

UNIVERSIDAD PRIVADA DE TACNA



FACULTAD DE INGENIERÍA
Escuela Profesional de Ingeniería de Sistemas

TITULO:

Informe de Trabajo Final 5
Sistema en CentOS 7

CURSO:

Sistemas Operativos II

DOCENTE:

Ing. Enrique Lanchipa Valencia

Integrantes:

Atahuachi Rivera, Gabriela Atahuachi

(2016055341)

Índice

1. Información General.	1
1.1. Objetivos:	1
1.2. Recursos Utilizados:	1
1.3. Conocimientos:	1
2. Marco Teórico.	2
2.1. Comandos en Centos 6.7.	2
2.2. Directorios en Centos 7.	2
2.3. Instalación de Servicios.	5
3. Procedimiento.	6
3.1. Instalar Paquetes del Repositorio.	6
3.1.1. Verificar Ejecución de Comandos (Tree y Finger).	6
3.1.2. Verificar si se Encuentran Instalados los Paquetes.	6
3.1.3. Ruta de los Paquetes del Repositorio Local en CentOS 6.7 y CentOS 7.	7
3.1.4. Buscar Paquetes en el Repositorio.	7
3.1.5. Instalar Paquetes.	8
3.1.6. Verificar si se Encuentran Instalados los Paquetes.	8
3.1.7. Probar Ejecución de Comandos Instalados.	9
3.2. Trabajando con Comandos del Sistemas.	10
3.2.1. Visualizar Tipo de Sistema de Archivos de Sistema.	10
3.2.2. Visualizar Particiones del Sistema.	10
3.2.3. Listar Recursos con el Comando Ls.	11
3.2.4. Listar Recursos con el Comando Ls -l.	11
3.2.5. Listar Inodos.	11
3.2.6. Listar Archivos Ocultos.	12
3.2.7. Listar Archivos Especiales.	12
3.2.8. Listar Archivos Tipo Caracter.	12
3.2.9. Buscar Archivos con Find.	13
3.2.10. Buscar Información que Coincida con Patrón Especificado con Comando Grep.	13
3.2.11. Visualizar el Usuario Actual Conectado con Who.	14
3.2.12. Ejecutar el Comando Logname.	14
3.2.13. Visualizar las Terminales Abiertas.	14
3.2.14. Visualizar Contenido de Archivos con Comando Cat.	15
3.2.15. Visualizar Contenido de Archivos con Comando Less.	15
3.2.16. Visualizar Contenido de Archivos con Comando More.	16
3.2.17. Utilizar la Ayuda del Sistema con Comando Help.	17
3.2.18. Utilizar la Ayuda con Comando Man.	18
4. Preguntas.	20
4.1. ¿5 Comandos para Gestionar la Memoria RAM y SWAP?.	20
4.2. ¿5 Comandos para Gestionar el Disco Duro?.	21
4.3. ¿5 Comandos para Gestionar los Procesos del Sistema?.	22
4.4. ¿5 Comandos para Gestionar el Sistema Operativo?.	22

4.5.	¿Diferencia entre CentOS 6.7 y 7 en la Ruta de Repositorio de Paquetes que se Encuantran en un CD o ISO?	22
4.5.1.	Arquitectura.	22
4.5.2.	Características de CentOS.	23
4.6.	¿Existen Otros Editores como en VIM, explique cuales?	23
4.7.	¿Cómo se Interpreta un Inodo? ¿Es Posible?	24
4.8.	¿Diferencia entre Cat, Less y More?	24
5.	Conclusiones.	25
6.	Recomendaciones.	26
6.1.	Principales.	26
6.2.	Secundarias.	26
7.	Bibliografía.	27

1. Información General.

1.1. Objetivos:

- Identificar los diferentes comandos sobre el Sistema en variedades en Linux
- Saber que hace cada comando
- Verificar su funcionalidad por medio de ejemplos e imágenes
- Tener en cuenta otras opciones para un solo comando
- Especificaciones detalladas del resultado de cada comando

1.2. Recursos Utilizados:

- Al menos 2 GB de RAM.
- Windows 10 64-bit: Pro, Enterprise o Education
- Espacio Disponible Mínima entre 50 a 100 GB
- Tener Instalado el Virtualizador VMware Workstation Pro
- Tener Instalado una Máquina Virtual CentOS versión 7 de Escritorio
- Terminal de Centos abierto
- Tener la Configuración Hyper-V activada en la Máquina Real para Iniciar o Encender la Máquina Virtual

1.3. Conocimientos:

- Conocimientos Básicos de Administracion de Sistemas Operativos en Windows.
- Conocimientos Basicos de Virtualización.

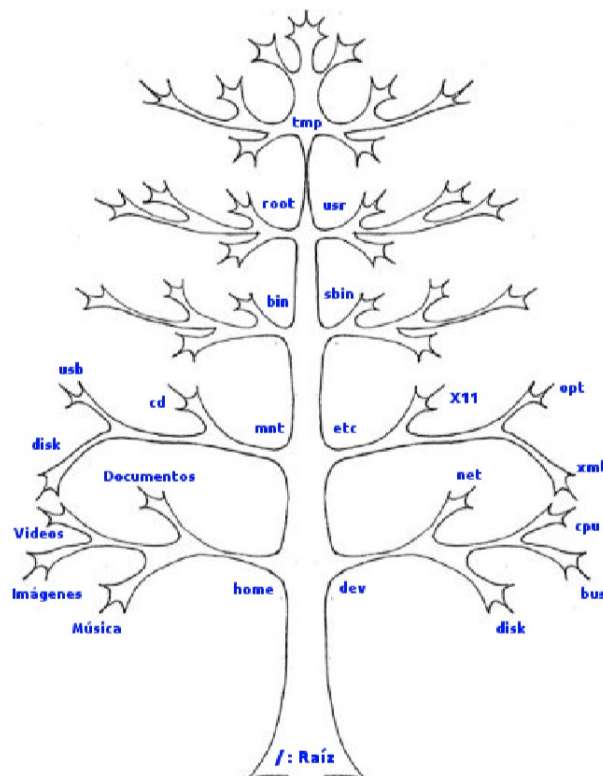
2. Marco Teórico.

2.1. Comandos en Centos 6.7.

- La versión 7 ya la coas es distinta; si bien la versión 7 ya incorpora comandos nuevos que no los tienen las versiones anteriores, en teoría la mayoría de estos no han cambiado.

2.2. Directorios en Centos 7.

- Con respecto a los directorios, estos son varios y cada uno tiene su propósito.



