

UNIVERSIDAD NACIONAL de
CORDOBA - Departamento Universitario
de Informática



Diplomatura SO Linux

**Prácticos curso Administración básica de
Sistemas Operativos GNU/Linux.**

BARRIRERO, Exequiel

Profesor: MIRIZIO, Esteban.

-2015-

in \LaTeX

Índice

1. Ejercicios Tema1: Introducción al entorno gráfico de un Sistema GNU/Linux y herramientas	1
1.1. Actividad 1.	1
1.1.1. Kernel: Definición	1
1.1.2. Versión actual de kernel estable en GNU's Not Unix! (GNU)/Linux	2
1.1.3. Versión mas usada de kernel estable en GNU/Linux	4
1.2. Actividad 2.	4
1.2.1. ¿De qué distribución deriva GNU/Linux Fedora?	4
1.3. Actividad 3.	4
1.3.1. Imprimir pantalla del escritorio GNU Network Object Model Environment (Gnome)	4
1.4. Actividad 4.	5
1.4.1. Navegación de directorios con Nautilus	5
1.5. Actividad 5.	10
1.5.1. LibreOffice Writer: Guardado de archivos .doc en .odt . .	10
1.6. Actividad 6.	11
1.6.1. Consolas virtuales y Graphical user interface toolkit (GTK)-Warning	11
2. Ejercicios Tema3: Línea de Comandos. Comandos básicosI	13
2.1. Actividad 1.	13
2.1.1. change directory (cd) command	13
2.2. Actividad 2.	13
2.2.1. Sudo y permisos de administración	13
2.3. Actividad 3.	16
2.3.1. less command	16
2.4. Actividad 4.	16
2.4.1. /bin and /sbin	16
2.5. Actividad 5.	18
2.5.1. Listado de permisos "ls -la" command	18
2.6. Actividad 6.	19
2.6.1. Listado de permisos "ls -l" command	19
2.7. Actividad 7.	20
2.7.1. mkdir make directory	20
2.8. Actividad 8.	21
2.8.1. Creación de archivos con "touch"	21
2.9. Actividad 9.	21
2.9.1. Cambio de permisos con "chmod"	21
2.10. Actividad 10.	22
2.10.1. Eliminar recursivamente con "rm" command	22
2.11. Actividad 11.	22

2.11.1. Copia recursiva de directorio con "cp" command	22
2.12. Actividad 12.	23
2.12.1. Renombrar archivos con "mv" command	23
2.13. Actividad 13.	24
2.13.1. Copia de dirs mediante path relativo	24
2.14. Actividad 14.	24
2.14.1. Enlaces simbólicos "ln" command	24
2.15. Actividad 15.	26
2.15.1. Navegación de directorios con Nautilus	26
2.16. Actividad 16.	27
2.16.1. Interpretación de comandos	27
3. Ejercicios Tema 4: Linea de comandos. Comandos básicos II	30
3.1. Actividad 1.	30
3.1.1. "find" and "locate" commands	30
3.2. Actividad 2.	30
3.2.1. locate -d command	30
3.3. Actividad 3.	32
3.3.1. "find" without name	32
3.4. Actividad 4.	32
3.4.1. "find" buscando por permisos de usuario	32
3.5. Actividad 5.	34
3.5.1. "find" para archivos modificados en un período de tiempo	34
3.6. Actividad 6.	36
3.6.1. "grep" command	36
3.7. Actividad 7.	37
3.7.1. "grep" recursivo	37
3.8. Actividad 8.	37
3.8.1. "df" commands	37
3.9. Actividad 9.	38
3.9.1. "tar.gz" command p/ comprimir	38
3.10. Actividad 10.	39
3.10.1. "tar.gz" command p/ descomprimir	39
4. Ejercicios Tema 6: Uso básico de la consola	40
4.1. Actividad 1.	40
4.1.1. Regular expressions - ^ , \$, . , *	40
4.2. Actividad 2.	42
4.2.1. Regular expressions - ^ , \$, . , *	42
4.3. Actividad 3.	43
4.3.1. Regular expressions - ^ , . , []	43
4.4. Actividad 4.	45
4.4.1. Regular expressions - ^ , \$, []	45
4.5. Actividad 5.	46

4.5.1. Regular expressions - ^ , \$, []	46
4.6. Actividad 6.	47
4.6.1. The command-line history	47
4.7. Actividad 7.	49
4.7.1. history, variable HISTSIZE	49
4.8. Actividad 8.	50
4.8.1. HISTCONTROL variable	50
4.9. Actividad 9.	51
4.9.1. clear history (history -c)	51
5. Parcial 1 - Parte 2 - Cuestionario Práctico	52
5.1. Actividad 1.	52
5.1.1. Regular expressions - ^ , \$, . , *	52
5.2. Actividad 2.	52
5.2.1. find command	52
5.3. Actividad 3.	52
5.3.1. Regular expressions and ls command	52
5.4. Actividad 4.	53
5.4.1. Regular expressions - Tar command	53
6. Ejercicios Tema 7: Shell scripting básico	54
6.1. Actividad 1.	54
6.1.1. STDERR redirection to file	54
6.2. Actividad 2.	55
6.2.1. STDOUT & STDERR redirection to file	55
6.3. Actividad 3.	55
6.3.1. Head - Tail - Pipe	55
6.4. Actividad 4.	57
6.4.1. Cat - Grep - Wc	57
6.5. Actividad 5.	58
6.5.1. Find & Sort script	58
6.6. Actividad 6.	59
6.6.1. “ - in between Grave accent character	59
6.7. Actividad 7.	61
6.7.1. FOR - loops and iteration	61
7. Ejercicios Tema 8: Herramientas para procesar texto	62
7.1. Actividad 1.	62
7.1.1. cat command	62
7.2. Actividad 2.	63
7.2.1. less command	63
7.3. Actividad 3.	64
7.3.1. grep & regular expressions	64
7.4. Actividad 4.	64

7.4.1. head command	64
7.5. Actividad 5.	65
7.5.1. sort command	65
7.6. Actividad 6.	69
7.6.1. cut command	69
7.7. Actividad 7.	73
7.7.1. sed command	73
8. Ejercicios Tema 10: Shell scripting intermedio	80
8.1. Actividad 1.	80
8.1.1. tar - gzip - date format output	80
8.2. Actividad 2.	81
8.2.1. script : "divisible por 2"	81
8.3. Actividad 3.	82
8.3.1. script: tar - gzip - date format output	82
8.4. Actividad 4.	85
8.4.1. head command	85
8.5. Actividad 5 (ejercitación evaluación).	85
8.5.1. script: find directories by name - for - if - test - ls	85
8.6. Actividad 6 (ejercitación evaluación).	85
8.6.1. script: True or False - for - if - test - ls	85
9. Ejercicios Tema 11: Configuración de Bash	86
9.1. Actividad 1.	86
9.1.1. Variable de entorno PS1	86
9.2. Actividad 2.	86
9.2.1. Limpiar History y /tmp	86
9.3. Actividad 3.	86
9.3.1. init.d and processes	86
9.4. Actividad 4.	86
9.4.1. backup script - tar gz	86
9.5. Actividad 5.	87
9.5.1. Alias for grep command	87
9.6. Actividad 6.	87
9.6.1. script: given a value to a variable	87
Bibliografía	88
Siglas y Acrónimos	90

Índice de figuras

1. <i>The Linux Kernel Archives [Linux Kernel Organization Inc., 2014a].</i>	3
2. <i>Capturando pantalla en ubuntu 14.04.</i>	5

3.	<i>Verificando que la captura se haya guardado correctamente en la folder /home/delivery/image/</i>	6
4.	<i>Navegación de archivos con Nautius folder: /home</i>	7
5.	<i>Navegación de archivos con Nautius folder: /etc</i>	8
6.	<i>Navegación de archivos con Nautius folder: /root.</i>	9
7.	<i>LibreOffice Writer: Guardado de archivos .doc en .odt.</i>	10
8.	<i>Linux virtual consoles.</i>	12

Índice de cuadros

1. Ejercicios Tema1: Introducción al entorno gráfico de un Sistema GNU/Linux y herramientas

- Conceptos básicos de sistemas operativos
 - Conceptos de multitarea y multiusuario
 - Explicar la naturaleza del software opensource
- Orígenes de GNU/Linux. Distribuciones GNU/Linux. Principios básicos de GNU/Linux.
- Entornos de escritorio más comunes: Gnome.
- Navegadores de sistemas archivos: Nautilus.
- Navegación web: Firefox.
- Ofimática: LibreOffice: Writer y Calc
- Multimedia: reproductores de audio y reproductores de video.
- Editores de texto: Gedit.
- Visor de PDF: Evince.
- Pasar de entorno gráfico a consola

1.1. Actividad 1.

1.1.1. Kernel: Definición

En informática, un núcleo o kernel (de la raíz germánica Kern, núcleo, hueso) es un software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo). Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al programador.

[Wikipedia Project, 2015c]

1.1.2. Versión actual de kernel estable en GNU/Linux

Here are several main categories into which kernel releases may fall:

- **Prepatch:**

Prepatch or “RC” kernels are mainline kernel pre-releases that are mostly aimed at other kernel developers and Linux enthusiasts. They must be compiled from source and usually contain new features that must be tested before they can be put into a stable release. Prepatch kernels are maintained and released by Linus Torvalds.

- **Mainline:**

Mainline tree is maintained by Linus Torvalds. It’s the tree where all new features are introduced and where all the exciting new development happens. New mainline kernels are released every 2-3 months.

- **Stable:**

After each mainline kernel is released, it is considered “stable.” Any bug fixes for a stable kernel are backported from the mainline tree and applied by a designated stable kernel maintainer. There are usually only a few bug-fix kernel releases until next mainline kernel becomes available – unless it is designated a “longterm maintenance kernel.” Stable kernel updates are released on as-needed basis, usually 2-3 a month.

- **Longterm:**

There are usually several “longterm maintenance” kernel releases provided for the purposes of backporting bugfixes for older kernel trees. Only important bugfixes are applied to such kernels and they don’t usually see very frequent releases, especially for older trees. Longterm release kernels are the Versions: 3.18, 3.14, 3.12, 3.10, 3.4, 3.2, 2.6.32.

- **Distribution kernels:**

Many Linux distributions provide their own “longterm maintenance” kernels that may or may not be based on those maintained by kernel developers. These kernel releases are not hosted at kernel.org and kernel developers can provide no support for them. It is easy to tell if you are running a distribution kernel. Unless you downloaded, compiled and installed your own version of kernel from kernel.org, you are running a distribution kernel. To find out the version of your kernel, run `uname -r`:

[Linux Kernel Organization Inc., 2014b]

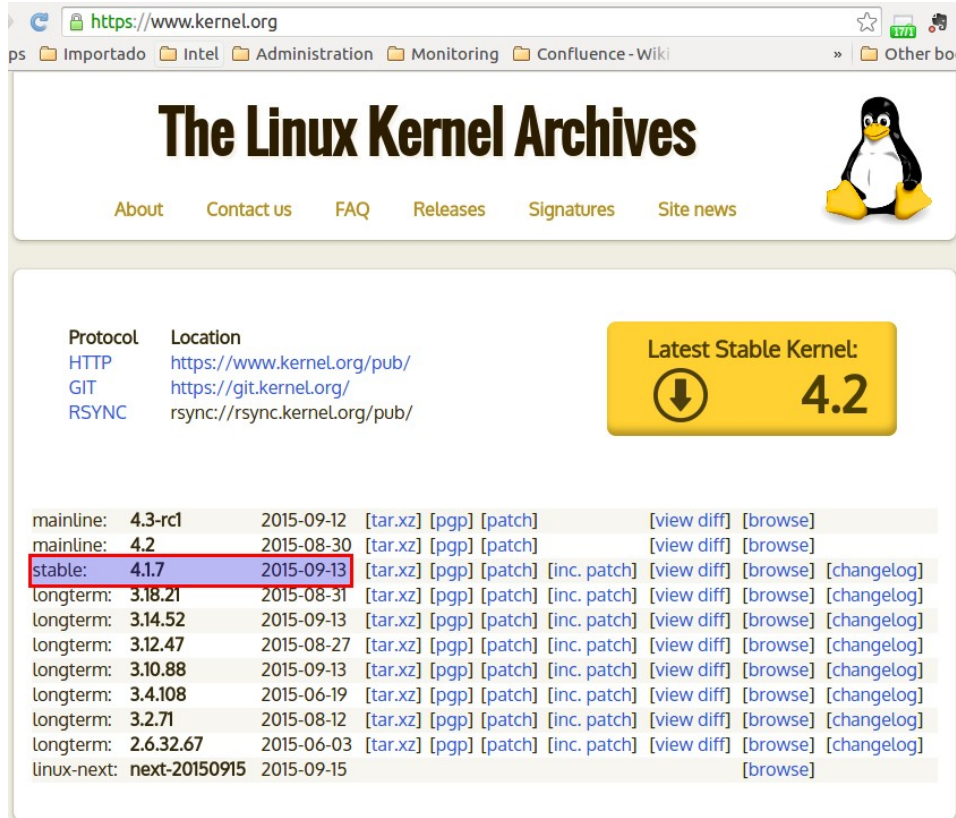
Por ejemplo:

```
delivery@delivery-laptop:~$ uname -r
3.16.0-33-generic
```



```
delivery@delivery-laptop:~$ uname -a
Linux delivery-laptop 3.16.0-33-generic #44~14.04.1-Ubuntu SMP Fri Mar
13 10:33:29 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
```

Luego la **última versión estable del Kernel de Linux** es la versión: **4.17**. Esto puede verificarse en el sitio web oficial de Linux Kernel Organization Inc (<https://www.kernel.org/>). Como se presenta en la figura debajo:



The screenshot shows the 'The Linux Kernel Archives' website. At the top, there's a navigation bar with links: About, Contact us, FAQ, Releases, Signatures, and Site news. Below this, a yellow box highlights the 'Latest Stable Kernel: 4.2' with a download icon. To the left, a table lists download protocols and locations. Below that, a table lists various kernel versions categorized as mainline, longterm, or linux-next, with links to download files and view diffs. The row for 'stable: 4.1.7' dated '2015-09-13' is highlighted with a red box.

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

mainline:	4.3-rc1	2015-09-12	[tar.xz]	[pgp]	[patch]	[view diff]	[browse]
mainline:	4.2	2015-08-30	[tar.xz]	[pgp]	[patch]	[view diff]	[browse]
stable:	4.1.7	2015-09-13	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.18.21	2015-08-31	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.14.52	2015-09-13	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.12.47	2015-08-27	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.10.88	2015-09-13	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.4.108	2015-06-19	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	3.2.71	2015-08-12	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	2.6.32.67	2015-06-03	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
linux-next:	next-20150915	2015-09-15					[browse]

Figura 1: The Linux Kernel Archives [Linux Kernel Organization Inc., 2014a].

1.1.3. Versión mas usada de kernel estable en GNU/Linux

Considerando que la versión estable 2.6.10 del Kernel de Linux se lanzó en Diciembre del 2003 y aún esta vigente y en use podemos decir que esta versión estable es la ampliamente desplegada y utilizada.

Versión 2.6, lanzada el 17 December 2003. Versión actual 2.6.32 - 2.6.39 End of Life (EOL) (maintained from May 2011 to August 2011), last stable release of the 2.6 kernel series. longterm: 2.6.32.67 2015-06-03

Por su parte, cabe considerar la versión y los releases 3.0 del Kernel ya que esta versión tomó lugar el 21 de Julio del 2011 y su último release longterm:3.18.21 (2015-08-31) demostrando una validez de ya cinco años.

[Wikipedia Project, 2015b]

1.2. Actividad 2.

1.2.1. ¿De qué distribución deriva GNU/Linux Fedora?

*“The Fedora Project is a global partnership of free software community members. The Fedora Project is sponsored by **Red Hat**, which invests in our infrastructure and resources to encourage collaboration and incubate innovative new technologies. Some of these technologies may later be integrated into Red Hat products. They are developed in Fedora and produced under a free and open source license from inception, so other free software communities and projects are free to study, adopt, and modify them. Read an overview to learn more about our mission, our community, our governance, and what makes Fedora unique. You can also learn about our vision and core values the foundations upon which the project is built. We also have information relating to our user base, and the objectives for our technical work.”*

[Red Hat, 2015]

1.3. Actividad 3.

1.3.1. Imprimir pantalla del escritorio Gnome

Imprimir pantalla del escritorio Gnome y guardar la imagen en el home del usuario dentro de un directorio llamado imagen. Cabe mencionar que el usuario es nombrado *delivery*.

