Linux

Comandos básicos

Gustavo C. Distel
gd@cs.uns.edu.ar
D.C.I.C. – U.N.S.

Motivación

- Académica
- Ideológica
- Económica

Virtual terminal

- El kernel de Linux soporta terminales virtuales, dispositivo usado para interactuar con la terminal física.

- Se conoce como "virtual" dado que se pueden ejecutar varias terminales virtuales (también llamadas consolas virtuales) en una misma terminal física.

- Tienen muchas utilidades; por ejemplo, una terminal virtual puede mostrar mensajes del sistema y warnings, otra se puede utilizar para una sesión de usuario en modo texto y una tercera podría ejecutar una sesión X, todo en paralelo.

- Para cambiar entre terminales virtuales se usan determinadas combinaciones de teclas, por lo general Alt-<function key>, desde una sesión X; Ctrl-Alt-<function key>.

Virtual terminal

- Se necesita al menos una terminal virtual para poder hacer uso del teclado y el monitor (excepto que estemos configurando un sistema embebido).

- La consola del sistema es el dispositivo que recibe todos los mensajes del kernel, advertencias y que permite el inicio de sesión en modo monousuario.

- Por defecto, la terminal virtual actualmente visible (/dev/tty0) se utiliza como consola del sistema.

SHELL

- En Linux con frecuencia se utiliza un tipo especial de programa llamado shell para interactuar con el kernel.

- La shell se ejecuta dentro de una terminal emitiendo un "prompt“ y esperando que se le diga qué hacer.
  - **Nota**: El prompt es configurable; por lo general aparece el nombre de usuario, hostname y el directorio actual.

- Luego, el usuario le pide a la shell que ejecute un programa escribiendo el nombre del mismo.

- Hay varias implementaciones para una shell, por ej.: bash, zsh, ksh, csh, tcsh.

BASH

- Es una shell Unix escrita por Brian Fox para el proyecto GNU como reemplazo en software libre de la shell de Bourne (sh).

- Su nombre es un acrónimo de *Bourne-Again Shell* (otro shell bourne), haciendo un juego de palabras (born-again significa renacimiento) sobre el Bourne shell (sh), que fue uno de los primeros intérpretes importantes de Unix.

Shell interactiva vs. Scripts de shell

- La shell bash está diseñada para dos diferentes tipos de uso.
- A continuación usaremos bash como una shell interactiva.
- Bash también está diseñada para ser un lenguaje de escritura de gran alcance: scripts de la shell bash.
- Los scripts de la shell bash son programas pequeños escritos mediante la misma sintaxis que se utiliza en la línea de comandos.
- Los scripts de shell permiten automatizar las acciones repetidas al combinar una serie de comandos.

¿Por qué una consola y no un entorno gráfico?

- En primer lugar, la máquina puede no tener suficiente memoria u otro hardware para soportar el entorno gráfico y en consecuencia sólo se encuentran disponibles las consolas.

- Sin embargo, incluso en máquinas modernas, es muy frecuente que las consolas sean una interfaz más rápida y más eficaz para los usuarios experimentados.

- Los usuarios experimentados de Linux prefieren la versatilidad y potencia de la interfaz de la línea de comandos en lugar de las aplicaciones gráficas.

- Dado que los servidores de red usualmente pasan la vida encerrados en una sala, los administradores quieren a menudo evitar la complejidad y “suciedad” que la interfaz gráfica le añade al sistema.

Terminales

- tty: teletype - *imprime el nombre de archivo de la terminal conectada al standard input*.
  - Terminales Virtuales /dev/tty<1..6>
  - Pseudo Terminales /dev/pts/<0...>

- echo $SHELL - *imprime una variable de entorno para saber que shell estamos usando.*
  - **Nota**: echo muestra una línea de texto
  - Variable de entorno:
    - Las variables de entorno forman un conjunto de valores dinámicos que normalmente afectan el comportamiento de los procesos en una computadora.
    - Es posible acceder al valor de una variable, ya sea para consultarlo como para modificarlo, tanto dentro de scripts como desde la línea de comandos.
    - Variables de entorno muy usadas son: $PATH, $HOME, $USER, $HOTNAME, entre otras.