1. Primero esta el **Directorio /**, la cual es la Raíz.

Toda la estructura de directorios en los sistemas basados en UNIX parte de un directorio raíz también llamado directorio root y que se simboliza por una barra inclinada o /. De este directorio, es desde donde nacen todo el resto de directorios, independientemente que estén almacenados físicamente en discos o unidades separadas.

Cualquier dirección de archivo o carpeta en Linux empieza por el directorio raíz o /, seguido de todos los directorios y subdirectorios que lo contienen, separados cada uno de ellos por /.

2. Luego esta el **Directorio Home**.

La cual es el directorio de los usuarios estándar, y por lo tanto, el destinado a almacenar todos los archivos del usuario, como documentos, fotos, vídeos, música, plantillas, etc. También incluye archivos temporales de aplicaciones ejecutadas en modo usuario, que sirven para guardar las configuraciones de programas, etc.

Dentro de /home hay los directorios personales de todos los usuarios, nombrados según el nombre de usuario utilizado. Así por ejemplo, si en un sistema pongamos que hay dos usuarios denominados User1 y User2, la estructura será así:

- a) /home/User1
- b) /home/User2

Cada directorio de usuario contiene asimismo diferentes carpetas para ayudarlo a clasificar la información. Estas generalmente son: /Documentos, /Imágenes, /Música, /Plantillas y /Vídeos /, así como otros archivos y carpetas ocultas, que son las encargados de guardar la información de configuraciones de las aplicaciones del usuario.

En muchas distribuciones es una práctica recomendada el hecho de ubicar el directorio /home es una partición separada del resto, por tal de facilitar que, en caso de reinstalar el sistema operativo, puedas mantener intacta la partición de la /home, y de este modo mantener todos los archivos personales.

3. En paralelo con el Directorio Home, tenemos el **Directorio Dev**.

Este directorio incluye todos los dispositivos de almacenamiento, **en forma de archivos**, conectados al sistema, es decir, cualquier disco duro conectado, partición, memoria USB, o CDROM conectado al sistema y que el sistema pueda entender como un volumen lógico de almacenamiento.

Siendo esto así, verás que la ruta en la que se encuentra cualquier volumen (partición o dispositivo externo) conectado al sistema siempre empieza por **/dev**.

Este es el directorio que contiene, por decirlo de algún modo, la información de cada uno de los volúmenes, a diferencia del directorio **/media**, que verás más adelante, que lo que contiene son solo los puntos de montaje, pero no la información real de estos volúmenes.

Para ver esto en la práctica, si abres una ventana de terminal y ejecutas el comando **sudo fdisk -l**, verás la estructura de particiones de tu sistema. En una instalación típica de cualquier distro GNU/Linux suele ser la siguiente:

- a) /dev/sda1 - Partición principal
- b) /dev/sda2 - Partición extendida
- c) /dev/sda5 - Partición Swap

La partición sda1 suele ser la partición principal, Obviamente si has editado manualmente el esquema de particiones, en tu caso será diferente, esto es solo un ejemplo típico

para ayudar a explicar la función del directorio **/dev**.

Eso en cuanto a particiones. Si se trata de un dispositivo externo, el volumen estará igualmente dentro de **/dev**, pero en este caso varía el nombre que el sistema le asigna a dicho volumen. Generalmente la estructura suele ser la siguiente (si ejecutas nuevamente el comando `sudo fdisk -l` con un dispositivo externo conectado lo podrás comprobar tu mismo).

- a) `/dev/sdb1`
- b) `/dev/sdb2`
- c) `/dev/sdb3`
- d) ...

4. Tenemos el **Directorio Etc**. Es el encargado de almacenar los archivos de configuración tanto a nivel de componentes del sistema operativo en sí, como de los programas y aplicaciones instaladas a posteriores.

Es un directorio que debería contener únicamente ficheros de configuración, y no debería contener binarios.

5. Luego está el **Directorio Root**.

No hay que confundirlo con `/`, una cosa es el directorio raíz o root y otra muy diferente `/root`. En este caso, se puede asemejar a un `/home` pero exclusivo para el usuario root o privilegiado.

Vendría a ser como el directorio `/home` del usuario root o superusuario del sistema.

A diferencia de los otros usuarios, que se encuentran todos dentro de `/home` en sus respectivas subcarpetas, el directorio del usuario root está en su propia carpeta colgando directamente de la raíz del sistema.

6. La que está por encima de todos es el **Directorio Tmp**.

Como ya da a entender su nombre, sirve para almacenar archivos temporales de todo tipo, ya sea de elementos del sistema, o también de diferentes aplicaciones a nivel de usuario como puedan ser Firefox o Chrome/Chromium.

Es un directorio dispuesto para almacenar contenido de corta duración, de hecho en la gran mayoría de los casos se suele vaciar de forma automática en cada reinicio del sistema. Aun así, no debes borrar su contenido de forma manual, puesto que puede contener archivos necesarios para ciertos programas o procesos que estén ejecutándose.

Las aplicaciones programadas para almacenar archivos en este directorio deben asumir que solo serán recuperables en la sesión actual. En este sentido, hay otro subdirectorio, **/var/tmp**, dispuesto igualmente para el almacenamiento de archivos temporales, pero cuyo contenido no se borra de forma automática tras el reinicio del sistema.

2.3. Instalación de Servicios.

- Hay diferentes formas de Instalar Servicios, como son: SAMBA, MariaDB, MySQL, DNS, FTP, entre otros. La manera más fácil de hacerlo es Descargarlo e Instalarlo desde Internet con el comando **YUM**, siempre y cuando tengas Internet en la Máquina Virtual y la Conexión a Red esté en NAT y pueda compartir internet desde la máquina real.
- Hay otra manera, la cual es sólo instalarlo desde la misma imagen ISO de la versión misma.

3. Procedimiento.

3.1. Instalar Paquetes del Repositorio.

3.1.1. Verificar Ejecución de Comandos (Tree y Finger).

- Para el Comando **tree**.

```
[root@localhost ~]# tree ←  
bash: tree: no se encontró la orden  
[root@localhost ~]#
```

Nota: Por lo visto no está Reconociendo el Comando.

- Para el Comando **finger**.

```
[root@localhost ~]# finger ←  
bash: finger: no se encontró la orden  
[root@localhost ~]#
```

Nota: Por lo visto no está Reconociendo el Comando.

3.1.2. Verificar si se Encuentran Instalados los Paquetes.

- Para el Comando **tree**.

```
[root@localhost ~]# rpm -q tree ←  
el paquete tree no está instalado  
[root@localhost ~]# _
```

Nota: Por lo visto no está Instalado el Comando.

