&T UNIX
## sed Text processing Stream editor Version 7 AT&T UNIX
## sort Text processing Sort, merge, or sequence check text files Version 1 AT&T UNIX
## tail Text processing Copy the last part of a file PWB UNIX[citation needed]
```

```
## touch      Filesystem  Change file access and modification times  Version 7 AT&T UNIX
## tr   Text processing  Translate characters      Version 4 AT&T UNIX
## tsort  Text processing  Topological sort         Version 7 AT&T UNIX
## unexpand Text processing  Convert spaces to tabs   3BSD
## uniq   Text processing  Report or filter out repeated lines in a file  Version 3 AT&T UNIX
## unlink Filesystem  Call the unlink function  Version 1 AT&T UNIX
## vi   Text processing  Screen-oriented (visual) display editor      1BSD
## wc   Text processing  Line, word and byte or character count  Version 1 AT&T UNIX
## zcat Text processing  Expand and concatenate data      4.3BSD
```

> redireccionado de la salida STOUT a un archivo

```
grep Filesystem linux_commands.tab > Filesystem_commands.tab
```

Inicios de programación en Bash

asignación y uso de variables

```
# vamos al home
cd

# guardamos la salida del comando pwd en una variable que llamamos wkdir
wkdir=$(pwd)

# para recuperar el valor de la variable tenemos que dereferenciarla con $
echo "working in: $wkdir"

## working in: /home/vinuesa
```

condicionales if/fi if/else/fi

```
if [ -d intro2linux ]; then
    echo "dir intro2linux exists, will move into it"
    cd intro2linux
else
    mkdir intro2linux && cd intro2linux
fi

## dir intro2linux exists, will move into it
```

bucles for

```
#>>> AVANZADO: usa un bucle for, acoplado a las herramientas de filtrado arriba mostradas,
#                para generar archivos que contengan solo los comandos de las diferentes categorias
#                nombrando a los archivos por estas

for type in $(cut -f2 linux_basic_commands.tab | sort -u); do
    grep "$type" linux_basic_commands.tab > ${type}.cmds
done
```

```
# o en una línea for type in $(cut -f2 linux_basic_commands.tab | sort -u); do grep "$type" linux_basic.
```

```
## cut: linux_basic_commands.tab: No such file or directory
```

veamos el resultado de ejecutar el bucle anterior:

```
ls *cmds
```

```
## administration.cmds
## Batch.cmds
## Category.cmds
## C.cmds
## Filesystem.cmds
## FORTRAN77.cmds
## management.cmds
## Misc.cmds
## Network.cmds
## Process.cmds
## processing.cmds
## programming (Case Conflict).cmds
## programming.cmds
## SCCS.cmds
## Shell.cmds
## System.cmds
## Text.cmds
## utilities.cmds
```

alinear todos los archivo fasta en el directorio seq_data

Este es un ejemplo hipotético. Asumimos que tenemos un directorio seq_data con las secuencias a alinear en formato FASTA.

1. ir al directorio con las secuencias

```
cd $HOME/seq_data
```

```
# 2. corremos el alineador muscle dentro de un bucle for
#   pasándole cada archivo fasta del directorio en forma
#   del alias $f; muscle lee $f (muscle < $f) y escribe
#   el lineamiento a un nuevo archivo (> ${f%.*}_musAln.fas)
#   que se llamará así: nombre_de_archivo_original_musAln.fas
for f in *fasta; do muscle < $f > ${f%.*}_musAln.fas; done
```