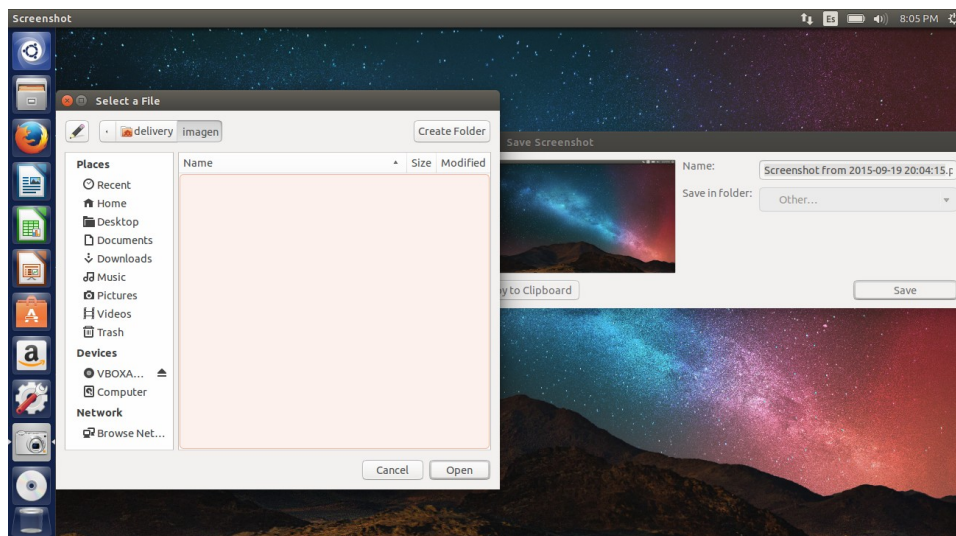


Figura 2: Capturando pantalla en ubuntu 14.04.

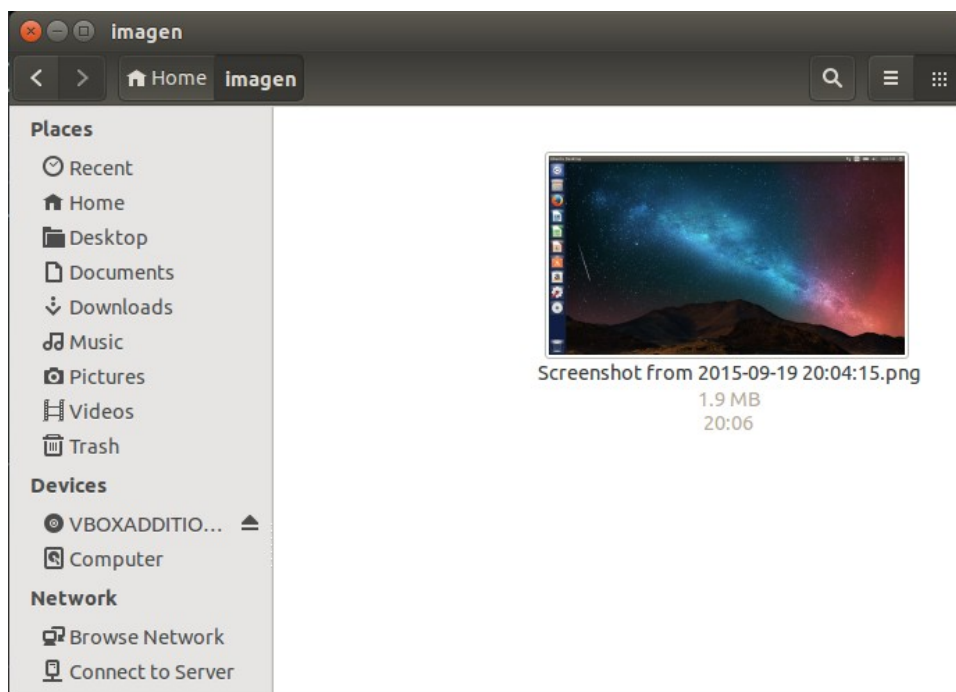


Figura 3: Verificando que la captura se haya guardado correctamente en la folder /home/delivery/imagen/.

1.4. Actividad 4.

1.4.1. Navegación de directorios con Nautilus

Navegar a través del Nautilus los siguientes directorios: `/home` `/etc` `/var/log` `/root` `/dev` ¿En que directorio/s no pudo acceder? ¿Qué tienen de particular los íconos de estos directorios a los que no pudo acceder?

Como usuario sin privilegios a través del navegador de archivos *nautilus* uno solo podría ingresar y tener control absoluto de los directorios y archivos dentro de */home*. Luego con este mismo usuario acceder a */etc*, */var/log*, */dev*, pero sin permisos de escritura o ejecución, solo lectura. Finalmente, el directorio */root* no es accesible a través de la interfaz gráfica ya que no poseemos permisos suficientes para realizar esta acción. Las carpetas a las cuales carecemos de acceso y permisos se presentan con un candado en el icono (considerar que esto no ocurre para el nuevo ubuntu 14.04 como puede verse en las capturas de pantalla).

Lo detallado en el párrafo anterior puede apreciarse en las siguientes capturas:

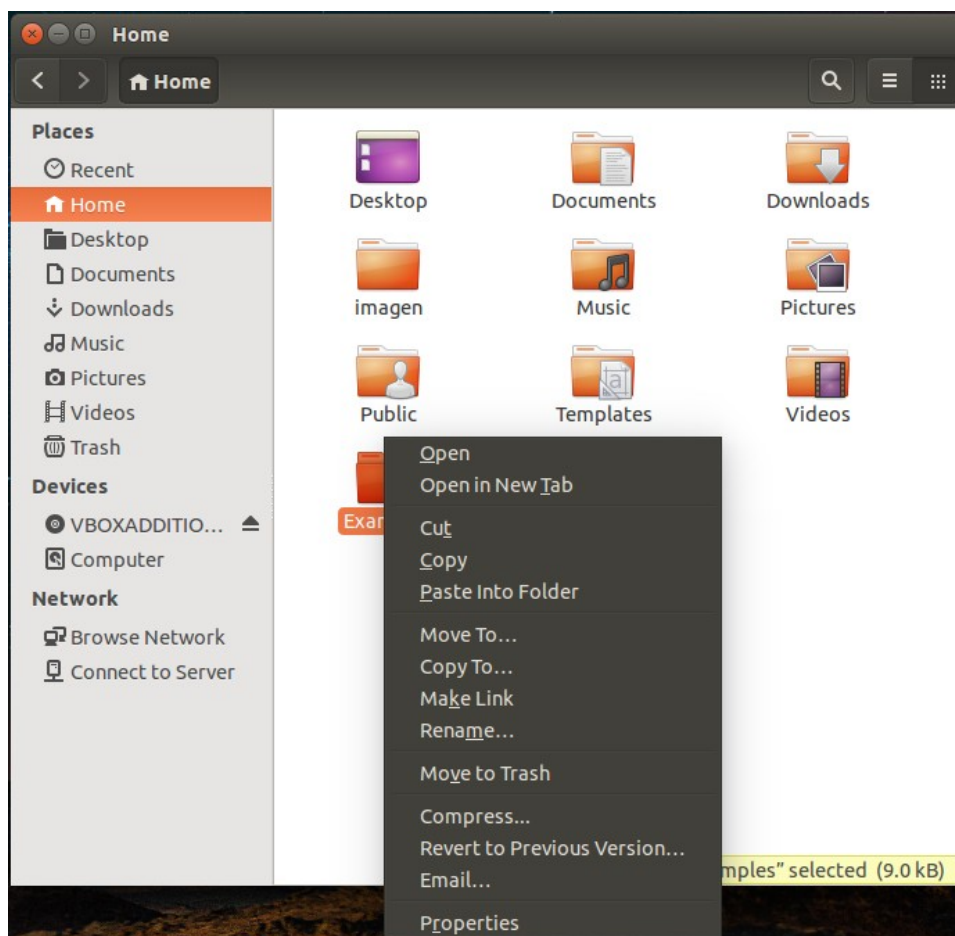


Figura 4: Navegación de archivos con Nautius folder: /home .

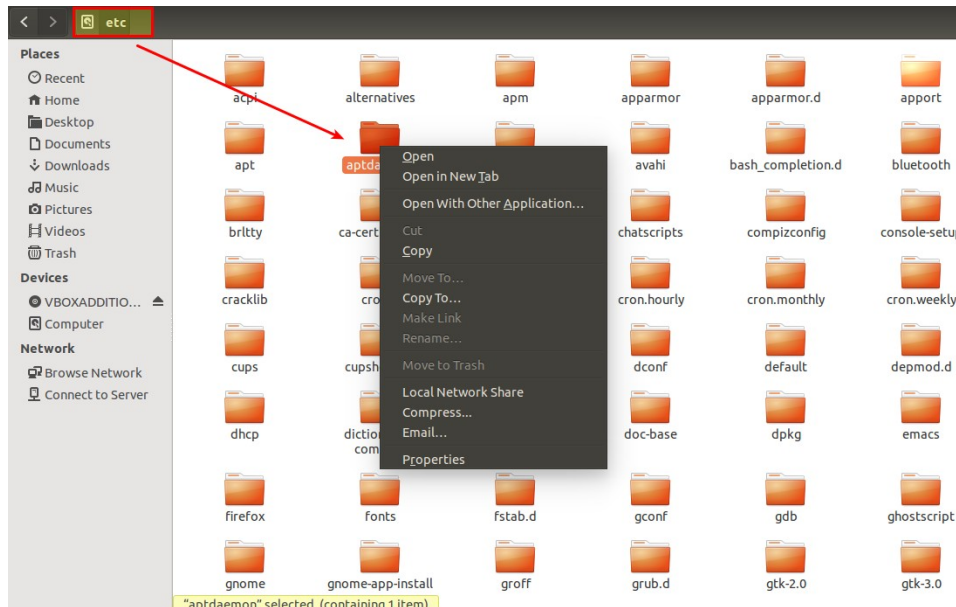


Figura 5: Navegación de archivos con Nautius folder: `/etc` .

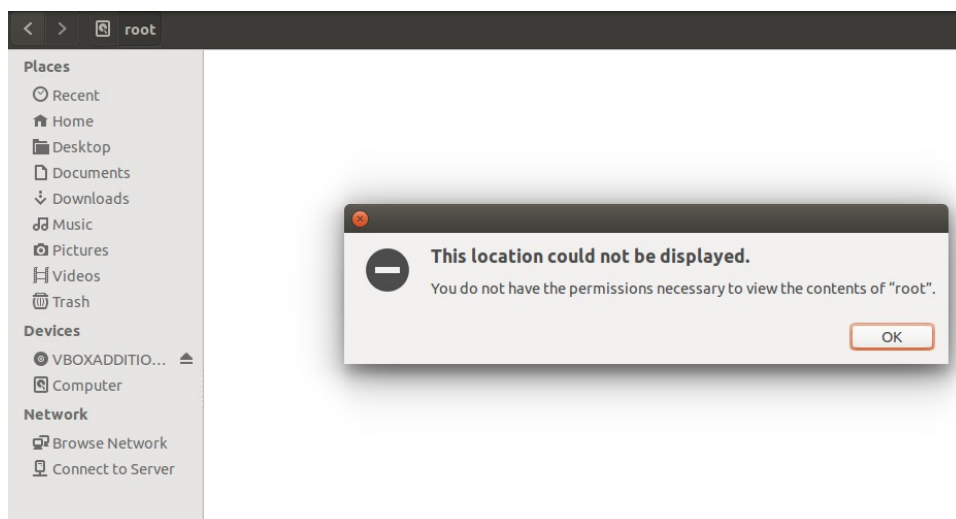


Figura 6: Navegación de archivos con Nautius folder: `/root`.

1.5. Actividad 5.

1.5.1. LibreOffice Writer: Guardado de archivos .doc en .odt

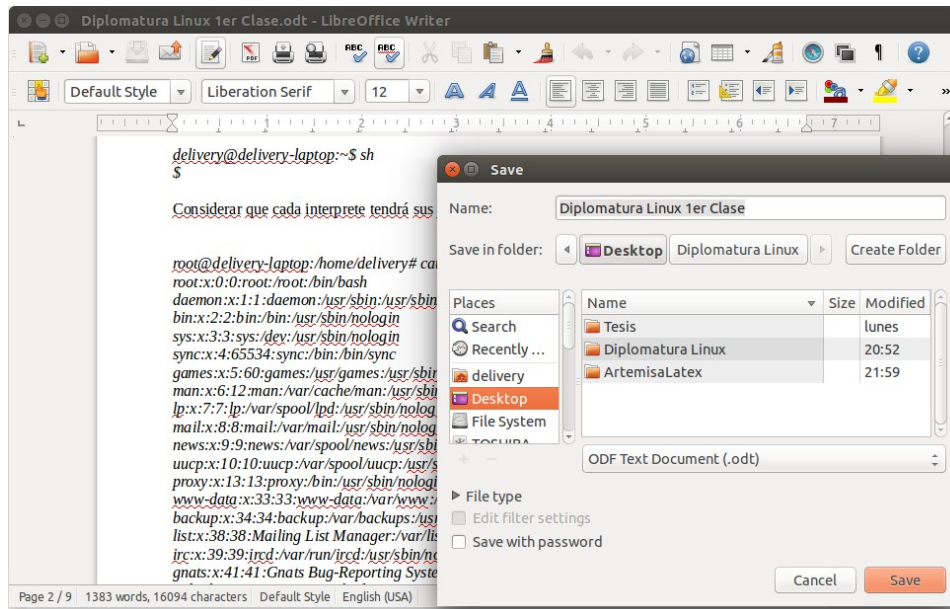


Figura 7: LibreOffice Writer: Guardado de archivos .doc en .odt.

A un archivo .doc, guardarlo como .odt con LibreOffice Writer.

1.6. Actividad 6.

1.6.1. Consolas virtuales y GTK-Warning

Ejecute la siguiente secuencia de teclas: **Alt+F2** y escriba “**gedit**”. ¿Qué sucedió? ¿Qué acción realiza la ejecución de **Alt+F2**?

Accedemos a la consola virtual número dos. Por su parte no es posible abrir un aplicativo que requiere de Gráfico User Interface (GUI) para funcionar.

Esto es un evento relacionado a la seguridad de SO normal. Linux es un sistema multiusuario donde muchos usuarios podrían estar loggeados, localmente o de forma remota en una sesión GUI. Luego, ¿qué ocurriría si otros users podrían abrir ventanas en tu escritorio sin su consentimiento? Claramente esto no sería un comportamiento deseado del SO. Por su parte esto permitiría abrir una ventana de gedit u otra aplicación más crítica como un navegador web de forma tal que a este usuario mal intencionado le permita leer todos nuestros inputs por teclado, pudiendo incluir datos confidenciales como nuestra cuenta bancaria. Es por esto que se utiliza *xhost*¹. Por otro lado, la sesión CLI de root no sabe en cual de los displays o pantallas debe abrir la ventana. Nuevamente cabe remarcar que podría haber varias, tanto locales como remotas. Por lo antedicho surge la necesidad de declarar la variable de entorno **DISPLAY**. De todas formas existen soluciones más prácticas desde la GUI de usuario normal para manejar estas sesiones, **gnomesu** en Gnome y **kdesu** en K desktop environment (KDE). Las cuales básicamente son parte de una librería para proveer de privilegios de super usuario a las aplicaciones de Gnome.