- Para el Comando **finger**.

```
[root@localhost Packages]# rpm -q finger ←  
el paquete finger no está instalado  
[root@localhost Packages]#
```

Nota: Por lo visto no está Instalado el Comando.

3.1.3. Ruta de los Paquetes del Repositorio Local en CentOS 6.7 y CentOS 7.

- En CentOS 6.7 se encuentra en el Directorio: `/etc/media/CentOS 6.7 Final/Packages`.
- En CentOS 7 se encuantra en el Directorio: `/run/media/grivera/CentOS 7 x86 64`.

3.1.4. Buscar Paquetes en el Repositorio.

- Antes de Empezar a Buscar los Paquetes, primero debemos estar dentro del Repositorio del Disco. Para ello, digitaremos los siguientes comandos.

```
[root@localhost grivera]# cd /run ←  
[root@localhost run]# cd media ←  
[root@localhost media]# cd grivera ← ***  
[root@localhost grivera]# cd CentOS\ 7\ x86_64/ ←  
[root@localhost CentOS 7 x86_64]# cd Packages/ ←  
[root@localhost Packages]#
```

Nota: En la parte Marcada (***) debemos aclarar que este será el Nombre de Usuario con el que Creaste en la Instalación o con el que te hayas Logeado.

- Ahora Buscaremos el Paquete para el Comando **tree**.

```
[root@localhost Packages]# find tree* ←  
tree-1.6.0-10.el7.x86_64.rpm  
[root@localhost Packages]#
```

- Luego Buscaremos el Paquete para el Comando **finger**.

```
[root@localhost Packages]# find finger* ←
finger-0.17-52.el7.x86_64.rpm
finger-server-0.17-52.el7.x86_64.rpm
[root@localhost Packages]#
```

Nota: Aquí tenemos dos paquetes, instalaremos el primero, ya que es el básico.

3.1.5. Instalar Paquetes.

- Para el Paquete tree.

```
[root@localhost Packages]# rpm -ivh tree-1.6.0-10.el7.x86_64.rpm ←
advertencia:tree-1.6.0-10.el7.x86_64.rpm: EncabezadoV3 RSA/SHA256 Signature, ID de clave f4a80eb5: NOKEY
Preparando... ##### [100%]
Actualizando / instalando...
  1:tree-1.6.0-10.el7 ##### [100%]
[root@localhost Packages]#
```

- Para el Paquete finger.

```
[root@localhost Packages]# rpm -ivh finger-0.17-52.el7.x86_64.rpm ←
advertencia:finger-0.17-52.el7.x86_64.rpm: EncabezadoV3 RSA/SHA256 Signature, ID de clave f4a80eb5: NOKEY
Preparando... ##### [100%]
    el paquete finger-0.17-52.el7.x86_64 ya está instalado
[root@localhost Packages]#
```

3.1.6. Verificar si se Encuentran Instalados los Paquetes.

- Para el Paquete tree.

```
[root@localhost Packages]# rpm -q tree ←  
tree-1.6.0-10.el7.x86_64  
[root@localhost Packages]#
```

- Para el Paquete finger.

```
[root@localhost Packages]# rpm -q finger ←  
finger-0.17-52.el7.x86_64  
[root@localhost Packages]#
```

3.1.7. Probar Ejecución de Comandos Instalados.

- Para el Paquete tree.

```
[root@localhost ~]# tree ←  
.  
├── anaconda-ks.cfg  
└── initial-setup-ks.cfg  
  
0 directories, 2 files  
[root@localhost ~]#
```

- Para el Paquete finger.

```
[root@localhost ~]# finger
```

Login	Name	Tty	Idle	Login Time	Office	Office Phone	Host
grivera	Gabriela Atahuachi Riv	*:0		Jan 23 22:55			(:0)
grivera	Gabriela Atahuachi Riv	pts/0		Jan 23 22:55			(:0)

```
[root@localhost ~]#
```

3.2. Trabajando con Comandos del Sistemas.

3.2.1. Visualizar Tipo de Sistema de Archivos de Sistema.

- DF (disk filesystem – Sistema de archivos de disco) nos proporciona información detallada del espacio usado en el disco. Si deseamos que dentro de la información desplegada este el tipo de sistema de archivos usaremos el parámetro `-T` y para que sea legible añadiremos `h`.

```
[root@localhost ~]# df -Th
```

S.ficheros	Tipo	Tamaño	Usados	Disp	Uso%	Montado en
/dev/sda1	ext3	34G	3,7G	29G	12%	/
devtmpfs	devtmpfs	895M	0	895M	0%	/dev
tmpfs	tmpfs	910M	0	910M	0%	/dev/shm
tmpfs	tmpfs	910M	11M	900M	2%	/run
tmpfs	tmpfs	910M	0	910M	0%	/sys/fs/cgroup
/dev/sda2	ext3	20G	50M	19G	1%	/home
tmpfs	tmpfs	182M	20K	182M	1%	/run/user/1000
/dev/sr0	iso9660	4,3G	4,3G	0	100%	/run/media/grivera/CentOS 7 x86_64

```
[root@localhost ~]#
```

- Otra Forma.

```
[root@localhost ~]# df -h
```

S.ficheros	Tamaño	Usados	Disp	Uso%	Montado en
/dev/sda1	34G	3,7G	29G	12%	/
devtmpfs	895M	0	895M	0%	/dev
tmpfs	910M	0	910M	0%	/dev/shm
tmpfs	910M	11M	900M	2%	/run
tmpfs	910M	0	910M	0%	/sys/fs/cgroup
/dev/sda2	20G	50M	19G	1%	/home
tmpfs	182M	20K	182M	1%	/run/user/1000
/dev/sr0	4,3G	4,3G	0	100%	/run/media/grivera/CentOS 7 x86_64

```
[root@localhost ~]#
```

3.2.2. Visualizar Particiones del Sistema.

```
[root@localhost ~]# fdisk -l ←
```

Disk /dev/sda: 64.4 GB, 64424509440 bytes, 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Identificador del disco: 0x000e308f

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sda1	*	2048	72581119	36289536	83	Linux
/dev/sda2		72581120	113541119	20480000	83	Linux
/dev/sda3		113541120	125829119	6144000	82	Linux swap / Solaris

```
[root@localhost ~]#
```

3.2.3. Listar Recursos con el Comando Ls.

- Como se muestra en la Imagen. La cual sólo Muestra los Nombres de los Archivos.

```
[root@localhost ~]# ls ←
anaconda-ks.cfg  initial-setup-ks.cfg
[root@localhost ~]#
```