Sistemas de Archivos

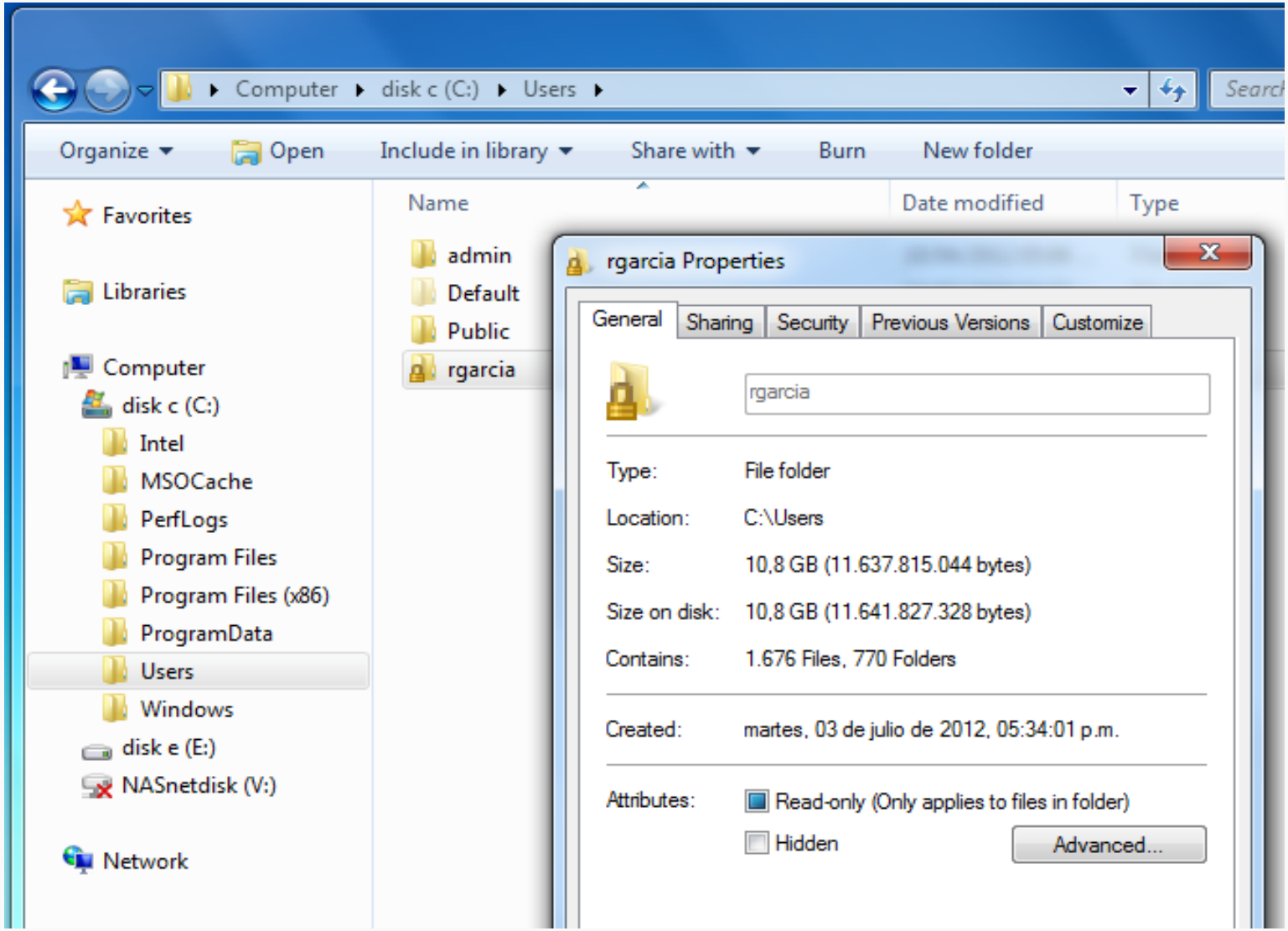
- Se verá en profundidad al final de la materia.

- Explicación breve para saber “donde están las cosas”; luego veremos como hacer para manipularlas a través de comandos.