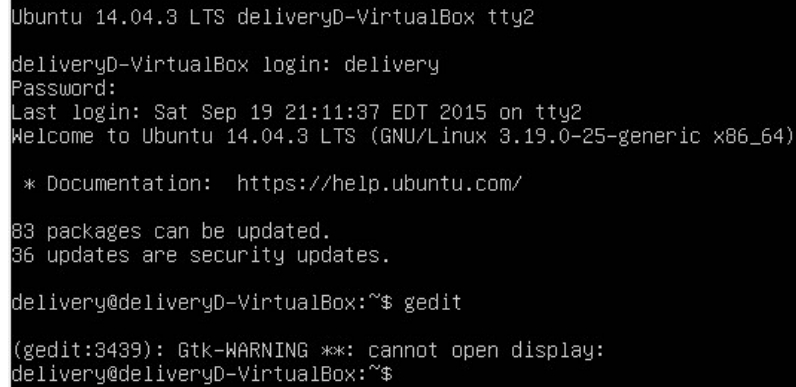
Cabe mencionar:

*“Virtual consoles: In the default Debian system, there are six switchable VT100-like character consoles available to start the command shell directly on the Linux host. Unless you are in a GUI environment, you can switch between the virtual consoles by **pressing the Left-Alt-key and one of the F1 — F6 keys simultaneously**. Each character console allows independent login to the account and offers the multiuser environment. This multiuser environment is a great Unix feature, and very addictive. If you are under the X Window System, you gain access to the character console 1 by pressing **Ctrl-Alt-F1** key, i.e., the left-Ctrl-key, the left-Alt-key, and the F1-key are pressed together. You can get back to the X Window System, normally running on the virtual console 7, by pressing **Alt-F7**. You can alternatively change to another virtual console, e.g. to the console 1, from the commandline.”*

¹NAME: xhost - server access control program for X. SYNOPSIS: xhost [[+-]name ...] DESCRIPTION: The xhost program is used to add and delete host names or user names to the list allowed to make connections to the X server. In the case of hosts, this provides a rudimentary form of privacy control and security. It is only sufficient for a workstation (single user) environment, although it does limit the worst abuses. Environments which require more sophisticated measures should implement the user-based mechanism or use the hooks in the protocol for passing other authentication data to the server.[Scheifler B.,]


```
# chvt 1
```

[Osamu Aoki, 2013]

A screenshot of a Linux virtual console window. The text is white on a black background. It shows the login process for Ubuntu 14.04.3 LTS on a virtual machine named 'deliveryD-VirtualBox'. The user 'delivery' logs in. The system displays the date, time, and last login information. It also shows the number of packages that can be updated and the number of security updates available. The user then runs the 'gedit' command, which results in a warning message from the GTK library: 'Gtk-WARNING **: cannot open display:'.

```
Ubuntu 14.04.3 LTS deliveryD-VirtualBox tty2
deliveryD-VirtualBox login: delivery
Password:
Last login: Sat Sep 19 21:11:37 EDT 2015 on tty2
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

83 packages can be updated.
36 updates are security updates.

delivery@deliveryD-VirtualBox:~$ gedit

(gedit:3439): Gtk-WARNING **: cannot open display:
delivery@deliveryD-VirtualBox:~$
```

Figura 8: *Linux virtual consoles.*

2. Ejercicios Tema3: Línea de Comandos. Comandos básicosI

- Concepto de líneas de comando. Presentación de shell bash.
- Entender la naturaleza de los privilegios del usuario root.
- Moverse y explorar el árbol de jerarquía del Sistema Operativo: ls, cd, mkdir, rmdir.
- Copiar, eliminar y renombrar archivos: cp, rm, mv, ln.
- Usuarios y permisos: chown, chmod, adduser, addgroup
- Crear y ver archivos: touch, less.

2.1. Actividad 1.

2.1.1. cd command

Desde el directorio /home/curso, cambiar al directorio /etc vía path absoluto. Vuelva al directorio /home/curso vía path absoluto.

```
delivery@ubuntu:~/curso$ pwd
/home/delivery/curso
delivery@ubuntu:~/curso$ cd /etc/
delivery@ubuntu:/etc$ pwd
/etc
delivery@ubuntu:/etc$ cd /home/delivery/curso/
delivery@ubuntu:~/curso$ pwd
/home/delivery/curso
```

Repita la acción anterior vía path relativo. Ejecutar el comando pwd en ambos casos para confirmar el cambio de directorio.

```
delivery@ubuntu:~/curso$ pwd
/home/delivery/curso
delivery@ubuntu:~/curso$ cd /etc/
delivery@ubuntu:/etc$ pwd
/etc
delivery@ubuntu:/etc$ cd ~delivery/curso/
delivery@ubuntu:~/curso$ pwd
/home/delivery/curso
```

2.2. Actividad 2.

2.2.1. Sudo y permisos de administración

¿Por qué el usuario “dui” tiene permisos de administración mientras que el usuario “curso” no? Algunas pistas: a.- Revisar a que grupo pertenecen el

usuario “dui” y “curso” b.- Revisar el contenido del archivo /etc/sudoers (archivo de configuración de Super-user do (sudo)) y ver que permisos tienen los grupos a los cuales pertenecen el usuario ”dui” y “curso”.

Se verifican los usuarios creados en el sistema a través del archivo de sistema *passwd* donde se encuentran definidas las cuentas de usuario. Por otro lado, puede verse que el user *curso* no se encuentra en el archivo *sudoers* por lo que carece de asignación de privilegios de súper usuario a través del comando *sudo*.

NOTA: se utilizó el user *delivery* en lugar de *dui*.

```
curso@ubuntu:~$ cat /etc/passwd | egrep -i 'root|delivery|curso'
root:x:0:0:root:/root:/bin/bash
delivery:x:1000:1000:delivery,,,:/home/delivery:/bin/bash
curso:x:1001:1001:curso,,,:/home/curso:/bin/bash
curso@ubuntu:~$ sudo su
[sudo] password for curso:
curso is not in the sudoers file. This incident will be reported.
```

Luego con el usuario *delivery* se lee y validan los privilegios declarados en */etc/sudoers*. Como ser el user *root*, con permisos *ALL=(ALL:ALL) ALL* (se detallará debajo su significado) y los más relevante para esta práctica revisar que se incluye una línea para permitir a todos los miembros del grupo *sudo* para ejecutar cualquier comando:

```
# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL
```

```
curso@ubuntu:~$ su - delivery
Password:
delivery@ubuntu:~$ sudo cat /etc/sudoers

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults env_reset
Defaults mail_badpass
Defaults secure_path=''/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin''

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
```

```
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on ``#include`` directives:

#includedir /etc/sudoers.d
```

Finalmente se verifica en */etc/group* lo comentado anteriormente donde el user *delivery* pertenece al grupo sudo con **GID = 27** y el user *curso* no.

```
delivery@ubuntu:~$ cat /etc/group | egrep -i 'curso|delivery|root'
root:x:0:
adm:x:4:syslog,delivery
cdrom:x:24:delivery
sudo:x:27:delivery
dip:x:30:delivery
plugdev:x:46:delivery
delivery:x:1000:
lpadmin:x:110:delivery
sambashare:x:111:delivery
curso:x:1001:
```

Sudo is a program designed to let system administrators allow some users to execute some commands as root (or another user). The basic philosophy is to give as few privileges as possible but still allow people to get their work done. sudo is also an effective way to log who ran which command and when.

As of most Debian based distributions, if you ask for the Desktop task during the installation, that pulls in sudo with a default configuration that automatically grants sudo-ing rights to any member of the sudo group. Depending on what user accounts you set up during the install, it's still possible that you may not have been added to that group - you can check by running groups.

Why sudo? Using sudo is better (safer) than opening a session as root for a number of reasons, including:

1. Nobody needs to know the root password (sudo prompts for the current user's password). Extra privileges can be granted to individual users temporarily, and then taken away without the need for a password change.
2. It's easy to run only the commands that require special privileges via sudo; the rest of the time, you work as an unprivileged user, which reduces the damage that mistakes can cause. Auditing/logging: when a sudo command is executed, the original username and the command are logged.
3. For the reasons above, switching to root using sudo -i (or sudo su) is usually deprecated because it cancels the above features.

Sudo is a program designed to allow a sysadmin to give limited root privileges to users and log root activity. The basic philosophy is to give as few privileges as possible but still allow people to get their work done.

[Debian org, 2014]

%sudo ALL=(ALL) ALL

- **%sudo** the group named "admin" (% prefix) is the group or users that are receiving the permises.
- **ALL=** The 2nd parameter refers to the host where the before mentioned group or users will have privileges. For this exampe in ALL hosts (this will work if you distribute the same sudoers file to many computers, or if someone access remotely - however no recommended from a secutiry point of view).
- **(ALL)** The third one is the user as you are running the command. In this case as any target user.
- **ALL** The last one is the commands allowed. So the users in sudo group, in any host as any user including root, can run any command.

2.3. Actividad 3.

2.3.1. less command

Revisar con el comando "less" el contenido del archivo /etc/passwd y en él buscar a los usuarios root, curso y dui

```
delivery@ubuntu:~$ less /etc/passwd | egrep 'root|curso|delivery'
root:x:0:0:root:/root:/bin/bash
delivery:x:1000:1000:delivery,,,:/home/delivery:/bin/bash
curso:x:1001:1001:curso,,,:/home/curso:/bin/bash
```

2.4. Actividad 4.

2.4.1. /bin and /sbin

¿Cuál es la diferencia entre los directorios /bin y /sbin?

■ /bin

This directory contains executable programs which are needed in single user mode and to bring the system up or repair it.

```
delivery@ubuntu:/sbin$ cd /bin
delivery@ubuntu:/bin$ pwd
/bin
delivery@ubuntu:/bin$ ls
bash          chvt          fgconsole    lesspipe     nc            ntfstruncate  sed
bunzip2       cp            fgrep        ln            nc.openbsd   ntfswipe      setfacl
busybox       cpio          findmnt      loadkeys     netcat       open          setfont
```

bzcat	dash	fuser	login	netstat	openvt	setupcon
bzcmp	date	fusermount	loginctl	nisdomainname	pidof	sh
bzdiff	dbus-cleanup-sockets	getfacl	lowntfs-3g	ntfs-3g	ping	sh.distrib
bzegrep	dbus-daemon	grep	ls	ntfs-3g.probe	ping6	sleep
bzexe	dbus-uuidgen	gunzip	lsblk	ntfs-3g.secaudit	plymouth	ss
bzfgrep	dd	gzexe	lsmod	ntfs-3g.usermap	plymouth-upstart-bridge	static-sh
bzgrep	df	gzip	mkdir	ntfscat	ps	stty
bzip2	dir	hostname	mknod	ntfsck	pwd	su
bzip2recover	dmesg	ip	mktemp	ntfscluster	rbash	sync
bzless	dnsdomainname	kbd_mode	more	ntfscmp	readlink	tailf
bzmore	domainname	kill	mount	ntfsdump_logfile	red	tar
cat	dumpkeys	kmod	mountpoint	ntfsfix	rm	tempfile
chac1	echo	less	mt	ntfsinfo	rmdir	touch
chgrp	ed	lessecho	mt-gnu	ntfsls	rnano	true
chmod	egrep	lessfile	mv	ntfsmftalloc	running-in-container	udevadm
chown	false	lesskey	nano	ntfsmove	run-parts	

■ /sbin

Like /bin, this directory holds commands needed to boot the system, but which are usually not executed by normal users.

```
delivery@ubuntu:/sbin$ ls
acpi_available  e2fsck      fstrim-all  iptables-apply  mkfs.bfs      nameif      rmdir
agetty          e2image     gdisk       iptables-restore  mkfs.cramfs   ntfsclone   route
alsa            e2label     getcap      iptables-save    mkfs.ext2     ntfsop      rtacct
apm_available  e2undo     getpcaps    iptunnel         mkfs.ext3     ntfslabel   rtmon
apparmor_parser  ethtool    getty       isosize         mkfs.ext4     ntfsresize  runlevel
badblocks      fatlabel    halt        iwconfig        mkfs.ext4dev  ntfsundelete  setcap
biosdevname     fdisk       hdparm      iwevent         mkfs.fat      on_ac_power  setvtrgb
blkid           findfs      hwclock     iwgetid         mkfs.minix    pam_tally   sfdisk
blockdev        fixparts   ifconfig    iwlist          mkfs.msdos    pam_tally2  sgdisk
bridge          fsck        ifdown      iwpriv          mkfs.ntfs     parted      shadowconfig
capsh           fsck.cramfs  ifquery     iwspy           mkfs.vfat     partprobe   shutdown
cfdisk          fsck.ext2   ifup        kbdrate         mkhomedir_helper  pivot_root  slattach
cgdisk          fsck.ext3   init        killall5        mkntfs        plipconfig  start
crda            fsck.ext4   initctl     ldconfig        mkswap        plymouthd   startpar
ctrlaltdel      fsck.ext4dev  insmod     ldconfig.real   mntctl        poweroff    startpar-upstart-inject
debugfs         fsck.fat    installkernel  logsave        modinfo       rarp        start-stop-daemon
depmod          fsck.minix  ip          losetup         modprobe      raw         status
dhclient        fsck.msdos  ip6tables  lsmod           mountall      reboot     stop
dhclient-script  fsck.nfs   ip6tables-apply  MAKEDEV        mount.fuse    regdbdump  sulogin
dmsetup         fsck.vfat  ip6tables-restore  mii-tool       mount.lowntfs-3g  reload    swaplabel
dosfsck         fsfreeze   ip6tables-save  mkdosfs        mount.ntfs    resize2fs  swapoff
dosfslabel      fstab-decode  ipmaddr      mke2fs         mount.ntfs-3g  resolvconf  swapon
dumpe2fs        fstrim     iptables     mkfs           mount.vboxsf  restart    switch_root
```

Directorios relacionados:

■ /usr/bin

This is the primary directory for executable programs. Most programs executed by normal users which are not needed for booting or for repairing the system and which are not installed locally should be placed in this directory.

■ /usr/local

This is where programs which are local to the site typically go.

■ /usr/local

This is where programs which are local to the site typically go.

■ /usr/local/bin

Binaries for programs local to the site.

■ /usr/local/sbin

Locally installed programs for system administration.

[Wikipedia Project, 2015a]

2.5. Actividad 5.

2.5.1. Listado de permisos "ls -la" command

Liste el contenido del directorio /home/curso (incluido los archivos ocultos). Se requiere saber el permiso de acceso, dueño y grupo de cada archivo del directorio.

```
delivery@ubuntu:~$ cd /home/delivery/
delivery@ubuntu:~$ pwd
/home/delivery
delivery@ubuntu:~$ ls -la
total 128
drwxr-xr-x 19 delivery delivery 4096 Sep 20 19:54 .
drwxr-xr-x  4 root      root      4096 Sep 20 19:49 ..
-rw-r----- 1 delivery delivery 1295 Sep 20 19:38 .bash_history
-rw-r----- 1 delivery delivery  220 Sep 16 10:17 .bash_logout
-rw-r----- 1 delivery delivery 3637 Sep 16 10:17 .bashrc
drwx----- 14 delivery delivery 4096 Sep 20 19:54 .cache
drwx-----  3 delivery delivery 4096 Sep 20 14:03 .compiz
drwx----- 17 delivery delivery 4096 Sep 20 19:54 .config
drwxrwxr-x  2 delivery delivery 4096 Sep 20 19:41 curso
drwx-----  3 delivery delivery 4096 Sep 20 17:56 .dbus
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Desktop
drwxrwxr-x  3 delivery delivery 4096 Sep 20 19:12 DiploLinux
-rw-r----- 1 delivery delivery   33 Sep 20 14:03 .dmrc
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Documents
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Downloads
drwx-----  3 delivery delivery 4096 Sep 20 17:56 .gconf
-rw-r----- 1 delivery delivery 1590 Sep 20 17:56 .ICEauthority
drwx-----  3 delivery delivery 4096 Sep 19 21:11 .local
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Music
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Pictures
drwx-----  3 delivery delivery 4096 Sep 20 19:54 .pki
-rw-r----- 1 delivery delivery  675 Sep 16 10:17 .profile
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Public
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Templates
-rw-r----- 1 delivery delivery    5 Sep 20 17:56 .vboxclient-clipboard.pid
-rw-r----- 1 delivery delivery    5 Sep 20 17:56 .vboxclient-display.pid
-rw-r----- 1 delivery delivery    5 Sep 20 17:56 .vboxclient-draganddrop.pid
-rw-r----- 1 delivery delivery    5 Sep 20 17:56 .vboxclient-seamless.pid
drwxr-xr-x  2 delivery delivery 4096 Sep 19 21:11 Videos
-rw-r----- 1 delivery delivery   51 Sep 20 17:56 .Xauthority
-rw-r----- 1 delivery delivery  908 Sep 20 17:56 .xsession-errors
-rw-r----- 1 delivery delivery 1294 Sep 20 17:14 .xsession-errors.old
```

Permisos a archivos Para permitir establecer los permisos en un archivo contamos con el comando `chmod`. Este comando funciona con la siguiente sintaxis:

```
Chmod [opciones] permisos archivo/directorio
```

Antes de adentrarnos en el comando es necesario explicar algunas cosas. Tenemos dos maneras de asignar permisos a los usuarios, mediante un modo octal y modo carácter.