3.2.4. Listar Recursos con el Comando Ls -l.

- Como se muestra en la Imagen. La cual Muestra Información Detallada de los Archivos.

```
[root@localhost ~]# ls -l ←
total 8
-rw-----. 1 root root 1695 ene  8 16:39 anaconda-ks.cfg
-rw-r--r--. 1 root root 1912 ene  8 17:35 initial-setup-ks.cfg
[root@localhost ~]#
```

3.2.5. Listar Inodos.

- Una de las formas habituales es utilizar el popular comando 'ls', que como sabéis sirve para listar el contenido de una carpeta. Para ver los números de inodo debemos añadir el parámetro '-li'. El número de inodo del archivo se mostrará en el primer campo de salida.

```
[root@localhost ~]# ls -li ←
total 8
958471 -rw-----. 1 root root 1695 ene  8 16:39 anaconda-ks.cfg
958481 -rw-r--r--. 1 root root 1912 ene  8 17:35 initial-setup-ks.cfg
[root@localhost ~]#
```

3.2.6. Listar Archivos Ocultos.

```
[root@localhost ~]# ls -a ←
.          .bash_history .bashrc  .cshrc          .tcshrc      .xauthT68bnr .xauthZFvVSn
..         .bash_logout  .cache   .dbus           .viminfo     .xauthXc2JEt
anaconda-ks.cfg .bash_profile .config  initial-setup-ks.cfg .xauthd4aIDN .xauthXzIIvG
[root@localhost ~]#
```

3.2.7. Listar Archivos Especiales.

```
[root@localhost ~]# ls -lia /dev/sda* ←
18137 brw-rw----. 1 root disk 8, 0 ene 23 22:52 /dev/sda
18172 brw-rw----. 1 root disk 8, 1 ene 23 22:52 /dev/sda1
18173 brw-rw----. 1 root disk 8, 2 ene 23 22:52 /dev/sda2
18174 brw-rw----. 1 root disk 8, 3 ene 23 22:52 /dev/sda3
[root@localhost ~]#
```

3.2.8. Listar Archivos Tipo Caracter.


```
[root@localhost ~]# ls -lia /dev/tty* ←
6485 crw-rw-rw-. 1 root tty      5,  0 ene 23 22:52 /dev/tty
6487 crw--w----. 1 root tty      4,  0 ene 23 22:52 /dev/tty0
6492 crw--w----. 1 root tty      4,  1 ene 23 22:53 /dev/tty1
6501 crw--w----. 1 root tty      4, 10 ene 23 22:52 /dev/tty10
6502 crw--w----. 1 root tty      4, 11 ene 23 22:52 /dev/tty11
6503 crw--w----. 1 root tty      4, 12 ene 23 22:52 /dev/tty12
6504 crw--w----. 1 root tty      4, 13 ene 23 22:52 /dev/tty13
6505 crw--w----. 1 root tty      4, 14 ene 23 22:52 /dev/tty14
6506 crw--w----. 1 root tty      4, 15 ene 23 22:52 /dev/tty15
6507 crw--w----. 1 root tty      4, 16 ene 23 22:52 /dev/tty16
6508 crw--w----. 1 root tty      4, 17 ene 23 22:52 /dev/tty17
6509 crw--w----. 1 root tty      4, 18 ene 23 22:52 /dev/tty18
6510 crw--w----. 1 root tty      4, 19 ene 23 22:52 /dev/tty19
6493 crw--w----. 1 root tty      4,  2 ene 23 22:52 /dev/tty2
6511 crw--w----. 1 root tty      4, 20 ene 23 22:52 /dev/tty20
6512 crw--w----. 1 root tty      4, 21 ene 23 22:52 /dev/tty21
6513 crw--w----. 1 root tty      4, 22 ene 23 22:52 /dev/tty22
6514 crw--w----. 1 root tty      4, 23 ene 23 22:52 /dev/tty23
6515 crw--w----. 1 root tty      4, 24 ene 23 22:52 /dev/tty24
6516 crw--w----. 1 root tty      4, 25 ene 23 22:52 /dev/tty25
6517 crw--w----. 1 root tty      4, 26 ene 23 22:52 /dev/tty26
6518 crw--w----. 1 root tty      4, 27 ene 23 22:52 /dev/tty27
6519 crw--w----. 1 root tty      4, 28 ene 23 22:52 /dev/tty28
6520 crw--w----. 1 root tty      4, 29 ene 23 22:52 /dev/tty29
6494 crw--w----. 1 root tty      4,  3 ene 23 22:52 /dev/tty3
6521 crw--w----. 1 root tty      4, 30 ene 23 22:52 /dev/tty30
6522 crw--w----. 1 root tty      4, 31 ene 23 22:52 /dev/tty31
6523 crw--w----. 1 root tty      4, 32 ene 23 22:52 /dev/tty32
6524 crw--w----. 1 root tty      4, 33 ene 23 22:52 /dev/tty33
6525 crw--w----. 1 root tty      4, 34 ene 23 22:52 /dev/tty34
6526 crw--w----. 1 root tty      4, 35 ene 23 22:52 /dev/tty35
6527 crw--w----. 1 root tty      4, 36 ene 23 22:52 /dev/tty36
6528 crw--w----. 1 root tty      4, 37 ene 23 22:52 /dev/tty37
6529 crw--w----. 1 root tty      4, 38 ene 23 22:52 /dev/tty38
6530 crw--w----. 1 root tty      4, 39 ene 23 22:52 /dev/tty39
6495 crw--w----. 1 root tty      4,  4 ene 23 22:52 /dev/tty4
6531 crw--w----. 1 root tty      4, 40 ene 23 22:52 /dev/tty40
6532 crw--w----. 1 root tty      4, 41 ene 23 22:52 /dev/tty41
6533 crw--w----. 1 root tty      4, 42 ene 23 22:52 /dev/tty42
6534 crw--w----. 1 root tty      4, 43 ene 23 22:52 /dev/tty43
6535 crw--w----. 1 root tty      4, 44 ene 23 22:52 /dev/tty44
```

3.2.9. Buscar Archivos con Find.

```
[root@localhost Packages]# rpm -q finger ←
finger-0.17-52.el7.x86_64
[root@localhost Packages]#
```

3.2.10. Buscar Información que Coincida con Patrón Especificado con Comando Grep.

- Busca el Usuario y lo Imprime, sin necesidad de Abrir el Fichero passwd.


```
[root@localhost dporlles]# grep dporlles /etc/passwd ←  
dporlles:x:1001:1001:~/home/dporlles:/bin/bash  
[root@localhost dporlles]#
```