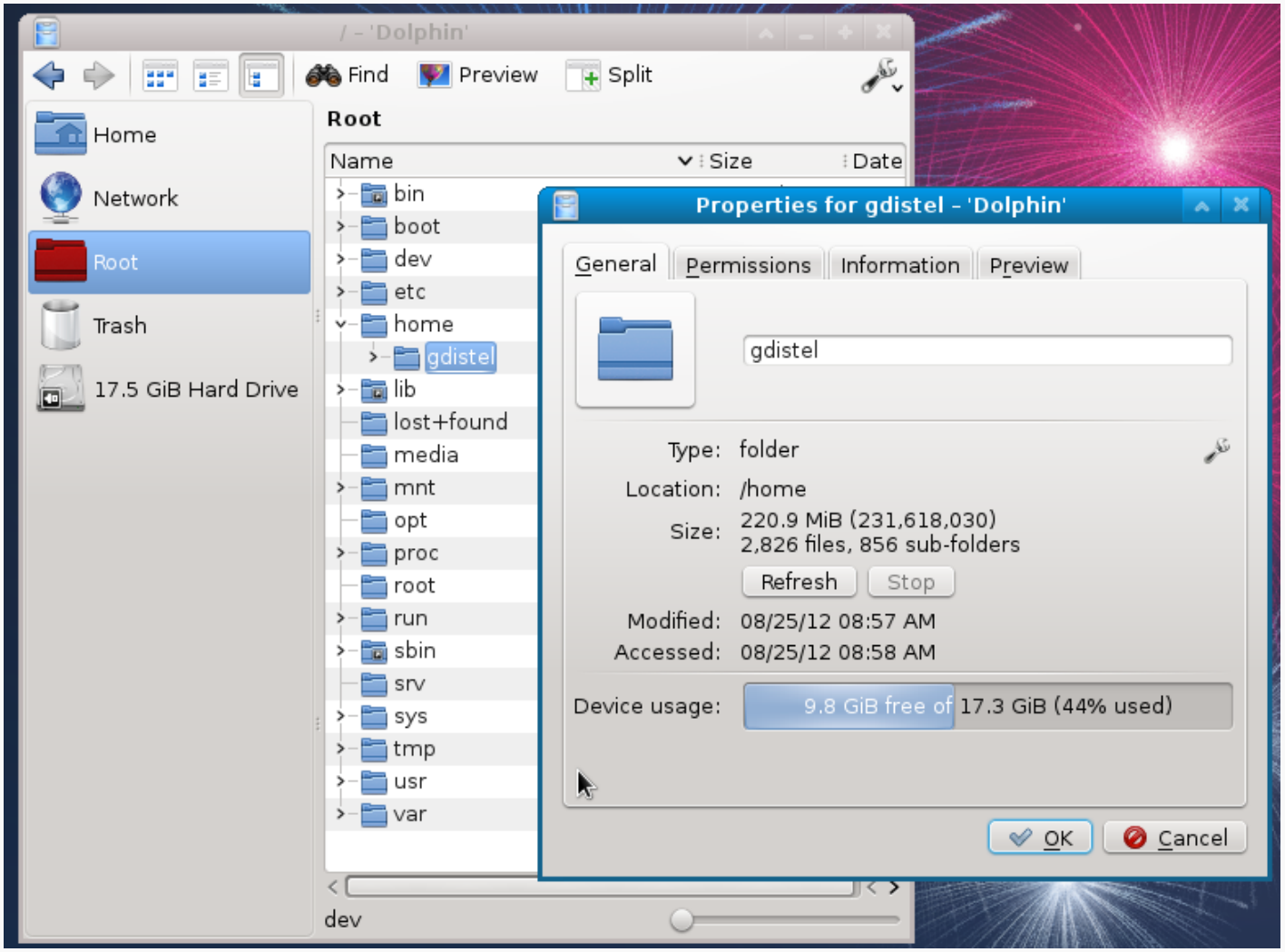
Archivos y carpetas

- La estructura de directorios suele ser jerárquica, ramificada o “en árbol”.
- Se declara la ubicación precisa de un archivo con una cadena de texto llamada "ruta" (o *path* en inglés).
- La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen una misma estructura.
- Ej.: en Linux se utiliza una diagonal ('/') y en Windows una diagonal invertida ('\').