Modo Octal El modo octal responde a la combinación de los tres permisos con las tres clases de usuario formando un numero binario de 3 cifras donde:

- El primer dígito corresponde a los permisos del dueño
- El segundo a los del grupo
- El tercero al resto de los usuarios

La instrucción quedaría así:

```
chmod 760 archivo.txt
```

Modo carácter Posee 3 modificadores que permiten realizar la tarea:

- + : añade un modo
- - : elimina un modo
- = : especifica un modo (sobrescribiendo el anterior)

Los modos son **r (read)**, **w (write)**, **x(ejecutar)**. y los usuarios están representados por:

- u: dueño
- g: grupo
- o : otros
- a : todos

Entonces si quiero agregar el permiso de escribir a todos, escribo:

```
chmod a+w archivo.txt
```

[Mirizio E., 2015a]

2.6. Actividad 6.

2.6.1. Listado de permisos "ls -l" command

Liste el contenido del directorio /etc. Debe aparecer cada archivo contenido en una línea aparte sin detalles de permisos de acceso, dueño o grupo.

```
delivery@ubuntu:/etc$ pwd
/etc
delivery@ubuntu:/etc$ ls -l
acpi
adduser.conf
alternatives
anacrontab
apg.conf
apm
```



```

apparmor
apparmor.d
apport
apt
at.deny
at-spi2
avahi
bash.bashrc
bash_completion
bash_completion.d
bindresvport.blacklist
blkid.conf
blkid.tab
bluetooth
...
update-manager
update-motd.d
update-notifier
UPower
upstart-xsessions
vim
vtrgb
w3m
wgetrc
wodim.conf
wpa_supplicant
X11
xdg
xml
zsh_command_not_found

```

2.7. Actividad 7.

2.7.1. mkdir make directory

Cree en el directorio /home/curso la siguiente estructura de directorio

```

./raiz
|__bin
|__home
| |__ApellidoNombre
| |__Desktop
| |__bin
|__var
..|__log

```

```

curso@ubuntu:~$ pwd
/home/curso

```

```

curso@ubuntu:~$ mkdir ./raiz ./raiz/bin ./raiz/home ./raiz/home/Barrirero
./raiz/home/Desktop ./raiz/home/bin ./raiz/var ./raiz/var/log

```

```

curso@ubuntu:~$ ls -R
.:

```

```

raiz

./raiz:
bin  home  var

./raiz/bin:

./raiz/home:
BarrireroExequiel  bin  Desktop

./raiz/home/BarrireroExequiel:

./raiz/home/bin:

./raiz/home/Desktop:

./raiz/var:
log

./raiz/var/log:

```

¿Qué significa ./?

Refiere a "este directorio", es decir al directorio actual en el que nos encontramos posicionados en la terminal.

2.8. Actividad 8.

2.8.1. Creación de archivos con "touch"

Sobre el esquema de directorios creado en la Actividad 7, cree con "touch" un archivo en /home/curso/raiz/var/log/messages

```

curso@ubuntu:~$ touch /home/curso/raiz/var/log/messages
curso@ubuntu:~$ ls -la /home/curso/raiz/var/log/
total 8
drwxrwxr-x 2 curso curso 4096 Sep 29 18:58 .
drwxrwxr-x 3 curso curso 4096 Sep 20 22:38 ..
-rw-rw-r-- 1 curso curso    0 Sep 29 18:57 messages

```

2.9. Actividad 9.

2.9.1. Cambio de permisos con "chmod"

Liste con "ls" los permisos de acceso del archivo creado en la Actividad 8. Luego con "chmod" cambie los permisos de acceso del archivo creado en la Actividad 8 a rwx solo para el grupo.

```

curso@ubuntu:~$ ls -la /home/curso/raiz/var/log/
total 8
drwxrwxr-x 2 curso curso 4096 Sep 29 18:58 .
drwxrwxr-x 3 curso curso 4096 Sep 20 22:38 ..
-rw-rw-r-- 1 curso curso    0 Sep 29 18:57 messages
curso@ubuntu:~$ chmod g+rw /home/curso/raiz/var/log/messages
curso@ubuntu:~$ ls -la /home/curso/raiz/var/log/
total 8
drwxrwxr-x 2 curso curso 4096 Sep 29 18:58 .
drwxrwxr-x 3 curso curso 4096 Sep 20 22:38 ..
-rw-rw-r-- 1 curso curso    0 Sep 29 18:57 messages

```

2.10. Actividad 10.

2.10.1. Eliminar recursivamente con "rm" command

Elimine recursivamente el directorio "/home/curso/raiz/var"

```

curso@ubuntu:~$ rm -r /home/curso/raiz/var/
curso@ubuntu:~$ ll /home/curso/raiz/
total 16
drwxrwxr-x  4 curso curso 4096 Sep 29 19:07 ./
drwxr-xr-x 17 curso curso 4096 Sep 29 19:02 ../
drwxrwxr-x  2 curso curso 4096 Sep 20 22:38 bin/
drwxrwxr-x  5 curso curso 4096 Sep 20 22:38 home/

```

2.11. Actividad 11.

2.11.1. Copia recursiva de directorio con "cp" command

Cree el directorio /home/curso/raiz/etc. Luego copie TODO el contenido del directorio /etc/ dentro de /home/curso/raiz/etc

```

curso@ubuntu:~$ mkdir /home/curso/raiz/etc
curso@ubuntu:~$ cp /etc/* /home/curso/raiz/etc/

curso@ubuntu:~$ sudo cp -R /etc/* /home/curso/raiz/etc/
curso@ubuntu:~$ ls -l /home/curso/raiz/etc/
total 1016
drwxr-xr-x  3 root  root   4096 Sep 29 19:21 acpi
-rw-r--r--  1 curso curso 2981 Sep 29 19:21 adduser.conf
drwxr-xr-x  2 root  root   4096 Sep 29 19:21 alternatives
-rw-r--r--  1 curso curso   401 Sep 29 19:21 anacrontab
-rw-r--r--  1 curso curso   112 Sep 29 19:21 apg.conf
drwxr-xr-x  6 root  root   4096 Sep 29 19:21 apm
drwxr-xr-x  3 root  root   4096 Sep 29 19:21 apparmor
drwxr-xr-x  8 root  root   4096 Sep 29 19:21 apparmor.d
drwxr-xr-x  4 root  root   4096 Sep 29 19:21 appport
drwxr-xr-x  6 root  root   4096 Sep 29 19:21 apt
-rw-r----- 1 root  root    144 Sep 29 19:21 at.deny
drwxr-xr-x  2 root  root   4096 Sep 29 19:21 at-spi2

```

```

drwxr-xr-x 3 root root 4096 Sep 29 19:21 avahi
-rw-r--r-- 1 curso curso 2177 Sep 29 19:21 bash.bashrc
-rw-r--r-- 1 curso curso 45 Sep 29 19:21 bash_completion
drwxr-xr-x 2 root root 4096 Sep 29 19:21 bash_completion.d
-rw-r--r-- 1 curso curso 356 Sep 29 19:21 bindresvport.blacklist
-rw-r--r-- 1 curso curso 321 Sep 29 19:21 blkid.conf
lrwxrwxrwx 1 root root 15 Sep 29 19:21 blkid.tab -> /dev/.blkid.tab
drwxr-xr-x 2 root root 4096 Sep 29 19:21 bluetooth
drwxr-xr-x 2 root root 4096 Sep 29 19:21 bonobo-activation
drwxr-xr-x 2 root root 4096 Sep 29 19:21 byobu
drwxr-xr-x 3 root root 4096 Sep 29 19:21 ca-certificates
...
-rw-r--r-- 1 curso curso 321 Sep 29 19:21 updatedb.conf
drwxr-xr-x 3 root root 4096 Sep 29 19:21 update-manager
drwxr-xr-x 2 root root 4096 Sep 29 19:21 update-motd.d
drwxr-xr-x 2 root root 4096 Sep 29 19:21 update-notifier
drwxr-xr-x 2 root root 4096 Sep 29 19:21 UPower
-rw-r--r-- 1 curso curso 222 Sep 29 19:21 upstart-xsessions
drwxr-xr-x 2 root root 4096 Sep 29 19:21 vim
lrwxrwxrwx 1 root root 23 Sep 29 19:21 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 2 root root 4096 Sep 29 19:21 w3m
-rw-r--r-- 1 curso curso 4812 Sep 29 19:21 wgetrc
-rw-r--r-- 1 curso curso 1343 Sep 29 19:21 wodim.conf
drwxr-xr-x 2 root root 4096 Sep 29 19:21 wpa_supplicant
drwxr-xr-x 10 root root 4096 Sep 29 19:21 X11
drwxr-xr-x 4 root root 4096 Sep 29 19:21 xdg
drwxr-xr-x 2 root root 4096 Sep 29 19:21 xml
-rw-r--r-- 1 curso curso 349 Sep 29 19:21 zsh_command_not_found

```

NOTA: man for "cp" command

CP (1)

NAME

cp - copy files and directories

SYNOPSIS

```

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

```

DESCRIPTION

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

...

```

-R, -r, --recursive
    copy directories recursively

```

...

2.12. Actividad 12.

2.12.1. Renombrar archivos con "mv" command

Renombre con "mv" el directorio /home/curso/raiz por /home/curso/root

```

curso@ubuntu:~$ ls -ls /home/curso/
total 2008
  4 drwxrwxr-x 3 curso curso    4096 Sep 21 19:25 BarrireroExequiel
1968 -rw-rw-r-- 1 curso curso 2013003 Sep 21 21:13 BarrireroExequiel.tar.gz
  4 drwxr-xr-x 2 curso curso    4096 Sep 29 19:18 Desktop
  4 drwxr-xr-x 2 curso curso    4096 Sep 21 14:06 Documents
  4 drwxr-xr-x 2 curso curso    4096 Sep 21 19:30 Downloads
  4 drwxr-xr-x 2 curso curso    4096 Sep 21 14:06 Music
  4 drwxr-xr-x 2 curso curso    4096 Sep 21 20:53 Pictures
  4 drwxr-xr-x 2 curso curso    4096 Sep 21 14:06 Public
  4 drwxrwxr-x 5 curso curso    4096 Sep 29 19:20 root
  4 drwxr-xr-x 2 curso curso    4096 Sep 21 14:06 Templates
  4 drwxr-xr-x 2 curso curso    4096 Sep 21 14:06 Videos

curso@ubuntu:~$ ls -ls /home/curso/ | grep root
  4 drwxrwxr-x 5 curso curso    4096 Sep 29 19:20 root

```

2.13. Actividad 13.

2.13.1. Copia de dirs mediante path relativo

Copie el directorio /home/curso/root/var/log a /home/curso/root/var/log.original utilizando PATH relativo desde /home/curso

```

curso@ubuntu:~$ pwd
/home/curso
curso@ubuntu:~$ cp -R ./root/var/log/ ./root/var/log.original

curso@ubuntu:~$ pwd
/home/curso
curso@ubuntu:~$ ls ./root/var/
log log.original

curso@ubuntu:~$ ls -la ./root/var/
total 16
drwxrwxr-x 4 curso curso 4096 Sep 29 19:44 .
drwxrwxr-x 6 curso curso 4096 Sep 29 19:42 ..
drwxrwxr-x 2 curso curso 4096 Sep 29 19:42 log
drwxrwxr-x 2 curso curso 4096 Sep 29 19:44 log.original

```

2.14. Actividad 14.

2.14.1. Enlaces simbólicos "ln" command

Cree un link simbólico de /etc/group en /home/curso

```

curso@ubuntu:~$ ln -s /etc/group /home/curso/
curso@ubuntu:~$ ls -la /home/curso/ | grep group
lrwxrwxrwx 1 curso curso    10 Sep 29 19:55 group -> /etc/group

curso@ubuntu:~$ ls -li /home/curso/ | grep group
533568 group
curso@ubuntu:~$ ls -li /etc/ | grep group
418231 group

```

Enlaces de Ficheros

Los enlaces ofrecen la posibilidad de dar a un único fichero múltiples nombres. Estos ficheros van a ser identificados mediante el sistema operativo por su número de inodo, el cual se genera de forma semialeatoria. Solo para ficheros y sólo en particiones linux.

Un inodo es un enlace que resulta el único identificador del fichero para el sistema de ficheros. Un directorio, por tanto, será una lista de números de inodo con sus correspondientes nombres de fichero. Cada nombre de fichero en un directorio es un enlace a un inodo particular.

■ *Enlaces Duros o hard links:*

La orden `ln` es usada para crear enlaces para un fichero.

Usando `ls -i`, veremos el número de inodo para el fichero.

`# ln fichero` creará un enlace para fichero.

`ln fichero fichero2` creará un enlace llamado `fichero2` que corresponderá al mismo fichero.

Utilizando `ls -i` veremos que los dos ficheros tienen el mismo inodo.

`# ls -i fichero fichero2`

Un fichero estará definitivamente eliminado del sistema cuando no queden enlaces a él. En realidad, la norma es que los ficheros tengan solamente un enlace duro.

Un modo de saber cuantos enlaces tiene un fichero es con la orden `ls -l`. Fíjate en la salida estándar por pantalla, la primera columna indica los permisos, como vimos en lecciones pasadas, y una segunda columna con un número te indicará el número de enlaces del fichero, o, si es un directorio, el número de directorios que contiene, en nuestro ejemplo te mostraría lo siguiente:

```
ls -l fichero fichero2
```

```
-rw-r--r- 2 root root 12 Aug 5 16:51 fichero
```

```
-rw-r--r- 2 root root 12 Aug 5 16:50 fichero2
```

un directorio, por tanto, no es otra cosa que un fichero que contiene información sobre la dirección del enlace al inodo. También, cada directorio tiene al menos dos enlaces duros en él: `.` (punto) enlace que apunta a sí mismo y `..` (punto punto) enlace que apunta al directorio padre. En el directorio raíz (`/`), el enlace `..` (punto punto) simplemente apunta a `/`.

Buscar todos los enlaces duros a un fichero. En ciertas ocasiones puede resultar difícil localizar en que partes del árbol de directorio existen enlaces a determinados archivos. Para encontrarlos lo podemos hacer con la orden `find`: `find / -inum número`

■ *Enlaces Simbólicos*

Un enlace simbólico permite dar a un fichero el nombre de otro, pero no enlaza el fichero con un inodo, es decir, en realidad lo que hacemos es enlazar directamente al nombre del fichero. Esto podría parecerse bastante a lo que Windows nos tiene acostumbrados.

Con la orden `ln -s` creamos un enlace simbólico a un fichero. Por ejemplo:

```
ln -s archivo archivo2
```

Hay que tener en cuenta que el nombre del enlace simbólico no soporta rutas completas, por lo que para crearlo, será imprescindible situarse dentro del directorio en el que queramos que quede colocado dicho enlace.

Si lo verificamos de nuevo con la orden `ls -l` vemos que el fichero fichero es un enlace simbólico apuntando a fichero2

```
ls -l fichero fichero2
```

Los bits de permisos en un enlace simbólico no se usan (siempre aparecen como `(rwxrwxrwx)`). En su lugar, los permisos del enlace simbólico son determinados por los permisos del fichero apuntado. Asimismo, si el fichero apuntado es eliminado, los enlaces simbólicos permanecen, pero ya no serán válidos y carecerán de sentido.