3.2.11. Visualizar el Usuario Actual Conectado con Who.

```
[root@localhost ~]# who ←  
grivera :0 2020-01-23 22:55 (:0)  
grivera pts/0 2020-01-23 22:55 (:0)  
[root@localhost ~]#
```

3.2.12. Ejecutar el Comando Logname.

```
[root@localhost ~]# logname ←  
grivera  
[root@localhost ~]#
```

3.2.13. Visualizar las Terminales Abiertas.

```
[root@localhost ~]# who ←
grivera  :0          2020-01-23 22:55 (:0)
grivera  pts/0       2020-01-23 22:55 (:0)
[root@localhost ~]#
```

3.2.14. Visualizar Contenido de Archivos con Comando Cat.

- Muestra el contenido de un archivo en la pantalla sin limites.

```
[root@localhost ~]# cat /proc/filesystems ←
nodev    sysfs
nodev    rootfs
nodev    ramfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cpuset
nodev    tmpfs
nodev    devtmpfs
nodev    debugfs
nodev    securityfs
nodev    sockfs
nodev    dax
nodev    bpf
nodev    pipefs
nodev    anon_inodefs
nodev    configfs
nodev    devpts
nodev    hugetlbfs
nodev    autofs
nodev    pstore
nodev    mqueue
nodev    selinuxfs
nodev    ext3
nodev    ext2
nodev    ext4
nodev    rpc_pipefs
nodev    fuseblk
nodev    fuse
nodev    fusectl
nodev    iso9660
[root@localhost ~]#
```

3.2.15. Visualizar Contenido de Archivos con Comando Less.

- Muestra el contenido del archivo con una pantalla de desplazamiento de manera que puedas manejar entre paginas utilizando PgUp, PgDn, Home, y End.

```
nodev    sysfs
nodev    rootfs
nodev    ramfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cpuset
nodev    tmpfs
nodev    devtmpfs
nodev    debugfs
nodev    securityfs
nodev    sockfs
nodev    dax
nodev    bpf
nodev    pipefs
nodev    anon_inodefs
nodev    configfs
nodev    devpts
nodev    hugetlbfs
nodev    autofs
nodev    pstore
nodev    mqueue
nodev    selinuxfs
          ext3
          ext2
          ext4
nodev    rpc_pipefs
          fuseblk
nodev    fuse
nodev    fusectl
          iso9660
/proc/filesystems (END)
```

3.2.16. Visualizar Contenido de Archivos con Comando More.

- Te permite mostrar el resultado de la ejecución de un comando en la terminal de a una página a la vez. Esto es especialmente útil cuando se ejecuta un comando que causa un gran desplazamiento, como el comando `ls` o el comando `du`.

```
[root@localhost ~]# more /proc/filesystems ←
nodev    sysfs
nodev    rootfs
nodev    ramfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cpuset
nodev    tmpfs
nodev    devtmpfs
nodev    debugfs
nodev    securityfs
nodev    sockfs
nodev    dax
nodev    bpf
nodev    pipefs
nodev    anon_inodefs
nodev    configfs
nodev    devpts
nodev    hugetlbfs
nodev    autofs
nodev    pstore
nodev    mqueue
nodev    selinuxfs
        ext3
        ext2
        ext4
nodev    rpc_pipefs
        fuseblk
nodev    fuse
nodev    fusectl
        iso9660
[root@localhost ~]#
```

3.2.17. Utilizar la Ayuda del Sistema con Comando Help.

```
[root@localhost ~]# help
GNU bash, versión 4.2.46(2)-release (x86_64-redhat-linux-gnu)
Estas órdenes del shell están definidas internamente. Teclee 'help' para
ver esta lista.
Teclee 'help nombre' para saber más sobre la función 'nombre'.
Use 'info bash' para saber más sobre el shell en general.
Use 'man -k' o 'info' para saber más sobre las órdenes que no están en
esta lista.
```

Un asterisco (*) junto a un nombre significa que el comando está desactivado.

id_trabajo [&]	history [-c] [-d despl] [n] ó history -anrw [fichero>
((expresión))	if ÓRDENES; then ÓRDENES; [elif ÓRDENES; then ÓR>
. fichero [argumentos]	jobs [-lnprs] [idtrabajo ...] ó jobs -x orden [args]>
:	kill [-s id_señal -n num_señal -id_señal] pid >
[arg...]	let arg [arg ...]
[[expresión]]	local [opción] nombre[=valor] ...
alias [-p] [nombre[=valor] ...]	logout [n]
bg [id_trabajo ...]	mapfile [-n cuenta] [-O origen] [-s cuenta] [-t] [-u >
bind [-lpvsPVS] [-m comb_teclas] [-f fichero] [-q nomb>	popd [-n] [+N -N]
break [n]	printf [-v var] formato [argumentos]
builtin [orden-interna-shell [arg ...]]	pushd [-n] [+N -N dir
caller [expresión]	pwd [-LP]
case PALABRA in [PATRÓN [PATRÓN]...) ÓRDENES ;;]>	read [-ers] [-a matriz] [-d delim] [-i texto] [-n nca>
cd [-L [-P [-e]]] [dir]	readarray [-n cuenta] [-O origen] [-s cuenta] [-t] [->
command [-pVv] orden [arg ...]	readonly [-aAf] [name[=value] ...] or readonly -p
compgen [-abcdfgjksub] [-o opción] [-A acción] [-G>	return [n]
complete [-abcdfgjksub] [-pr] [-DE] [-o opción] [-A >	select NOMBRE [in PALABRAS ... ;] do ÓRDENES; done
comptopt [-o +o opción] [-DE] [nombre ...]	set [-abefhkmnptuvxBCHP] [-o option-name] [--] [arg .>
continue [n]	shift [n]
coproc [NOMBRE] orden [redirecciones]	shopt [-pqsu] [-o] [nombre_opción...]
declare [-aAfFgIlrtux] [-p] [name[=value] ...]	source fichero [argumentos]
dirs [-clpv] [+N] [-N]	suspend [-f]
disown [-h] [-ar] [idtrabajo ...]	test [expresión]
echo [-neE] [arg ...]	time [-p] tubería
enable [-a] [-dnps] [-f fichero] [nombre ...]	times
eval [arg ...]	trap [-lp] [[arg] id_señal ...]
exec [-cl] [-a nombre] [orden [argumentos ...]] [redir>	true
exit [n]	type [-afptP] nombre [nombre ...]
export [-fn] [nombre[=valor] ...] ó export -p	typeset [-aAfFgIlrtux] [-p] name[=value] ...
false	ulimit [-SHacdefilmnpqrstuvx] [límite]
fc [-e nombre_e] [-lnr] [primero] [último] ó fc -s [>	umask [-p] [-S] [modo]