Windows


Computer ▸ disk c (C:) ▸ Users ▸
Organize
Open
Include in library
Share with
Burn
New folder
Favorites
Libraries
Computer
disk c (C:)
Intel
MSOCache
PerfLogs
Program Files
Program Files (x86)
ProgramData
Users
Windows
disk e (E:)
NASnetdisk (V:)
Network
Name
Date modified
Type
admin
Default
Public
rgarcia
rgarcia Properties
General
Sharing
Security
Previous Versions
Customize
rgarcia
Type: File folder
Location: C:\Users
Size: 10,8 GB (11.637.815.044 bytes)
Size on disk: 10,8 GB (11.641.827.328 bytes)
Contains: 1.676 Files, 770 Folders
Created: martes, 03 de julio de 2012, 05:34:01 p.m.
Attributes:
Read-only (Only applies to files in folder)
Hidden
Advanced...

Linux


/ – 'Dolphin'
Find
Preview
Split
Home
Network
Root
Trash
17.5 GiB Hard Drive
Root
Name
Size
Date
bin
boot
dev
etc
home
gdistel
lib
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
dev
Properties for gdistel – 'Dolphin'
General
Permissions
Information
Preview
gdistel
Type: folder
Location: /home
Size: 220.9 MiB (231,618,030)
2,826 files, 856 sub-folders
Refresh
Stop
Modified: 08/25/12 08:57 AM
Accessed: 08/25/12 08:58 AM
Device usage: 9.8 GiB free of 17.3 GiB (44% used)
OK
Cancel

Estándar de jerarquía del sistema de archivos (FHS)

FHS

- El estándar de jerarquía del sistema de archivos (FHS: File System Hierarchy Standard) es una norma que define los directorios principales y sus contenidos en el sistema operativo GNU/Linux y otros sistemas de la familia Unix.

- Ventaja: es más fácil escribir software, portar software y administrar máquinas de Linux, dado que todo debería estar en lugares estandarizados.

- No hay una autoridad detrás de la norma que obligue a cumplir con ella, pero se ha ganado el apoyo de muchas distribuciones de Linux.

- Esto no evita que algunas hagan sus propias modificaciones sobre la misma o que no estén de acuerdo.

FHS

- '/' Jerarquía primaria: la raíz (*root*) y directorio raíz o contenedor de todo el sistema de jerarquía.
  - '/bin/' Comandos esenciales para todos los usuarios ej.: ls, cp, etc.
  - '/boot/' Archivos cargadores de arranque; ej.: los núcleos (kernels).
  - '/etc/' Archivos de configuración del sistema.
  - '/home/' Directorio home de los usuarios; ej.: /home/gdistel.
  - '/lib/' Bibliotecas de programas, para binarios en /bin/ y /sbin/.
  - '/media/' Puntos de montaje de medios extraíbles (CD-rom, Pendrives).
  - '/mnt/' Sistema de archivos montados temporalmente.
  - '/root/' Directorio raíz del usuario root.
  - '/sbin/' Comandos y programas exclusivos del usuario root.
  - '/usr/' Jerarquía secundaria de los datos de usuario.
  - '/var/' Archivos variables, tales como logs, bases de datos, etc.

Comandos

- Un comando es una orden que se le da a un programa que actúa como intérprete del mismo, para así realizar una tarea específica.

- La mayoría de los comandos están formados por acrónimos.

¿Que es un acrónimo según la RAE?

- Tipo de sigla que se pronuncia como una palabra; por ej., o(bjeto) v(olador) n(o) i(dentificado).
- Vocablo formado por la unión de elementos de dos o más palabras, tomando generalmente del primer elemento el inicio y del segundo el final.

- A continuación daremos una Referencia de Comandos, agrupados según el tipo.
- Al ejecutar un comando se puede verificar el estado de finalización con: echo$?