Los enlaces duros y simbólicos son similares en su funcionamiento, pero hay algunas diferencias. Pueden crearse enlaces simbólicos a un fichero que no esté en el mismo dispositivo de almacenamiento. Los enlaces simbólicos son procesados por el núcleo de forma diferente a los duros, lo cual es solo una diferencia técnica, pero a veces importante. Los enlaces simbólicos son de ayuda puesto que identifican al fichero al que apuntan; con enlaces duros no es tan fácil saber que fichero esta enlazado al mismo inodo.

Aunque en un principio no pudiera parecernos que los enlaces valgan para mucho, el sistema operativo los usa muy a menudo, Los enlaces simbólicos son, por ejemplo, especialmente importantes para las imágenes de las librerías compartidas en `/lib`, lo que facilita mucho la conexión de los diferentes programas con esas librerías.

[Asociación Cultural U.T.L.A.I. (Cantabria España), 2009]

2.15. Actividad 15.

2.15.1. Navegación de directorios con Nautilus

Luego de la ejecución de un comando `ls -l` ¿Qué significa el número que se encuentra entre el listado de permisos y el nombre del usuario dueño?

```
curso@atlas: >ls -l
-rwx---- 1 curso curso 11 2011-05-11 23:56 password
```

Un modo de saber cuantos enlaces tiene un fichero es con la orden `ls -l`. Fíjate en la salida estándar por pantalla, la primera columna indica los permisos, como vimos en lecciones pasadas, y una segunda columna con un número te indicará el número de enlaces del fichero, o, si es un directorio, el número de directorios que contiene, en nuestro ejemplo te mostraría lo siguiente:

```
ls -l fichero fichero2
-rw-r--r-- 2 root root 12 Aug 5 16:51 fichero
-rw-r--r-- 2 root root 12 Aug 5 16:50 fichero2
```

Un directorio, por tanto, no es otra cosa que un fichero que contiene información sobre la dirección del enlace al inodo. También, cada directorio tiene al menos dos enlaces duros en el: `.` (punto) enlace que apunta a si mismo y `..` (punto punto) enlace que apunta al directorio padre. En el directorio raíz (`/`), el enlace `..` (punto punto) simplemente apunta a `/`.

[Asociación Cultural U.T.L.A.I. (Cantabria España), 2009]

2.16. Actividad 16.

2.16.1. Interpretación de comandos

Explicar cada uno de los 6 comandos siguientes:

```
curso@atlas:~/borrar$ touch hola
curso@atlas:~/borrar$ ls -l
total 0
-rw-r--r-- 1 curso users 0 mar 26 23:45 hola
curso@atlas:~/borrar$ chmod u-w .
curso@atlas:~/borrar$ touch chau
touch: no se puede efectuar 'touch' sobre «chau»: Permiso denegado
curso@atlas:~/borrar$ touch hola
curso@atlas:~/borrar$ rm hola
rm: no se puede borrar «hola»: Permiso denegado
```

1. curso@atlas: ~/borrar\$ touch hola

Crea el archivo “hola” dentro de /home/curso/borrar.

TOUCH(1) NAME touch - change file timestamps

SYNOPSIS touch [OPTION]... FILE...

DESCRIPTION Update the access and modification times of each FILE to the current time. A FILE argument that does not exist is created empty, unless `-c` or `-h` is supplied.

2. curso@atlas: ~/borrar\$ ls -l

total 0


```
-rw-r-r- 1 curso users 0 mar 26 23:45 hola
```

Se lista el contenido del directorio /home/user/borrar

LS(1)

NAME ls - list directory contents

SYNOPSIS ls [OPTION]... [FILE]...

DESCRIPTION List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified. Mandatory arguments to long options are mandatory for short options too.

-l use a long listing format

3. curso@atlas: /borrar\$ chmod u-w .

Se cambian los permisos del directorio actual /home/user/borrar. Se quita el permiso de escritura para el user actual (curso) el cual ya no podrá escribir (write) un cambio en este directorio.

CHMOD(1)

NAME chmod - change file mode bits

SYNOPSIS chmod [OPTION]... MODE[,MODE]... FILE... chmod [OPTION]...

OCTAL-MODE FILE... chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION This manual page documents the GNU version of chmod. chmod changes the file mode bits of each given file according to mode, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

The format of a symbolic mode is [ugoa...][[+|=][perms...]...], where perms is either zero or more letters from the set rwxXst, or a single letter from the set ugo. Multiple symbolic modes can be given, separated by commas.

4. curso@atlas: /borrar\$ touch chau

```
touch: no se puede efectuar 'touch' sobre «chau»: Permiso denegado
```

Se intenta crear el archivo “chau” dentro del directorio /home/curso/borrar lo cual no será posible ya que no disponemos más de write permises.

5. curso@atlas: /borrar\$ touch hola

Se reailza “touch hola” el cual no aplica ningún cambio evidente ya que el archivo “hola” ya había sido creado en el paso 1).

6. curso@atlas: /borrar\$ rm hola

rm: no se puede borrar «hola»: Permiso denegado

Se intenta borrar el archivo “hola” dentro del directorio /home/curso/-borrar lo cual no será posible ya que no disponemos más de write permises.

3. Ejercicios Tema 4: Línea de comandos. Comandos básicos II

- Búsqueda de archivos: locate, find.
- Obtener información de uso de programas: man.
- Buscar expresiones: grep.
- Monitorear el uso de espacio: du, df.
- Archivar y comprimir archivos: tar, gzip, bzip2.

3.1. Actividad 1.

3.1.1. “find” and “locate” commands

Si comparamos el comando “find” y “locate” ¿Cuál posee mayor velocidad de respuesta? ¿Por qué?

Find Files Using Locate An alternative to using **emphfind** is the **emphlocate** command. This command is often quicker and can search the entire file system with ease.

You can install the command with apt-get:

```
sudo apt-get update
sudo apt-get install mlocate
```

The reason locate is faster than find is because it relies on a database of the files on the filesystem.

The database is usually updated once a day with a cron script, but you can update it manually by typing:

```
sudo updatedb
```

Run this command now. Remember, the database must always be up-to-date if you want to find recently acquired or created files.

[DigitalOcean™ Inc., 2015]

3.2. Actividad 2.

3.2.1. locate -d command

Si hacemos uso del comando locate ¿Qué significa la opción -d? Actualice la base de datos de locate, y busque todos los archivos cuyo nombre contenga

la palabra bash dentro del directorio /etc.

La opción -d se utiliza según se especifica debajo:

-d, --database DBPATH Replace the default database with DBPATH. DBPATH is a :-separated list of database file names. If more than one --database option is specified, the resulting path is a concatenation of the separate paths.

```
curso@ubuntu:~$ sudo updatedb
curso@ubuntu:~$ locate '/etc/*bash*'
/etc/bash.bashrc
/etc/bash_completion
/etc/bash_completion.d
/etc/apparmor.d/abstractions/bash
/etc/bash_completion.d/apport_completion
/etc/bash_completion.d/axi-cache
/etc/bash_completion.d/debconf
/etc/bash_completion.d/desktop-file-validate
/etc/bash_completion.d/grub
/etc/bash_completion.d/initramfs-tools
/etc/bash_completion.d/insserv
/etc/bash_completion.d/m-a
/etc/bash_completion.d/pon
/etc/bash_completion.d/pulseaudio-bash-completion.sh
/etc/bash_completion.d/ufw
/etc/bash_completion.d/upstart
/etc/profile.d/bash_completion.sh
/etc/skel/.bash_logout
/etc/skel/.bashrc
```

locate(1)

NAME locate - find files by name

SYNOPSIS locate [OPTION]... PATTERN...

DESCRIPTION locate reads one or more databases prepared by updatedb(8) and writes file names matching at least one of the PATTERNS to standard output, one per line.

If --regex is not specified, PATTERNS can contain globbing characters. If any PATTERN contains no globbing characters, locate behaves as if the pattern were *PATTERN*.

By default, locate does not check whether files found in database still exist (but it does require all parent directories to exist if the database was built with --require-visibility no). locate can never report files created after the most recent update of the relevant database.

EXIT STATUS locate exits with status 0 if any match was found or if locate was invoked with one of the --limit 0, --help, --statistics or --version options. If no match was found or a fatal error was encountered, locate exits with status 1.

Errors encountered while reading a database are not fatal, search continues in other specified databases, if any.

OPTIONS -A, --all Print only entries that match all PATTERNS instead of requiring only one of them to match.

-b, --basename Match only the base name against the specified patterns. This is the opposite of --wholename.

-c, --count Instead of writing file names on standard output, write the number of matching entries only.

-d, --database DBPATH Replace the default database with DBPATH. DBPATH is a :-separated list of database file names. If more than one --database option is specified, the resulting path is a concatenation of the separate paths.

3.3. Actividad 3.

3.3.1. “find” without name

¿Cómo haría uso del comando find si desea buscar una fotografía (archivo PNG) y no recuerda su nombre sino solo que se encuentra en su home.

```
curso@ubuntu:~$ find /home/curso/ -iname '*.PNG'

/home/curso/picture.PNG
/home/curso/BarrireroExequiel/TP1/Imagenes/Actividad12.png
/home/curso/BarrireroExequiel/TP1/Imagenes/Actividad13.png
/home/curso/BarrireroExequiel/TP1/Imagenes/Actividad2.png
/home/curso/BarrireroExequiel/TP1/Imagenes/Actividad10.png
/home/curso/BarrireroExequiel/TP1/Imagenes/Actividad11.png
/home/curso/BarrireroExequiel/TP1/Imagenes/Actividad5.png
/home/curso/BarrireroExequiel/TP1/Imagenes/Actividad14.png
/home/curso/.local/share/Trash/files/Actividad5.png
/home/curso/.cache/software-center/icons/tomato:i386-icon-Icon64.png
/home/curso/.cache/software-center/icons/umamu-icon-Umamu_r_64.png
/home/curso/.cache/software-center/icons/capsized-icon-Icon64.png
/home/curso/.cache/software-center/icons/2048:i386-icon-64-2048.png
/home/curso/.cache/software-center/icons/tic-tac-toe2:i386-icon-64_hmojXQC.png
/home/curso/.cache/software-center/icons/mycraft-icon-mc-launcher.svg64.png
/home/curso/.cache/software-center/icons/audovia:i386-icon-SongBuilderColourIcon64.png
/home/curso/.cache/software-center/icons/mc2048-icon-mc2048_1.png
/home/curso/.cache/software-center/icons/my-weather-indicator:i386-icon-mwi_064.png
/home/curso/.cache/thumbnails/large/be9745aaa4dd17df931fd3f4b1b37b74.png
/home/curso/.cache/thumbnails/large/0f635725858090944921b5ae0e1c25ae.png
...
/home/curso/.cache/thumbnails/normal/f458b2702111282ba10cca5ddd511daf.png
/home/curso/.cache/thumbnails/normal/ee9cf2ab26402213d518127e9b5b1149.png
/home/curso/.cache/thumbnails/normal/b2bdf6f2a85194544d0558983df41eb0.png
/home/curso/.cache/thumbnails/normal/76fbdec1d2173df80d31f3d4a2ca4e3.png
```

3.4. Actividad 4.

3.4.1. “find” buscando por permisos de usuario

Haciendo uso del comando find, ¿Cuáles son los archivos o directorios que poseen permisos de escritura para cualquier usuario?. Agregue una opción para que no incluya los enlaces simbólicos sino los archivos apuntados por los enlaces.

Finding by Type

You can specify the type of files you want to find with the `type` parameter. It works like this:

`find -type type_descriptor query` Some of the most common descriptors that you can use to specify the type of file are here:

- `f`: regular file
- `d`: directory
- `l`: symbolic link
- `c`: character devices
- `b`: block devices

[DigitalOcean™ Inc., 2015]

Finding files by permission

Searching for files by permission is an excellent way to turn up security issues on your system or uncover access issues. Just as you changed permissions on files using numbers or letters (with the `chmod` command), you can likewise find files based on number or letter permissions along with the `-perm` options. (Refer to Chapter 4, “Moving around the Filesystem,” to see how to use numbers and letters with `chmod` to reflect file permissions.) If you use numbers for permission, as I do below, remember that the three numbers represent permissions for the user, group, and other. Each of those three numbers varies from no permission (0) to full read/write/execute permission (7), by adding read (4), write (2), and execute (1) bits together. With a hyphen (-) in front of the number, all three of the bits indicated must match; with a plus (+) in front of it, any of the numbers can match for the search to find a file. The full, exact numbers must match if neither a hyphen or plus is used.

Consider the following examples:

```
$ find /bin -perm 755 -ls
788884 28 -rwxr-xr-x 1 root root 28176 Mar 10 2014 /bin/echo
$ find /home/chris/ -perm -222 -type d -ls
144503 4 drwxrwxrwx 8 chris chris 4096 Jun 23 2014 /home/chris
```

By searching for `-perm 755`, any files or directories with exactly `rwxr-xr-x` permission are matched. By using `-perm -222`, only files that have write permission for user, group, and other are matched. Notice that, in this case, the `-type d` is added to match only directories.

```
$ find /myreadonly -perm +222 -type f
```

```
685035 0 -rw-rw-r- 1 chris chris 0 Dec 30 2014 /tmp/write/abc
$ find . -perm -002 -type f -ls
266230 0 -rw-rw-rw- 1 chris chris 0 Dec 20 2014 ./LINUX_BIBLE/a
```

Using `-perm +222`, you can find any file (-type f) that has write permission turned on for the user, group, or other. You might do that to make sure that all files are read-only in a particular part of the filesystem (in this case, beneath the /myread-only directory). The last example, `-perm +002`, is very useful for finding files that have open write permission for “other,” regardless of how the other permission bits are set.

[Negus, 2015]

```
curso@ubuntu:~$ sudo find / -type f -perm +002 -ls | less
```

```
23477 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/sys/kernel/ns_last_pid
24246 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/task/1/attr/current
24248 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/task/1/attr/exec
24249 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/task/1/attr/fscreate
24250 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/task/1/attr/keycreate
24251 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/task/1/attr/socketcreate
24337 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/attr/current
24339 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/attr/exec
24340 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/attr/fscreate
24341 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/attr/keycreate
24342 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/1/attr/socketcreate
24433 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/task/2/attr/current
24435 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/task/2/attr/exec
24436 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/task/2/attr/fscreate
24437 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/task/2/attr/keycreate
24438 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/task/2/attr/socketcreate
24445 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/attr/current
24447 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/attr/exec
24448 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/attr/fscreate
24449 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/attr/keycreate
24450 0 -rw-rw-rw- 1 root root 0 Oct 3 21:16 /proc/2/attr/socketcreate
```

```
curso@ubuntu:~$ sudo find / -type d -perm +002 -ls
532295 4 drwxrwxrwt 2 root root 4096 Sep 29 18:57 /var/crash
525560 4 drwxrwxrwt 2 root root 4096 Sep 29 21:02 /var/tmp
find: '/proc/2960/task/2960/fd/5': No such file or directory
find: '/proc/2960/task/2960/fdinfo/5': No such file or directory
find: '/proc/2960/fd/5': No such file or directory
find: '/proc/2960/fdinfo/5': No such file or directory
8715 0 drwxrwxrwt 2 root root 140 Oct 3 20:45 /run/shm
8712 0 drwxrwxrwt 2 root root 40 Oct 3 15:01 /run/lock
131090 4 drwxrwxrwt 4 root root 4096 Oct 3 21:17 /tmp
156905 4 drwxrwxrwt 2 root root 4096 Oct 3 16:25 /tmp/.ICE-unix
156904 4 drwxrwxrwt 2 root root 4096 Oct 3 15:00 /tmp/.X11-unix
```

3.5. Actividad 5.

3.5.1. “find” para archivos modificados en un período de tiempo

Haciendo uso del comando `find` busque los archivos que han sido modificados en los últimos 7 días.

Finding by Time

Linux stores time data about access times, modification times, and change times.

- **Access Time:** Last time a file was read or written to.
- **Modification Time:** Last time the contents of the file were modified.
- **Change Time:** Last time the file's inode meta-data was changed.

We can use these with the “*-atime*”, “*-mtime*”, and “*-ctime*” parameters. These can use the plus and minus symbols to specify greater than or less than, like we did with size.