3.2.18. Utilizar la Ayuda con Comando Man.

- Ejecutaremos el comando `man ls` para obtener ayuda del comando `ls`.

```
LS(1)                                     General Commands Manual                                     LS(1)

NOMBRE
ls, dir, vdir - listan los contenidos de directorios

SINOPSIS
ls [opciones] [fichero...]
dir [fichero...]
vdir [fichero...]

Opciones de POSIX: [-CFRacdilqrtul]

Opciones de GNU (en la forma más corta): [-labcdfghiklmnopqrstuvwxABCDGHLNQRSUX] [-w cols] [-T cols] [-I patrón] [--full-time] [--show-control-chars] [--block-size=tamaño] [--format={long,verbose,commas,across,vertical,single-column}] [--sort={none,time,size,extension}] [--time={atime,access,use,ctime,status}] [--color={none,auto,always}] [--help] [--version] [--]

DESCRIPCIÓN
El programa ls lista primero sus argumentos no directorios fichero, y luego para cada argumento directorio todos los ficheros susceptibles de listarse contenidos en dicho directorio. Si no hay presente ningún argumento aparte de las opciones, se supone un argumento predeterminado .' (el directorio de trabajo). La opción -d hace que los directorios se traten como argumentos no directorios; es decir, como ficheros normales. Un fichero es susceptible de listarse cuando su nombre no comienza con .' o cuando se da la opción -a (o -A, vea más abajo).

Cada una de las listas de ficheros (las de ficheros no directorios, y para cada directorio la lista de ficheros de dentro) se clasifica separadamente según la secuencia de clasificación en la localización en curso. Cuando se da la opción -l, cada lista se precede por una línea de sumario que da el tamaño total de todos los ficheros en la lista, medidos en semi-kilobytes (512 B).

La salida es a la salida estándar, una entrada por cada línea, a menos que se pida una salida multi-columna mediante la opción -C. Sin embargo, para la salida en una terminal, no está definido si la salida será en una sola columna o en varias. Las opciones -l y -C se pueden emplear para forzar la salida en una o en varias columnas, respectivamente.

OPCIONES DE POSIX
-C      Lista los ficheros en columnas, ordenados verticalmente.

-F      Añade tras cada nombre de directorio un /', tras cada nombre de FIFO un |', y tras cada nombre de un ejecutable un *'.

-R      Lista recursivamente los subdirectorios encontrados.

-a      Incluye en el listado ficheros cuyos nombres empiecen por .'.

-c      Emplea el tiempo de cambio de estado en vez del de modificación para la clasificación (con -t) o el listado (con -l).
```

Manual page ls(1) line 1 (press h for help or q to quit)

4. Preguntas.

4.1. ¿5 Comandos para Gestionar la Memoria RAM y SWAP?

- Una forma es con el Comando free; la cual muestra la Cantidad total de memoria física libre y usada, Cantidad total de memoria de intercambio en el Sistema, Buffers y cachés usados por el kernel.

```
[root@localhost Packages]# free ←
              total        used        free      shared    buffers     cached
Mem:          1907936      704368      1203568         3544        88628       259548
-/+ buffers/cache:      356192      1551744
Swap:          6143996           0       6143996
[root@localhost Packages]#
```

- Otra forma es con el comando top; la cual proporciona una vista dinámica y en tiempo real de un sistema en ejecución. Incluido en ese resumen del sistema es la capacidad de controlar el uso de memoria por proceso.

```
[grivera@localhost ~]$ top ←
top - 00:24:11 up 37 min, 2 users, load average: 0.02, 0.01, 0.00
Tasks: 148 total, 1 running, 143 sleeping, 4 stopped, 0 zombie
Cpu(s): 2.0%us, 0.0%sy, 0.0%ni, 98.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1907936k total, 520664k used, 1387272k free, 29680k buffers
Swap: 6143996k total, 0k used, 6143996k free, 179332k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 2223 root        20   0  281m  36m 8252 S  1.3  1.9   0:07.39 Xorg
 2504 grivera    20   0  479m  14m 10m S  0.7  0.8   0:00.52 gnome-panel
 2774 grivera   20   0  301m  13m 9.9m S  0.3  0.7   0:01.12 gnome-terminal
 2960 grivera   20   0 15032 1248  924 R  0.3  0.1   0:00.20 top
    1 root        20   0 19356 1540 1228 S  0.0  0.1   0:01.43 init
    2 root        20   0   0     0   0 S  0.0  0.0   0:00.00 kthreadd
    3 root        RT   0   0     0   0 S  0.0  0.0   0:00.00 migration/0
    4 root        20   0   0     0   0 S  0.0  0.0   0:00.00 ksoftirqd/0
    5 root        RT   0   0     0   0 S  0.0  0.0   0:00.00 stopper/0
    6 root        RT   0   0     0   0 S  0.0  0.0   0:00.00 watchdog/0
    7 root        20   0   0     0   0 S  0.0  0.0   0:01.50 events/0
    8 root        20   0   0     0   0 S  0.0  0.0   0:00.00 events/0
    9 root        20   0   0     0   0 S  0.0  0.0   0:00.00 events_long/0
   10 root        20   0   0     0   0 S  0.0  0.0   0:00.00 events_power_of
   11 root        20   0   0     0   0 S  0.0  0.0   0:00.00 cgroup
   12 root        20   0   0     0   0 S  0.0  0.0   0:00.00 khelper
   13 root        20   0   0     0   0 S  0.0  0.0   0:00.00 netns
   14 root        20   0   0     0   0 S  0.0  0.0   0:00.00 async/mgr
   15 root        20   0   0     0   0 S  0.0  0.0   0:00.00 pm
   16 root        20   0   0     0   0 S  0.0  0.0   0:00.00 sync_supers
   17 root        20   0   0     0   0 S  0.0  0.0   0:00.00 bdi-default
   18 root        20   0   0     0   0 S  0.0  0.0   0:00.00 kintegrityd/0
   19 root        20   0   0     0   0 S  0.0  0.0   0:00.14 kblockd/0
   20 root        20   0   0     0   0 S  0.0  0.0   0:00.00 kacpid
   21 root        20   0   0     0   0 S  0.0  0.0   0:00.00 kacpi_notify
   22 root        20   0   0     0   0 S  0.0  0.0   0:00.00 kacpi_hotplug
   23 root        20   0   0     0   0 S  0.0  0.0   0:00.00 ata_aux
   24 root        20   0   0     0   0 S  0.0  0.0   0:00.46 ata_sff/0
   25 root        20   0   0     0   0 S  0.0  0.0   0:00.00 kxspend_usbd
   26 root        20   0   0     0   0 S  0.0  0.0   0:00.01 khubd
```