Sistema de archivos

- pwd: print working directory – *imprimir directorio de trabajo.*
- ls: list – *listar.*
  - Por defecto muestra la lista en orden alfabético y sin ocultos.
  - Ejemplos: `ls -l; ls -a; ls -la; ls -l prueba*; ls -l *eba*`
- cd: change directory – *cambiar directorio.*
  - **Nota**: '.' directorio actual, y '..' un nivel más arriba.
  - **Nota**: Los nombres de directorios y archivos son case-sensitive.
- mkdir: make directory – *crear directorio.*
- cp: copy – *copiar.*
  - Uso: cp <source> <destination>.
- mv: move – *mover.*
- rm: remove – *borrar.*
  - `]$ rm -rf` - borra directorios recursivamente (usar con precaución).

Sistema de archivos

- cat: catenate, sinonimo de concatenate - *concatenar y listar archivos*.
  - `cat prueba1.txt; cat -n prueba1.txt; cat /etc/passwd`
- stat: status - *muestra información del estado de archivos o sistema de archivos.*
  - `stat prueba1.txt`
  - `stat -f /`
- touch: - *modifica las marcas de tiempo de un archivo. También se utiliza para crear un nuevo archivo vacío.*
  - `stat prueba1.txt`
  - `touch prueba1.txt.`
  - `stat prueba1.txt`

Sistema de archivos

- du: disk usage - *espacio usado por archivo o directorio*.
  - `du; du -h; du -sh`
- df: disk free - *cantidad de espacio disponible en disco*.
  - `df -h`
- mount: montar un sistema de archivos.
  - Sin parámetros muestra todos los sistemas de archivos montados.
  - Se verá en profundidad cuando se trate en la materia el tema de sistema de archivos.

Procesamiento de textos

- head: *muestra el comienzo de un archivo de texto*.
  - `head -n 20 prueba1.txt`
- tail: *muestra las últimas líneas de un archivo*.
- wc: word count - *lee de la entrada estándar o de una lista de archivos y genera estadísticas.*
  - La estadística cuenta con: número de saltos de línea, número de palabras y cantidad de bytes.
  - `wc prueba1.txt`
  - Nota: luego veremos como usarlo en un pipe.
- less/more: *muestra el contenido de un archivo de texto por pantalla.*
- sort: *ordena las líneas de un archivo de texto.*
  - `sort -n prueba1.txt`
  - `du /bin/* | sort -n`
  - `ls -s | sort -n`
- vi/vim: *editor de texto.*

Búsqueda

- which: *muestra el path completo de un comando.*
  - Se usa para buscar un archivo binario.
  - **Nota**: Busca en los directorios de la variable de entorno PATH.

- whatis: *muestra descripciones de las páginas de manual.*
  - Cada página de manual contiene una pequeña descripción; whatis busca en ésta el argumento pasado.

- whereis: *Usado para buscar más información sobre un comando, como el binario, el fuente o la página de manual.*

Búsqueda

- grep: *se usa en la búsqueda de patrones (pattern)*.
  - grep pattern <archivo>
  - grep -r pattern <directorio>
  - comando | grep pattern
  - grep TouchPad /var/log/Xorg.0.log
- locate: *busca archivos por nombre*.
  - **Nota**: usa una base de datos para la búsqueda (/var/lib/mlocate/mlocate.db).
- find: *busca archivos en una jerarquía de directorios*.
  - Uso: find <ruta> <expresión>.
  - find /etc -name '*conf'
  - find /home -name '.bash*'
  - find ~ -name '*txt' , archivos que terminan con txt.

Información de sistema

- uname: unix name - *muestra el nombre, versión y otros detalles de la computadora actual y del sistema operativo en la misma*.
- lscpu: list cpu - *muestra información del CPU*.
  - **Nota**: Generalmente los comandos del sistema obtienen información de /proc.
- lsof: list open files - *muestra una lista de todos los archivos abiertos y de los procesos que los abrieron*.
- lshw: list hardware - *muestra información de hardware*.
  - **Nota**: no viene por defecto en Fedora 19; para instalarlo: yum install lshw.
  - **Nota**: En algunos casos lshw y lscpu discrepan sobre cachés.
- lspci: list pci - *muestra información de PCI buses y dispositivos*.
- lsusb: list usb - *idem anterior, pero para USB buses*.