The value of this parameter specifies how many days ago you'd like to search. To find files that have a modification time of a day ago, type:

```
find / -mtime 1
```

If we want files that were accessed in less than a day ago, we can type:

```
find / -atime -1
```

To get files that last had their meta information changed more than 3 days ago, type:

```
find / -ctime +3
```

There are also some companion parameters we can use to specify minutes instead of days:

```
find / -mmin -1
```

This will give the files that have been modified type the system in the last minute. Find can also do comparisons against a reference file and return those that are newer:

```
find / -newer myfile
```

[DigitalOcean™ Inc., 2015]

```
curso@ubuntu:~$ find /home/curso/ -mtime -7
...
/home/curso/.cache/oneconf/cf31d3950d1cc51f862b52e955f9872b
/home/curso/.cache/oneconf/cf31d3950d1cc51f862b52e955f9872b/host
/home/curso/.vboxclient-display.pid
/home/curso/.gconf
/home/curso/.gconf/apps/guake/style/background
/home/curso/.gconf/apps/guake/style/background/%gconf.xml
/home/curso/.gconf/apps/guake/style/font
/home/curso/.gconf/apps/guake/style/font/%gconf.xml
/home/curso/.gconf/apps/guake/general
/home/curso/.gconf/apps/guake/general/%gconf.xml
```



```

curso@ubuntu:~$ find /home/curso/ -mtime -7 -ls
...
532723      4 -rwxrw----    1 curso      curso  198 Sep 29 19:24 /home/curso/.cache/one...
533520      4 -rwxr-----    1 curso      curso   5 Oct  3 16:25 /home/curso/.vboxclient-...
565991      4 drwx-----    4 curso      curso 4096 Oct  3 16:25 /home/curso/.gconf...
566119      4 drwx-----    2 curso      curso 4096 Oct  3 16:28 /home/curso/.gconf/ap...
566211      4 -rwx-----    1 curso      curso  110 Oct  3 16:28 /home/curso/.gconf/app...
566121      4 drwx-----    2 curso      curso 4096 Oct  3 16:28 /home/curso/.gconf/ap...
569336      4 -rwx-----    1 curso      curso  467 Oct  3 16:28 /home/curso/.gconf/app...
566123      4 drwx-----    2 curso      curso 4096 Oct  3 16:28 /home/curso/.gconf/ap...
569337      4 -rwx-----    1 curso      curso  954 Oct  3 16:28 /home/curso/.gconf/app...

```

3.6. Actividad 6.

3.6.1. “grep” command

Haciendo uso del comando grep buscar la palabra ”dui” en el archivo /etc/group sin distinguir entre mayúsculas y minúsculas. Luego haciendo uso del mismo comando contar el número de ocurrencias en que aparece esa cadena si se hace o no distinción entre mayúsculas y minúsculas.

GREP(1) General Commands Manual GREP(1)

NAME grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS grep [OPTIONS] PATTERN [FILE...] grep [OPTIONS] [-e PATTERN | -f FILE] [FILE...]

DESCRIPTION grep searches the named input FILES (or standard input if no files are named, or if a single hyphen-minus (-) is given as file name) for lines containing a match to the given PATTERN. By default, grep prints the matching lines.

In addition, three variant programs egrep, fgrep and rgrep are available. egrep is the same as grep -E. fgrep is the same as grep -F. rgrep is the same as grep -r. Direct invocation as either egrep or fgrep is deprecated, but is provided to allow historical applications that rely on them to run unmodified.

-i, --ignore-case Ignore case distinctions in both the PATTERN and the input files. (-i is specified by POSIX.)

General Output Control -c, --count Suppress normal output; instead print a count of matching lines for each input file. With the -v, --invert-match option (see below), count non-matching lines. (-c is specified by POSIX.)

```

curso@ubuntu:~$ cat /etc/group | grep -i dui
curso@ubuntu:~$ cat /etc/group | grep -i curso
sudo:x:27:delivery,curso
curso:x:1001:curso
grupol:x:1002:curso
curso@ubuntu:~$ cat /etc/group | grep -i CURSO
sudo:x:27:delivery,curso
curso:x:1001:curso
grupol:x:1002:curso

```

```
curso@ubuntu:~$ cat /etc/group | grep -c -i curso
3
```

3.7. Actividad 7.

3.7.1. "grep" recursivo

Haciendo uso del comando grep buscar la cadena `.auto` en los archivos del directorio `/etc` y sus subdirectorios..

```
curso@ubuntu:/$ ls -R /etc/ | grep "auto"
01autoremove
01autoremove-kernels
ls: cannot open directory /etc/chatscripts: Permission denied
ls: cannot open directory /etc/cups/ssl: Permission denied
10-autohint.conf
apt-auto-removal
lightdm-autologin
ls: cannot open directory /etc/polkit-1/localauthority: Permission denied
ls: cannot open directory /etc/ppp/peers: Permission denied
ls: cannot open directory /etc/ssl/private: Permission denied
autostart
/etc/xdg/autostart:
nautilus-autostart.desktop
```

3.8. Actividad 8.

3.8.1. "df" commands

¿Con que comando puedo ver el porcentaje de uso de los dispositivos de bloques montados en human-readable?

DF(1) User Commands DF(1)
NAME df - report file system disk space usage
SYNOPSIS df [OPTION]... [FILE]...

DESCRIPTION This manual page documents the GNU version of df. df displays the amount of disk space available on the file system containing each file name argu[U+2010]ment. If no file name is given, the space available on all currently mounted file systems is shown. Disk space is shown in 1K blocks by default, unless the environment variable POSIXLY_CORRECT is set, in which case 512-byte blocks are used.

If an argument is the absolute file name of a disk device node containing a mounted file system, df shows the space available on that file system rather than on the file system containing the device node (which is always the root file system). This version of df cannot show the space available on unmounted file systems, because on most kinds of systems doing so requires very nonportable intimate knowledge of file system structures.

OPTIONS Show information about the file system on which each FILE resides, or all file systems by default.

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

-a, --all include dummy file systems

-B, --block-size=SIZE scale sizes by SIZE before printing them. E.g., '-BM' prints sizes in units of 1,048,576 bytes. See SIZE format below.

--total produce a grand total

-h, --human-readable print sizes in human readable format (e.g., 1K 234M 2G)

-T, --print-type print file system type

```
curso@ubuntu:/$ df -h -T
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/sda1       ext4      9.8G  3.2G  6.1G  34% /
none            tmpfs     4.0K   0    4.0K   0% /sys/fs/cgroup
udev            devtmpfs  991M  4.0K  991M   1% /dev
tmpfs           tmpfs     201M  480K  200M   1% /run
none            tmpfs     5.0M   0    5.0M   0% /run/lock
none            tmpfs    1001M  152K  1001M   1% /run/shm
none            tmpfs     100M   40K   100M   1% /run/user
/dev/sr0        iso9660    56M   56M    0 100% /media/curso/VBOXADDITIONS_4.3.26_98988
```

3.9. Actividad 9.

3.9.1. "tar.gz" command p/ comprimir

Comprima el directorio /etc con tar.gz y guardelo en el directorio /home/-curso.

```
curso@ubuntu:~$ tar -zcvf /home/curso/etc.tar.gz /etc/
```

```
curso@ubuntu:/$ cd /home/curso
```

```
curso@ubuntu:~$ ls
```

```
BarrireroExequiel BarrireroExequiel.tar.gz Desktop Documents Downloads etc.tar.gz group Mus
```

```
curso@ubuntu:~$ ls -la | grep 'tar.gz'
```

```
-rwxrw---- 1 curso curso 2013003 Sep 21 21:13 BarrireroExequiel.tar.gz
```

```
-rw-rw-r-- 1 curso curso 718191 Oct 3 22:33 etc.tar.gz
```

You need to use the tar command as follows (syntax of tar command):

```
tar -zcvf archive-name.tar.gz directory-name
```

Where,

-z : Compress archive using gzip program

-c : Create archive

-v : Verbose i.e display progress while creating archive

-f : Archive File name

[Vivek G., 2006]

3.10. Actividad 10.

3.10.1. "tar.gz" command p/ descomprimir

Descomprima el archivo .tar.gz creado en la Actividad 9 en el directorio /home/curso/Desktop/root-etc.

-x: Extract files

If you wish to extract files in particular directory, for example in /tmp then you need to use the following command:

```
$ tar -zxvf prog-1-jan-2005.tar.gz -C /tmp
$ cd /tmp
$ ls -
```

[Vivek G., 2006]

```
mkdir /home/curso/Desktop/root-etc
tar -xvzf etc.tar.gz -C /home/curso/Desktop/root-etc

curso@ubuntu:~$ cd /home/curso/Desktop/root-etc/
curso@ubuntu:~/Desktop/root-etc$ ls
etc

curso@ubuntu:~/Desktop/root-etc$ cd etc/
curso@ubuntu:~/Desktop/root-etc/etc$ ls
acpi                                cron.d                             gnome-app-install                 kernel-img.conf
adduser.conf                       cron.daily                         gnome-settings-daemon            landscape
alternatives                      cron.hourly                       gnome-vfs-2.0                    ldap
anacrontab                        cron.monthly                      groff                             ld.so.cache
apg.conf                          crontab                           group                             ld.so.conf
apm                               cron.weekly                       grub.d                           ld.so.conf.d
apparmor                          cups                              gtk-2.0                          legal
apparmor.d                        cupshelpers                       gtk-3.0                          libaudit.conf
appport                          dbus-1                            hdparm.conf                      libnl-3
apt                               dconf                             hostname                         libpaper.d
at-spi2                           debconf.conf                     hostname                         lightdm
avahi                             debian_version                   hosts                             lintianrc
bash.bashrc                       default                           hosts.allow                      locale.alias
bash_completion                  deluser.conf                     hosts.deny                       localtime
bash_completion.d                depmod.d                         ifplugd                          logcheck
bindresvport.blacklist           dhcp                              ImageMagick                      login.defs
blkid.conf                       dictionaries-common              init                             logrotate.conf
blkid.tab                         doc-base                         init.d                           logrotate.d
bluetooth                        dpkg                             initramfs-tools                  lsb-release
bonobo-activation                drirc                            inputrc                          ltrace.conf
byobu                            emacs                             insserv                          magic
ca-certificates                  environment                      insserv.conf                    magic.mime
ca-certificates.conf            fonts                            insserv.conf.d                  mailcap
calendar                        fstab                             iproute2                        mailcap.order
chromium-browser                 fstab.d                          iscsi                           manpath.config
colord.conf                      gai.conf                         issue                            mime.types
compizconfig                     gconf                           issue.net                       mke2fs.conf
console-setup                   ghostscript                      kbd                             modprobe.d
cracklib                         gnome                            kernel                           modules
```

4. Ejercicios Tema 6: Uso básico de la consola

- Gnome-terminal.
- Shortcuts de líneas de comando.
 - Expansión de expresiones con wildcards.
 - Tecla “tabulación”.
 - Comando “history” y “Ctrl - r”.
- Expansión de líneas de comando
 - El tilde: .
 - Expansión de líneas de comando: \$() o “.
 - Llaves de expansión: .
- Trucos en las líneas de comando: Ctrl - a, Ctrl - e, Ctrl - u, Ctrl - k, Ctrl - flechas de dirección.

4.1. Actividad 1.

4.1.1. Regular expressions - ^, \$, ., *

Cree una expresión que corresponda con todos los archivos que comiencen con una 'a' y terminen con una 'z'.

An expression is a string of characters. Those characters having an interpretation above and beyond their literal meaning are called metacharacters. A quote symbol, for example, may denote speech by a person, ditto, or a meta-meaning for the symbols that follow. Regular Expressions are sets of characters and/or metacharacters that match (or specify) patterns.

A Regular Expression contains one or more of the following:

- **A character set:** These are the characters retaining their literal meaning. The simplest type of Regular Expression consists only of a character set, with no metacharacters.
- **An anchor:** These designate (anchor) the position in the line of text that the RE is to match. For example, ^, and \$ are anchors.
- **Modifiers:** These expand or narrow (modify) the range of text the RE is to match. Modifiers include the asterisk, brackets, and the backslash.

The main uses for Regular Expressions (REs) are text searches and string manipulation. An RE matches a single character or a set of characters – a string or a part of a string.

- **The asterisk – * – :** matches any number of repeats of the character string or RE preceding it, including zero instances.

"1133*" matches 11 + one or more 3's: 113, 1133, 1133333, and so forth.

Example: Match all files which have a word twt, twet, tweet etc in the file name.

```
ls -l | grep 'twe*t'
```

Example: How about searching for apple word which was spelled wrong in a given file where apple is misspelled as ale, aple, appple, apppple, apppppple etc. To find all patterns

```
grep 'ap*le' filename
```

Readers should observe that the above pattern will match even ale word as * indicates 0 or more of previous character occurrence.

- **The dot – . – :** matches any one character, except a newline.

"13." matches 13 + at least one of any character (including a space): 1133, 11333, but not 13 (additional character missing).

- **The caret – ^ – :** matches the beginning of a line, but sometimes, depending on context, negates the meaning of a set of characters in an RE.

- **The dollar sign – \$ – :** at the end of an RE matches the end of a line.

"XXX\$" matches XXX at the end of a line.

"^\$" matches blank lines.

[Hogbin E., 2005a]

```
curso@ubuntu:~$ pwd
/home/curso
```

```
curso@ubuntu:~$ ls -la
total 2800
drwxr-x--x 17 curso curso 4096 Nov  1 16:28 .
drwxrwx--x  4 root root  4096 Sep 20 22:31 ..
-rw-rw-r--  1 curso curso    0 Nov  1 16:28 a123z
-rw-rw-r--  1 curso curso    0 Nov  1 16:28 abz
-rw-rw-r--  1 curso curso    0 Nov  1 16:28 axi123z
-rw-rw-r--  1 curso curso    0 Nov  1 16:28 aaz
-rw-rw-r--  1 curso curso    0 Nov  1 16:28 az
-rw-rw-r--  1 curso curso    0 Nov  1 16:28 azz
-rw-rw-r--  1 curso curso    0 Nov  1 16:28 a.z
drwxrwx--x  3 curso curso  4096 Sep 21 19:25 BarrireroExequiel
-rwxrw----  1 curso curso 2013003 Sep 21 21:13 BarrireroExequiel.tar.gz
-rwx-----  1 curso curso  5857 Oct  3 23:19 .bash_history
-rwxr-----  1 curso curso   220 Sep 20 19:49 .bash_logout
-rwxr-----  1 curso curso  3637 Sep 20 19:49 .bashrc
drwx----- 17 curso curso  4096 Sep 29 18:53 .cache
drwx----- 22 curso curso  4096 Sep 29 19:12 .config
drwxr-x--x  3 curso curso  4096 Oct  3 22:41 Desktop
-rwxr-----  1 curso curso    33 Sep 21 19:15 .dmrc
```

```

drwxr-x--x  2 curso curso      4096 Sep 21 14:06 Documents
drwxr-x--x  2 curso curso      4096 Sep 21 19:30 Downloads
-rw-rw-r--  1 curso curso 718191 Oct  3 22:33 etc.tar.gz
drwx----- 4 curso curso      4096 Nov  1 14:44 .gconf
lrwxrwxrwx  1 curso curso        10 Sep 29 19:55 group -> /etc/group
-rwx----- 1 curso curso     2544 Nov  1 14:44 .ICEauthority
-rw----- 1 curso curso        42 Oct  3 22:18 .lessht
drwx----- 3 curso curso      4096 Sep 21 14:06 .local
drwxr-x--x  2 curso curso      4096 Sep 21 14:06 Music
-rw-rw-r--  1 curso curso        0 Oct  3 20:53 picture.PNG
drwxr-x--x  2 curso curso      4096 Sep 21 20:53 Pictures
drwx----- 3 curso curso      4096 Sep 21 19:26 .pki
-rwxr----- 1 curso curso       675 Sep 20 19:49 .profile
drwxr-x--x  2 curso curso      4096 Sep 21 14:06 Public
drwxrwx--x  5 curso curso      4096 Oct  3 20:57 root
drwxr-x--x  2 curso curso      4096 Sep 21 14:06 Templates

curso@ubuntu:~/BarrireroExequiel$ ls | grep '^a.*z$'
a123z
aaz
abz
axil23z
az
a.z
azz

curso@ubuntu:~$

```

Puede notarse la diferencia al utilizar la expresión debajo sin incluir el “^”, wildcard: cualquier único carácter. El cual quedo desmotrado que al acompañar a la expresión “*” permite buscar la coincidencia de una expresión que comience con “a” luego coincida con cero o más del carácter previo y termine con “z”. Permitiendo este carácter identificado con “.” lograr el resultado que estamos buscando.

```

curso@ubuntu:~/BarrireroExequiel$ ls | grep '^a*z$'
aaz
az

```

4.2. Actividad 2.

4.2.1. Regular expressions - ^, \$, ., *

Cree una expresión que corresponda con todos los archivos cuyo nombre tenga como segunda letra una 't'. Para realizar la actividad aprender el uso del comodín \$

- **The backslash – \ :** Escapes a special character, which means that character gets interpreted literally (and is therefore no longer special).
- **An anchor:** A “\$” reverts back to its literal meaning of “\$”, rather than its RE meaning of end-of-line. Likewise a “\” has the literal meaning of “\”.