- Esto reportará el uso de memoria en MB.

```
[grivera@localhost ~]$ free -m ←
              total        used        free      shared    buffers     cached
Mem:           1863          507         1355           3         28         175
-/+ buffers/cache:          303         1559
Swap:           5999           0         5999
[grivera@localhost ~]$
```

- Si tu sistema es moderno, querrás usar la opción -g (gigabytes).

```
[grivera@localhost ~]$ free -g
              total        used        free      shared    buffers     cached
Mem:           1           0           1           0           0           0
-/+ buffers/cache:           0           1
Swap:          5           0           5
```

- Informa las estadísticas de la memoria virtual

```
[grivera@localhost ~]$ vmstat
procs -----memory----- --swap--  -----io----- --system--  -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs us sy id wa st
 0  0       0 1388048 29684 179412    0    0   73    6   34   81  0  1 97  2  0
```

4.2. ¿5 Comandos para Gestionar el Disco Duro?.

- `fdisk -l`: Para listar todas las particiones existentes en nuestro sistema pasaremos el argumento “-l”, que hará que se listen ordenadas por el nombre del dispositivo.
- `fdisk -l /dev/sdb`: Para ver todas las particiones de un único disco, al comando anterior añadiremos el nombre de dicho disco, expresado de la forma en la que se mostraba en la captura anterior.
- `fdisk /dev/sdb (m)`: Si queremos ver todas las opciones que nos ofrece `fdisk`, únicamente deberemos pasar el comando seguido de una unidad de disco (para darle algo sobre lo que trabajar).
Ya en el menú de `fdisk`, pulsaremos ‘m’ para entrar en la ayuda y ver todas las opciones que podríamos aplicar al disco seleccionado.
- `fdisk /dev/sdb (p)`: Para mostrar al completo la lista de particiones de nuestro sistema, al igual que antes deberemos pasar una unidad con la que conseguir acceso al menú de `fdisk`, tras lo cual, una vez en el mismo, pulsaremos en esta ocasión ‘p’, para obtener el listado que buscamos.
- `fdisk /dev/sdb (d)`: Si lo que queremos es borrar una partición en concreto (por ejemplo, `/dev/sdb2`), primeramente deberemos seleccionar el disco en el que podemos encontrarla, que siguiendo el caso del ejemplo sería `/dev/sdb`.
Ya en el menú de `fdisk` pulsamos ‘d’, letra correspondiente a la opción de ‘delete’ o borrar de esta herramienta, tras lo cual se nos pedirá que introduzcamos el número que se le ha dado a la partición que nos interesa. Recomiendo siempre antes de realizar ninguna operación, hacer un listado completo para tener claro qué es lo que tenemos, trabajar de memoria puede jugarnos malas pasadas.
Una vez tenemos el número de partición que queremos eliminar claro, lo introducimos y lo siguiente que deberemos hacer es pulsar ‘w’ (write) para escribir los cambios o confirmar la acción. Los cambios se aplicarán en el siguiente inicio del sistema.

4.3. ¿5 Comandos para Gestionar los Procesos del Sistema?.

- Top: El comando top es un clásico para mostrar los recursos usados por tu sistema y así poder ver cual de ellos consume más.
- Htop: Es un 'top' mejorado, no suele venir instalado en las distribuciones Linux. Sobre todo es más fácil de interpretar que su hermano top y además puedes realizar operaciones como parar un proceso usando las teclas (la imagen que encabeza este post es una captura de htop).
- Ps: El comando más usado (y universal) para manipular procesos en Linux. Muestra todos los procesos y además puedes realizar operaciones con ellos.
- Renice: Este comando cambia el valor 'nice' de un proceso en ejecución. El valor 'nice' determina la prioridad del proceso. Un valor de -19 es de muy alta prioridad, mientras que un valor 19 es de baja prioridad. Por defecto el valor es el 0.
- Atop: Monitor de procesos en ASCII en tiempo real donde muestra CPU, memoria, capas de red, usuario, prioridad, etc para cada proceso activo. Fantástico, podéis echar un vistazo a la web oficial. Gracias Yago.

4.4. ¿5 Comandos para Gestionar el Sistema Operativo?.

- sudo shutdown -h now: Apagar la Máquina.
- shutdown -r now o reboot: Reiniciar la Máquina.
- su username ('Contraseña'): Sesiones.
- tty: es la Terminal.
- uname -m: Mostrar Arquitectura de la Máquina.

4.5. ¿Diferencia entre CentOS 6.7 y 7 en la Ruta de Repositorio de Paquetes que se Encuentran en un CD o ISO?.

4.5.1. Arquitectura.

Arquitectura	CentOS 4	CentOS 5	CentOS 6	CentOS 7
Sistemas i386 (AMD, Via, Pentium de 32 bits)	Sí	Sí	Sí	Sí
Sistemas x86_64 (AMD64/EM64T)	Sí	Sí	Sí	Sí
Sistemas ia64 (Itanium2)	Sí	No	No	No
Sistemas s390/s390x (IBM zSeries & IBM S/390)	Sí	No	No	No
Sistemas ppc/ppc64 (IBM Power, Mac)	Sí	No	No	No
Sistemas SPARC (Procesadores Sun SPARC)	Sí	No	No	No
Sistemas Alpha (Procesadores DEC Alpha)	Sí	No	No	No
Sistemas armhfp (arm32)	No	No	No	Sí
Sistemas aarch64 (arm64)	No	No	No	Sí

4.5.2. Características de CentOS.

La actual versión 7.0 de CentOS (enero 2017) se basa en el núcleo de Linux 3.10.0, incluyendo la extensión de seguridad mencionada anteriormente SELinux, y ha implementado GCC (GNU Compiler Collection). Esta colección contiene el compilador para los lenguajes de programación más importantes, como por ejemplo C, C++ y Java. Esta distribución de Linux también es compatible con Hyperthreading (la división de un procesador en dos procesadores virtuales para aumentar el rendimiento), Plug and Play, Bluetooth y la sexta versión del protocolo de Internet (IPv6). Para las versiones anteriores de CentOS 5 y 6 existen bibliotecas de compatibilidad. El paquete estándar de la distribución también incluye los siguientes componentes de software:

- **Servidor web:** httpd 2.4.6 (Apache)
- **LAN Manager:** Samba-4.1.x
- **Base de datos:** MariaDB 5.5.x, PostgreSQL 9.2.x
- **Lenguaje de script:** PHP 5.4, Python 2.7, Perl 5.16.3
- **Interfaz de escritorio:** Gnome 3.14, KDE 4.14
- **Servidor de pantalla:** X.org 7.7
- **Cliente de correo electrónico:** Evolution 3.12, Thunderbird 45
- **Navegador web:** Firefox 45
- **Office Suite:** LibreOffice 4.3.7

4.6. ¿Existen Otros Editores como en VIM, explique cuales?.