Redireccionando stdin y stdout

- command1 > file1 (Clobber mode): *ejecuta command1, colocando el resultado en archivo1*.
  - **nota**: clobbering: es sobreescribir el contenido de un archivo o memoria.
  - **nota**: mueve el contenido desde STDOUT(pantalla) a un archivo.
  - cat > pruebaCat.txt, escribir y finalar con <Control-D> → crear archivo.
  - cat prueba1.txt prueba2.txt > losdosjuntos.txt
- command1 >> file1 (Append): *agrega la salida al final del archivo*.
  - cat prueba3.txt >> losdosjuntos.txt
- command1 < file1 (STDIN, leer de archivo): *ejecuta command1, con archivo1 como entrada (en contrapartida del teclado)*.
  - wc -l prueba1.txt

- Ejercicio: Ver como direccionar standard error a un archivo.

Unix Pipe

- command1 | command2: *ejecuta command1, y su salida es la entrada para command2. stdout → stdin*.

  - cat prueba1.txt | wc -l
  - cat prueba1.txt | grep hola
  - cat /var/log/Xorg.0.log | more → para el caso de un archivo grande, igualmente no es necesario el cat.
  - cat /var/log/Xorg.0.log | less → idem anterior.
  - cat prueba1.txt | uniq | sort | wc -l
  - lspci | grep VGA

Encadenamiento de comandos

- Ejecutar todos los comandos sin importar el estado de salida del comando previo:
  - `cat prueba1.txt ; wc -l prueba1.txt ; echo “DONE”`
- Si un comando falla entonces los siguientes no se ejecutan.
  - `cat prueba1.txt && wc -l prueba1.txt && echo “DONE”`
- AND y OR
  - `cat prueba1.txt && wc -l ; echo “Done”`
- OR
  - `cat prueba1.txt ||  wc -l prueba1.txt ; echo “DONE`

Ayuda: Comando ‘man’

- Recordar siempre utilizar el comando man.

- Es quizás uno de los comandos más importantes para cualquier aprendiz (y a veces no tan aprendiz).

- El comando man sirve para desplegar en pantalla las páginas del manual, que proporcionan ayuda en línea acerca de cualquier comando, función de programación, archivo de configuración, etc.

- Investigar como es la estructura de las páginas de manual.

Ayuda: Comando ‘info’

- El comando info sirve cuando la ayuda de man es insuficiente, o directamente no existe una entrada en su base de datos sobre lo que estamos buscando.
- Tiene una interfaz navegable: muchas de las páginas tienen enlaces a otras páginas relacionadas, muy similar a las páginas web.
- Los enlaces se indican mediante un asterisco (*) seguido por el título del enlace, seguido por un doble dos puntos (::). Para seguir los enlaces hay que situar el cursor sobre ellos y pulsar la tecla <INTRO>.
- Para ayudar en la navegación por los documentos, también se usan las siguientes teclas:
  - b: para ir un paso hacia atrás, a la última página visitada
  - u: para ir al nivel superior en el árbol de páginas
  - n: para ir a la siguiente página
  - p: para ir a la página previa
  - Para salir del comando info hay que pulsar la tecla q.

Ayuda: comando ‘apropos’

- Se utiliza para buscar una cadena específica en todas la páginas del manual.
- Es a menudo un wrapper del comando man -k.
- Ejemplo: apropos mount.
  - En este ejemplo apropos retornará todas las página de manual que contengan el término “mount”.

Referencias

- Manual de Linux desde la consola de CentOS y Fedora.
- www.centos.org.
- www.fedoraproject.org.
- www.linuxfoundation.org.
- www.redhat.com.
- www.wikipedia.org.
- www.oreilly.com.
- http://www.pathname.com/fhs/
- http://itexpertscentral.com/operating-system/unix/unix-shell-script-structure.html
- http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/drivers/tty/Kconfig?id=HEAD
- http://en.wikipedia.org/wiki/List_of_Unix_programs
- http://en.wikipedia.org/wiki/Redirection_(Unix)
- http://en.wikipedia.org/wiki/Unix_pipe
- http://www.cyberciti.biz/faq/linux-unix-appleosx-bsd-cat-command-examples/
- http://1984.lsi.us.es/wiki-practicas-ssoo/index.php/P%C3%A1gina_Principal
- http://tomeko.net/other/linux/command_line_cheatsheet.php