- **Escaped “angle brackets” – \<...\> :** Escaped “angle brackets” – \<...\> mark word boundaries. The angle brackets must be escaped, since otherwise they have only their literal character meaning. "\<the\>" matches the word “the,” but not the words “them,” “there,” “other,” etc.

[Hogbin E., 2005a]

```
curso@ubuntu:~$ pwd
/home/curso

curso@ubuntu:~$ ls -la
total 2800
drwxr-x--x 17 curso curso 4096 Nov  1 16:28 .
drwxrwx--x  4 root  root 4096 Sep 20 22:31 ..
-rw-rw-r--  1 curso curso  0 Nov  1 16:28 al23z
-rw-rw-r--  1 curso curso  0 Nov  1 16:28 abz
-rw-rw-r--  1 curso curso  0 Nov  1 16:28 axil23z
-rw-rw-r--  1 curso curso  0 Nov  1 16:28 az
-rw-rw-r--  1 curso curso  0 Nov  1 16:28 a.z
drwxrwx--x  3 curso curso 4096 Sep 21 19:25 BarrireroExequiel
-rwxr----- 1 curso curso 2013003 Sep 21 21:13 BarrireroExequiel.tar.gz
-rwx----- 1 curso curso 5857 Oct  3 23:19 .bash_history
-rwxr----- 1 curso curso  220 Sep 20 19:49 .bash_logout
-rwxr----- 1 curso curso 3637 Sep 20 19:49 .bashrc
drwx----- 17 curso curso 4096 Sep 29 18:53 .cache
drwx----- 22 curso curso 4096 Sep 29 19:12 .config
drwxr-x--x  3 curso curso 4096 Oct  3 22:41 Desktop
-rwxr----- 1 curso curso  33 Sep 21 19:15 .dmrc
drwxr-x--x  2 curso curso 4096 Sep 21 14:06 Documents
drwxr-x--x  2 curso curso 4096 Sep 21 19:30 Downloads
-rw-rw-r--  1 curso curso 718191 Oct  3 22:33 etc.tar.gz
drwx----- 4 curso curso 4096 Nov  1 14:44 .gconf
lrwxrwxrwx  1 curso curso  10 Sep 29 19:55 group -> /etc/group
-rwx----- 1 curso curso 2544 Nov  1 14:44 .ICEauthority
-rw-----  1 curso curso  42 Oct  3 22:18 .lessht
drwx----- 3 curso curso 4096 Sep 21 14:06 .local
drwxr-x--x  2 curso curso 4096 Sep 21 14:06 Music
-rw-rw-r--  1 curso curso  0 Oct  3 20:53 picture.PNG
drwxr-x--x  2 curso curso 4096 Sep 21 20:53 Pictures
drwx----- 3 curso curso 4096 Sep 21 19:26 .pki
-rwxr----- 1 curso curso  675 Sep 20 19:49 .profile
drwxr-x--x  2 curso curso 4096 Sep 21 14:06 Public
drwxrwx--x  5 curso curso 4096 Oct  3 20:57 root
drwxr-x--x  2 curso curso 4096 Sep 21 14:06 Templates

curso@ubuntu:~$ ls /home/curso | grep '^\.t'
etc.tar.gz
curso@ubuntu:~$
```

4.3. Actividad 3.

4.3.1. Regular expressions - ^, ., []

Cree una expresión que corresponda con todos los archivos cuyo nombre tenga como tercer letra una 'letra mayúscula' o una 'coma'.

- **Brackets – [...] – :** enclose a set of characters to match in a single RE. which means that character gets interpreted literally (and is therefore no longer special).

"[xyz]": matches any one of the characters x, y, or z.

"[c-n]": matches any one of the characters in the range c to n.

"[B-Pk-y]": matches any one of the characters in the ranges B to P and k to y.

"[a-z0-9]": matches any single lowercase letter or any digit.

"[^b-d]": matches any character except those in the range b to d. This is an instance of ^ negating or inverting the meaning of the following RE (taking on a role similar to ! in a different context).

Combined sequences of bracketed characters match common word patterns.

"[Yy][Ee][Ss]" matches yes, Yes, YES, yEs, and so forth. "[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9]" matches any Social Security number.

[Hogbin E., 2005a]

```
curso@ubuntu:~$ pwd
/home/curso
```

```
curso@ubuntu:~$ ls -la
total 2800
drwxr-x--x 17 curso curso 4096 Nov 1 16:28 .
drwxrwx--x 4 root root 4096 Sep 20 22:31 ..
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 a123z
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 abz
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 axi123z
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 az
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 a.z
drwxrwx--x 3 curso curso 4096 Sep 21 19:25 BarrireroExequiel
-rwxrwx---- 1 curso curso 2013003 Sep 21 21:13 BarrireroExequiel.tar.gz
-rwx----- 1 curso curso 5857 Oct 3 23:19 .bash_history
-rwxr----- 1 curso curso 220 Sep 20 19:49 .bash_logout
-rwxr----- 1 curso curso 3637 Sep 20 19:49 .bashrc
drwx----- 17 curso curso 4096 Sep 29 18:53 .cache
drwx----- 22 curso curso 4096 Sep 29 19:12 .config
drwxr-x--x 3 curso curso 4096 Oct 3 22:41 Desktop
-rwxr----- 1 curso curso 33 Sep 21 19:15 .dmrc
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Documents
drwxr-x--x 2 curso curso 4096 Sep 21 19:30 Downloads
-rw-rw-r-- 1 curso curso 718191 Oct 3 22:33 etc.tar.gz
drwx----- 4 curso curso 4096 Nov 1 14:44 .gconf
lrwxrwxrwx 1 curso curso 10 Sep 29 19:55 group -> /etc/group
-rwx----- 1 curso curso 2544 Nov 1 14:44 .ICEauthority
-rw----- 1 curso curso 42 Oct 3 22:18 .lessht
drwx----- 3 curso curso 4096 Sep 21 14:06 .local
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Music
-rw-rw-r-- 1 curso curso 0 Oct 3 20:53 picture.PNG
drwxr-x--x 2 curso curso 4096 Sep 21 20:53 Pictures
drwx----- 3 curso curso 4096 Sep 21 19:26 .pki
-rwxr----- 1 curso curso 675 Sep 20 19:49 .profile
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Public
drwxrwx--x 5 curso curso 4096 Oct 3 20:57 root
```

```
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Templates
```

```
curso@ubuntu:~$ ls | grep '^..[A-Z ,]'
```

```
curso@ubuntu:~$
```

Realizando una búsqueda con una expresión menos restrictiva, es decir quitando el anclado que indica que la expresión debe comenzar con dos caracteres cualquier y luego buscar coincidencia a partir del 3ero obtenemos algunos resultados. El enuncia podría cambiar por “nombre de archivo que posea 2 caracteres cualquiera antes de una mayúscula o una coma”.

```
curso@ubuntu:~$ ls | grep '..[A-Z ,]'
```

```
BarrireroExequiel
```

```
BarrireroExequiel.tar.gz
```

```
picture.PNG
```

4.4. Actividad 4.

4.4.1. Regular expressions - ^, \$, []

Cree una expresión que corresponda con todos los archivos que no terminen con una letra de la 'a a la d' ó de la 'f a la z'.

```
curso@ubuntu:~$ pwd
```

```
/home/curso
```



```
curso@ubuntu:~$ ls -la
```

```
total 2800
```

```
drwxr-x--x 17 curso curso 4096 Nov 1 16:28 .
```

```
drwxrwx--x 4 root root 4096 Sep 20 22:31 ..
```

```
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 a123z
```

```
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 abz
```

```
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 axil23z
```

```
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 az
```

```
-rw-rw-r-- 1 curso curso 0 Nov 1 16:28 a.z
```

```
drwxrwx--x 3 curso curso 4096 Sep 21 19:25 BarrireroExequiel
```

```
-rwxrw---- 1 curso curso 2013003 Sep 21 21:13 BarrireroExequiel.tar.gz
```

```
-rwx----- 1 curso curso 5857 Oct 3 23:19 .bash_history
```

```
-rwxr----- 1 curso curso 220 Sep 20 19:49 .bash_logout
```

```
-rwxr----- 1 curso curso 3637 Sep 20 19:49 .bashrc
```

```
drwx----- 17 curso curso 4096 Sep 29 18:53 .cache
```

```
drwx----- 22 curso curso 4096 Sep 29 19:12 .config
```

```
drwxr-x--x 3 curso curso 4096 Oct 3 22:41 Desktop
```

```
-rwxr----- 1 curso curso 33 Sep 21 19:15 .dmrc
```

```
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Documents
```

```
drwxr-x--x 2 curso curso 4096 Sep 21 19:30 Downloads
```

```
-rw-rw-r-- 1 curso curso 718191 Oct 3 22:33 etc.tar.gz
```

```
drwx----- 4 curso curso 4096 Nov 1 14:44 .gconf
```

```
lrwxrwxrwx 1 curso curso 10 Sep 29 19:55 group -> /etc/group
```

```
-rwx----- 1 curso curso 2544 Nov 1 14:44 .ICEauthority
```

```
-rw----- 1 curso curso 42 Oct 3 22:18 .lessht
```

```
drwx----- 3 curso curso 4096 Sep 21 14:06 .local
```

```
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Music
```

```
-rw-rw-r-- 1 curso curso 0 Oct 3 20:53 picture.PNG
```

```
drwxr-x--x 2 curso curso 4096 Sep 21 20:53 Pictures
```

```
drwx----- 3 curso curso 4096 Sep 21 19:26 .pki
```

```
-rwxr----- 1 curso curso      675 Sep 20 19:49 .profile
drwxr-x--x  2 curso curso    4096 Sep 21 14:06 Public
drwxrwx--x  5 curso curso    4096 Oct  3 20:57 root
drwxr-x--x  2 curso curso    4096 Sep 21 14:06 Templates

curso@ubuntu:~$ ls | grep '^[a-df-z]$_'
picture.PNG
```

Luego como variante se propone una expresión que corresponda con todos los archivos que SI terminen con una letra de la 'a a la d' ó de la 'f a la z'.

```
curso@ubuntu:~$ ls | grep '[a-df-z]$_'
a123z
abz
axi123z
az
a.z
BarrireroExequiel
BarrireroExequiel.tar.gz
Desktop
Documents
Downloads
etc.tar.gz
group
Music
Pictures
Public
root
Templates
Videos
```

4.5. Actividad 5.

4.5.1. Regular expressions - ^, \$, []

Cree una expresión que corresponda con todos los archivos que no comiencen con un 'dígito' y que no terminen con una letra 'minúscula'.

```
curso@ubuntu:~$ ls | grep '^[^0-9].*[^a-z]$_'
picture.PNG
```

Por su parte, se presenta una variante buscando una expresión que corresponda con todos los archivos que no comiencen con un 'dígito' y que SI terminen con una letra 'minúscula'.

```
curso@ubuntu:~$ ls | grep '^[^0-9].*[a-z]$_'
a123z
abz
axi123z
az
a.z
BarrireroExequiel
BarrireroExequiel.tar.gz
Desktop
Documents
Downloads
etc.tar.gz
```

```
group
Music
Pictures
Public
root
Templates
Videos
```

4.6. Actividad 6.

4.6.1. The command-line history

Exportar el contenido del comando history a un fichero de texto plano llamado 'historial.txt'.

The **history command** is a convenient tool that you can use to review previous commands. To look at the most recent commands, simply press the up arrow key. Each time you press it, you will go deeper into the command history. To see all of the command history, type “history” from the command line. You can even export that list to a file to save for future use by entering:

```
history -w history-list.txt
```

MORE ABOUT the HISTORY COMMAND:

If you need to append your current history to the end of a previously created history file, type:

```
history -a history-list.txt
```

You can also append whatever you have in your history file to the end of your current history list (useful if it has been erased). Type:

```
history -r history-list.txt
```

You will notice that each command in the history list is numbered. Each number represents a shortcut that you can use to call any previous command. For example, if “cat /proc/cpuinfo” is number 136 on the list, you can call it by placing a “!” in front of the number. For example:

```
!136
```

Press Enter, and it will automatically execute the associated command.

Finally, if you ever need to empty your history list, for either security or privacy reasons, type:

```
history -C
```

[Hosting News LLC, 2010]

The command-line history

Using the command history

Use the up and down key's to scroll through previously typed commands. Press [Enter] to execute them or use the left and right arrow keys to edit the command first. Also see history (below).

The history command

The history command can be used to list Bash's log of the commands you have typed:

This log is called the "history". To access it type:

```
history n
```

This will only list the last *n* commands. Type "history" (without options) to see the entire history list. You can also type "!"*n* to execute command number *n*. Use "!" to execute the last command you typed. "!"*n* will execute the command *n* times before (in other words "!"*-1* is equivalent to "!"!). "!"*string* will execute the last command starting with that "string" and "!"*?string* will execute the last command containing the word "string". For example:

```
!cd
```

Will re-run the command that you last typed starting with "cd". "command-Name !" will execute the "commandName" with any arguments you used on your last command. This maybe useful if you make a spelling mistake, for example. If you typed:

```
emacs /home/fred/mywork.java /tmp/testme.java
```

In an attempt to execute "emacs" on the above two files this will obviously fail. So what you can do is type:

```
emacs !*
```

This will execute emacs with the arguments that you last typed on the command-line. In other words this is equivalent to typing:

```
emacs /home/fred/mywork.java /tmp/testme.java
```

Searching through the Command History (CTRL-R) Use the CTRL-R key to perform a “reverse-i-search”. For example, if you wanted to use the command you used the last time you used snort, you would type:

CTRL-R then type “snort”.

What you will see in the console window is:

(reverse-i-search) `: After you have typed what you are looking for, use the CTRL-R key combination to scroll backward through the history. Use CTRL-R repeatedly to find every reference to the string you’ve entered. Once you’ve found the command you’re looking for, use [Enter] to execute it. Alternatively, using the right or left arrow keys will place the command on an actual command-line so you can edit it.

[Hogbin E., 2005b]

4.7. Actividad 7.

4.7.1. history, variable HISTSIZE

Seteando la variable de entorno HISTSIZE, en el archivo .bash_profile podrá limitar la cantidad de líneas almacenadas por el history. Investigue al respecto y limite a 10 la cantidad de líneas almacenadas por el history.

```
curso@ubuntu:~$ cat /home/curso/.bashrc
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

curso@ubuntu:~$ echo $HISTSIZE
1000
curso@ubuntu:~$ export HISTSIZE=10
curso@ubuntu:~$ history
450  ls | grep '[^a-df-z]${'
451  ls | grep '[a-df-z]${'
452  ls | grep '^[^0-9].*[a-z]${'
453  ls | grep '[0-9].*[a-z]${'
454  ls | grep '^[^0-9].*[a-z]${'
455  clear
456  man history
457  echo $HISTSIZE
458  export HISTSIZE=10
459  history
```

Disable the usage of history using HISTSIZE

If you want to disable history all together and don’t want bash shell to remember the commands you’ve typed, set the HISTSIZE to 0 as shown below.

```
# export HISTSIZE=0
# history
# [Note that history did not display anything]
```

[Natarajan R., 2015]

4.8. Actividad 8.

4.8.1. HISTCONTROL variable

Seteando la variable de entorno HISTCONTROL = ignoredups borra entradas duplicadas consecutivas. Investigue de que modo debería setear la variable de entorno HISTCONTROL para eliminar entradas duplicadas en todo el archivo.