- Vi: Se abrirá el archivo correspondiente en texto plano, sin formato alguno. Si este no existe, se creará un archivo en blanco, el cual se guardará en el directorio desde donde se ejecutó el comando, siempre y cuando lo guarde al terminar la edición.

- Nano: Está presente por defecto en la mayoría de los sistemas basados en Linux, pero si este no es su caso.
- Emacs: A diferencia de vi, emacs es un editor en pantalla, por lo que puede desplazarse y escribir texto con más libertad. Sin embargo, el uso de los comandos puede ser incluso más confuso que en vi.

4.7. ¿Cómo se Interpreta un Inodo? ¿Es Posible?.

- Cada archivo en un sistema UNIX tiene asociado un inodo. El inodo contiene la información necesaria para que un proceso pueda acceder al archivo. Esta información incluye: propietario, derechos de acceso, tamaño, localización en sistema de archivos, etc.
- Cada vez que un archivo se abre, el inodo es leído por el kernel del servidor. Cuantos más archivos/carpetas tengas, más inodos se acumularán en la cuenta. Cuantos más inodos se creen, más recursos son consumidos por tu cuenta. Por eso es algo que comúnmente se limita en las compañías de alojamiento web en los servidores compartidos, donde no es posible que una cuenta use los recursos del sistema y deje sin poder a las demás cuentas.

4.8. ¿Diferencia entre Cat, Less y More?.

- EL comando **Cat**: Muestra el contenido de un archivo en la pantalla sin limites.
- El Comando **Less**: Muestra el contenido del archivo con una pantalla de desplazamiento de manera que puedas manejar entre paginas utilizando PgUp, PgDn, Home, y End.
- El Comando **More**: Te permite mostrar el resultado de la ejecución de un comando en la terminal de a una página a la vez. Esto es especialmente útil cuando se ejecuta un comando que causa un gran desplazamiento, como el comando ls o el comando du.

5. Conclusiones.

- El hecho de Manejar Comandos es muy sencillo, pero la cosa cambia cuando quieres añadirle algo más a la hora de crear, modificar, eliminar grupos usando opciones de cada comando.
- Aunque muchas opciones de cada comando tienen sus variaciones, hay que prestar mucha atención a la parte de digitar las opciones, ya que cada opción o es en minúscula o mayúscula.
- Si deseas escribir un comando como crear un grupo con la sesión de un usuario, ese usuario no lo podrá hacer sino está loguado con los permisos o privilegios del usuario root.
- Así como para Gestión de Grupos, existen varios comandos que te permiten crear, modificar, eliminar y/o verificar otras cosas relacionado a los grupos.
- No se necesita Internet para Instalar un Paquete que no se ejecute, basta con que lo instalemos desde la ISO o CD.

6. Recomendaciones.

6.1. Principales.

- Hay que recordar que cada comando tiene su propio uso y varios de ellos pueden servir para imprimir un solo resultado.
- Se recomienda discreción con los nombres de usuarios y grupos mencionados en este Laboratorio.

6.2. Secundarias.

- Si deseas cerrar sesión de un usuario con el que estás logueado ahora, sólo debes digitar en la consola **logout** y aparecerá la parte para que te loguees de nuevo.
- Tener muy en cuenta que cada ejecución de comando o comandos deben reconocidos con anterioridad.
- Si quieres salir del contenido de algún archivo, solo debes teclear Ctrl + x; Ctrl + z.

7. Bibliografía.

- Comandos Principales De Linux Guía Fácil.

<https://likegeeks.com/es/comandos-principales-de-linux/>

- Cómo usar comando who en Linux.

<https://www.solvetic.com/tutoriales/article/5139-como-usar-comando-who-en-linux/>

- Cómo usar comando Finger en Linux.

<https://www.solvetic.com/tutoriales/article/7833-como-usar-comando-finger-en-linux/>

- 4 comandos para conocer datos de nuestros HDD o particiones.

<https://blog.desdelinux.net/4-comandos-para-conocer-datos-de-nuestros-hdd-o-particiones/>

- Cómo ver sistema de archivos, discos o particiones Linux.

<https://www.solvetic.com/tutoriales/article/3700-como-ver-sistema-de-archivos-disco-particiones-linux/>

- Listar ficheros de un directorio – ls.

<https://www.cambiatealinux.com/listar-ficheros-de-un-directorio-ls>

- ls.

<https://francisconi.org/linux/comandos/ls>

- ¿Cómo saber el uso de inodos en Linux?.

<https://www.ochobitshacenunbyte.com/2019/04/08/como-saber-el-uso-de-inodos-en-linux/>

- 12 ejemplos prácticos del comando grep.

<https://www.librebyte.net/cli/12-ejemplos-practicos-del-comando-grep/>

- Ayuda con los comandos linux desde el terminal.

<https://www.profesionalreview.com/2016/07/23/ayuda-comandos-linux/>

- El comando more: Para qué sirve y cómo se utiliza.

https://www.neoguias.com/comando-more-linux/#Ejemplos_de_usos_del_comando_more

- 5 COMANDOS PARA VERIFICAR EL USO DE LA MEMORIA EN GNU/LINUX.

<https://maslinux.es/5-comandos-para-verificar-el-uso-de-la-memoria-en-gnu-linux/>

- 10 comandos útiles para manejar los procesos desde la consola Linux.

<https://www.cyberhades.com/2012/03/05/10-comandos-utiles-para-manejar-los-procesos-desde-la-consola-linux/>

- 9 comandos básicos Fdisk para gestionar el disco duro.

<https://openwebinars.net/blog/9-comandos-basicos-fdisk-para-gestionar-el-disco-duro/>

- TEMA 4. SISTEMA DE ARCHIVOS.

https://w3.ual.es/~acorral/DS0/Tema_4.pdf

- Reiniciar servidor Linux por SSH.

<https://clouding.io/kb/reiniciar-servidor-linux-por-ssh/>

- Guía práctica de los editores de texto nano y vi en Linux.

<https://docs.bluehosting.cl/tutoriales/servidores/guia-practica-de-los-editores-de-texto-nano-y-vi-en-linux.html>