Ignore Duplicate Commands in History With the below command will help us to ignore duplicate commands entry made by user. Only single entry will be shown in history, if a user execute a same command multiple times in a Bash Prompt.

```
curso@ubuntu:~$ export HISTSIZE=10
curso@ubuntu:~$ export HISTCONTROL=

curso@ubuntu:~$ history
414  history
415  export HISTCONTROL=
416  history
417  history
418  export HISTCONTROL=ignoredups
419  history
420  exit
421  export HISTSIZE=10
422  export HISTCONTROL=
423  history

curso@ubuntu:~$ history
415  export HISTCONTROL=
416  history
417  history
418  export HISTCONTROL=ignoredups
419  history
420  exit
421  export HISTSIZE=10
422  export HISTCONTROL=
423  history
424  history

curso@ubuntu:~$ history
416  history
417  history
418  export HISTCONTROL=ignoredups
419  history
420  exit
421  export HISTSIZE=10
422  export HISTCONTROL=
```

```

423  history
424  history
425  history

curso@ubuntu:~$ export HISTCONTROL=ignoredups

curso@ubuntu:~$ history
418  export HISTCONTROL=ignoredups
419  history
420  exit
421  export HISTSIZE=10
422  export HISTCONTROL=
423  history
424  history
425  history
426  export HISTCONTROL=ignoredups
427  history

curso@ubuntu:~$ history
418  export HISTCONTROL=ignoredups
419  history
420  exit
421  export HISTSIZE=10
422  export HISTCONTROL=
423  history
424  history
425  history
426  export HISTCONTROL=ignoredups
427  history
curso@ubuntu:~$

```

4.9. Actividad 9.

4.9.1. clear history (history -c)

Haga uso del comando history para limpiar el historial de comandos.

if you ever need to empty your history list, for either security or privacy reasons, type:

```
history -C
```


5. Parcial 1 - Parte 2 - Cuestionario Práctico

5.1. Actividad 1.

5.1.1. Regular expressions - ^, \$, ., *

Haciendo uso del comando "grep", contar la cantidad de ocurrencias de palabras: que comiencen con la letra h y terminen con la letra e sin distinguir entre minúsculas y mayúsculas dentro de todos los archivos que se encuentren en la carpeta home y sus subdirectorios.

```
grep -i -r '^h.*e$' /home/
```

5.2. Actividad 2.

5.2.1. find command

Haciendo uso del comando find, encuentre todos los archivos que cumplan TODAS las siguientes condiciones:

- Pertenecen al usuario cuyo uid es igual a 0.
- Que fueron accedidos en los últimos 30 minutos.
- Se encuentren en el directorio /etc.
- Limite el nivel máximo de recursividad (profundidad) a 2.

```
find /etc -uid 0 -maxdepth 2 -amin -30
```

5.3. Actividad 3.

5.3.1. Regular expressions and ls command

Haciendo uso del comando ls, encuentre todos los archivos que cumplan TODAS las siguientes condiciones:

- Comiencen con la letra e y terminen con .log
- Se encuentren en el directorio /var/log.
- Muestre archivos ocultos y links simbólicos.

```
ls -la /var/log | grep '^e.*\.log$'
```

5.4. Actividad 4.

5.4.1. Regular expressions - Tar command

Utilizando el comando tar, realice la siguiente actividad en un solo comando:

- Comprima la carpeta /home/curso utilizando algoritmo bzip2.
- Excluya los archivos que finalizen con la palabra ".txt".
- El archivo debe llamarse backup.tar.bz2
- El archivo debe guardarse en /tmp.
- Sin modo verboso.

```
tar --exclude='.*\.txt$' -cjSf /tmp/backup.tar.bz2 /home/curso
```

6. Ejercicios Tema 7: Shell scripting básico

- Razones para hacer scripting.
- Creación de shell scripts.
- Estándar I/O: STDIN, STDOUT, STDERR
- Redirección de estándares I/O a archivos.
 - Redirección de estándar output a programas: pipe.
 - Combinación de estándar output y error.
 - Redirección del estándar input desde un archivo.
- Expresiones repetitivas: for

6.1. Actividad 1.

6.1.1. STDERR redirection to file

Redireccione el STDERR de la salida que daría el comando `grep da *` a un archivo llamado `errores-de-grep.txt`.

Bash and other modern shell provides I/O redirection facility. There are 3 default standard files (standard streams) open:

0. **stdin** - Use to get input (keyboard) i.e. data going into a program.
1. **stdout** - Use to write information (screen)
2. **stderr** - Use to write error message (screen)

NOTE: Consider the numbers before to also identify the type of input in the console.

Redirecting the standard error stream to a file The following will redirect program error message to a file called `error.log`:

```
$ program-name 2> error.log
$ command1 2> error.log

curso@ubuntu:~$ pwd
/home/curso
curso@ubuntu:~$ mkdir tema7
curso@ubuntu:~$ cd tema7/
curso@ubuntu:~/tema7$ touch errores-de-grep.txt; da * 2> errores-de-grep.txt
curso@ubuntu:~/tema7$ ls
errores-de-grep.txt
curso@ubuntu:~/tema7$ cat errores-de-grep.txt
da: command not found
```

6.2. Actividad 2.

6.2.1. STDOUT & STDERR redirection to file

Hacer que la salida STDOUT del comando grep da * se escriba en el mismo descriptor de fichero que STDERR.

Redirecting the standard error (stderr) and stdout to file Use the following syntax:

```
$ command-name &>file
```

OR

```
$ command > file-name 2>&1
```

Another useful example:

```
# find /usr/home -name .profile 2>&1 | more
```

Redirect stderr to stdout Use the command as follows:

```
$ command-name 2>&1
```

Respuesta:

```
curso@ubuntu:~/tema7$ pwd
/home/curso/tema7
curso@ubuntu:~/tema7$ ls
errores-de-grep.txt
curso@ubuntu:~/tema7$ da * 1>&2
da: command not found
curso@ubuntu:~/tema7$
```

6.3. Actividad 3.

6.3.1. Head - Tail - Pipe

Obtener la 5 linea del archivo .Ejemplos de scripts" de la sección Presentación del Tema 7, para ello podemos combinar 2 comandos:

- Head, el cual retorna las primeras lineas de un archivo.
- Tail el cual retorna las últimas lineas.
- Ayuda: Recuerde que debe hacer uso de pipe para combinar dos o mas comandos.

Recordando el archivo **ejemplos_de_script.txt**

```

curso@ubuntu:~/tema7$ cat ejemplos_de_script.txt
RECORDAR

* Estos scripts tienen que tener permiso de ejecución
* Para ejecutar debe hacer referencia a el mediante path relativo o absoluto

----- MODO TEXTO -----

(para que puedan copiarlo y probarlo)

Script1:
ermirizio@atlas:~> cat script1.sh
#!/bin/bash

#Todo comentario empieza con un # por delante
#Esto es un comentario
echo "Hola mundo" #Esto tambien es un comentario

Script2:
ermirizio@atlas:~> cat script2.sh
#!/bin/bash

#Trabajando con variables
VARIABLE="Hola mundo"
echo $VARIABLE

Script 3:
ermirizio@atlas:~> cat calendario
#!/bin/bash

#Este script muestra en el STDOUT el calendario anual del año actual y
ademas guarda en un archivo de texto, llamado calendario.log, el resultado y la
fecha y hora en que se lo ejecuto
#Para practicar, ud podria agregar los comentarios entre cada linea de ejecucion explicando
que es lo que hace

echo "Fecha de ejecucion" > calendario.log
date >> calendario.log
echo "Calendario del año actual" >> calendario.log
ANIOACTUAL=`date | cut -d' ' -f7`
cal $ANIOACTUAL | tee -a calendario.log

```

Luego se presentan una serie de outputs relacionados a la solución de esta actividad

```

curso@ubuntu:~/tema7$ ls
ejemplos_de_script.txt  errores-de-grep.txt
curso@ubuntu:~/tema7$ head -n 5 ejemplos_de_script.txt | tail -n 1

curso@ubuntu:~/tema7$ head -n 5 ejemplos_de_script.txt
RECORDAR

* Estos scripts tienen que tener permiso de ejecución
* Para ejecutar debe hacer referencia a el mediante path relativo o absoluto

```

Notar que la 5ta línea no posee ningún caracter de allí la salida en pantalla presenta una línea vacía. Luego si mediante el comando **head** obtenemos las 1eras

11 líneas y filtramos esta última, es decir la nº 11, el output sera el que se presenta debajo:

```
curso@ubuntu:~/tema7$ head -n 11 ejemplos_de_script.txt | tail -n 1
ermirizio@atlas:~> cat script1.sh
```

6.4. Actividad 4.

6.4.1. Cat - Grep - Wc

Consultar en el archivo actividad4.txt el número de veces que una IP ha sido registrada, para ello podemos combinar los comandos:

- cat para obtener los contenidos del log.
- grep para filtrar las lineas que contengan la cadena "192.168.0.1".
- wc para contar los resultados obtenidos.

```
WC(1) User Commands

NAME
    wc - print newline, word, and byte counts for each file

SYNOPSIS
    wc [OPTION]... [FILE]...
    wc [OPTION]... --files0-from=F

DESCRIPTION
    Print newline, word, and byte counts for each FILE, and a total line if
    more than one FILE is specified.  With
    no FILE, or when FILE is -, read standard input.  A word is a
    non-zero-length sequence of characters delimited
    by white space.  The options below may be used to select which counts
    are printed, always in the following
    order: newline, word, character, byte, maximum line length.

    -c, --bytes
        print the byte counts

    -m, --chars
        print the character counts

    -l, --lines
        print the newline counts

    --files0-from=F
        read input from the files specified by NUL-terminated names in file
        F; If F is - then read names from
        standard input

    -L, --max-line-length
        print the length of the longest line

    -w, --words
        print the word counts
```

Respuesta:

```
curso@ubuntu:~/tema7$ cat actividad4.txt
Acceso 192.168.1.5
Acceso 192.168.1.5
Acceso 192.168.1.5
Acceso 192.168.1.1
Acceso 192.168.1.3
Acceso 192.168.1.2
Acceso 192.168.1.1
Acceso 192.168.1.1
Acceso 192.168.1.1
Acceso 192.168.1.3
Acceso 192.168.1.4
Acceso 192.168.1.5
Acceso 192.168.1.5
Acceso 192.168.1.7
Acceso 192.168.1.1
Acceso 192.168.1.2
Acceso 192.168.1.5
Acceso 192.168.1.3
Acceso 192.168.1.1
Acceso 192.168.1.5
Acceso 192.168.1.5
Acceso 192.168.1.5
Acceso 192.168.1.1
Acceso 192.168.1.5

curso@ubuntu:~/tema7$ cat actividad4.txt | grep "192.168.1.5" | wc -l
10
```

6.5. Actividad 5.

6.5.1. Find & Sort script

Cree un script llamado script1_T7.sh que sea interpretado por bash. Al ejecutarse debe buscar en el directorio HOME/CURSO todos los archivos y ordenelos por nombres alfabéticamente. Redireccione el contenido en un archivo llamado salida_script1_T7.txt. Ayuda: Puede emplear la opción -R del comando ls para listar subdirectorios, y la opción -u del comando sort.

We use `sort -u` to sort and remove duplicates from the text file we are sorting. Check if the duplicate has been removed or not.

Script:

```
#!/bin/bash

ls -R /home/curso | sort -u > salida_script1_T7.txt
```

Output:

```
curso@ubuntu:~/tema7$ ls
actividad4.txt  ejemplos_de_script.txt  errores-de-grep.txt  script1_T7.sh
curso@ubuntu:~/tema7$ chmod +x script1_T7.sh
```

```

curso@ubuntu:~/tema7$ ./script1_T7.sh
curso@ubuntu:~/tema7$ cat salida_script1_T7.txt
.:
128x128
16x16
256x256
32x32
48x48
a123z
aaz
abe
abx
abz
ActionScript.sublime-package
Actividad10.png
Actividad11.png
Actividad12.png
Actividad13.png
Actividad14.png
Actividad15.txt
Actividad15.txt~
Actividad2.png
Actividad3.txt
Actividad3.txt~
actividad4.txt
Actividad4.txt
Actividad4.txt~
...
User.sublime-package
var
Videos
./Videos:
Vintage.sublime-package
XML.sublime-package
YAML.sublime-package
curso@ubuntu:~$

```

6.6. Actividad 6.

6.6.1. “ - in between Grave accent character

¿Qué hace el comando echo ‘ls’ ? (note que se usan apóstrofes invertidos).

Ejecuta el comando ls y el resultado del mismo como cadena lo presenta con el comando echo. Si no tiene las comillas el comando echo mostraría ”ls” Un comando encerrado entre apóstrofes invertidos (Ej.: ‘ls’) o entre las cadenas “\$(” y “)”, será expandido al resultado que tal comando envíe a salida estándar cuando es ejecutado.

También considerar los **TIPS PARA CREAR SCRIPTS:**

—CREACION—

1. Todo script debe comenzar con #!/bin/bash

2. Luego de creado el script, hay que hacerlo ejecutable con `chmod +x nombre_del_script`
3. Para ejecutarlo, si estoy parado en el mismo lugar donde esta el script, debo utilizar `./nombre_del_script`

—DENTRO DEL SCRIPT—

1. Crear Variables

-Asignar una cadena o mas conocida como string

`Nombre_Variable=Valor`

-Asignar una cadena o mas conocida como string

`Nombre_Variable=Valor`

2. Imprimir valor de variable

`$Nombre_Variable`

3. Igualar una nueva variable al valor de otra

`Nueva_Variable=$Nombre_Variable`

4. Impresion de cosas literales

`echo "nombre_variable" ->nombre_variable`

5. Impresión de valores de una variable

`echo "$nombre_variable" ->imprimo el valor que almacena la variable`

6. Impresión de salidas de comandos

`echo "date" ->la palabra date`

`echo "$(date)" ->la fecha de hoy`

`echo `date` ->la fecha de hoy`

`echo `cat /etc/passwd | grep 'root'``

7. Igualar la salida de un comando a una variable

`Nombre_variable=`cat /etc/passwd | grep 'root'``

`ej: variable=`cat /etc/passwd | grep root``

`Para Imprimir ->echo $variable`

[Mirizio E., 2015b]

6.7. Actividad 7.

6.7.1. FOR - loops and iteration

Haciendo uso de la sentencia repetitiva FOR , realice un script que genere un fichero llamado lista.txt que contenga el nombre de los archivos que haya en el directorio /etc. RECUERDE: Debe hacer uso de la expresión repetitiva for.

Script:

```
#!/bin/bash

#Haciendo uso de la sentencia repetitiva FOR , realice un script que genere
#un fichero llamado lista.txt que contenga el nombre de los archivos que haya
#en el directorio /etc. RECUERDE: Debe hacer uso de la expresión repetitiva
#for.

for F in /etc/*
do
echo $F >> lista.txt
done
```

Output:

```
curso@ubuntu:~/tema7$ ./script2_T7.sh

curso@ubuntu:~/tema7$ cat lista.txt
/etc/acpi
/etc/adduser.conf
/etc/alternatives
/etc/anacrontab
/etc/apg.conf
/etc/apm
/etc/apparmor
/etc/apparmor.d
/etc/appport
/etc/apt
/etc/at.deny
/etc/at-spi2
/etc/avahi
...
/etc/vim
/etc/vtrgb
/etc/w3m
/etc/wgetrc
/etc/wodim.conf
/etc/wpa_supplicant
/etc/X11
/etc/xdg
/etc/xml
/etc/zsh_command_not_found
curso@ubuntu:~/tema7$
```

7. Ejercicios Tema 8: Herramientas para procesar texto

- Herramientas para extraer texto.
 - Ver contenido de archivos: less, cat.
 - Ver extractos de archivos: head, tail.
 - Extraer texto en base a palabras claves: grep.
- Extraer texto por columna: cut.
 - Ver contenido de archivos: less, cat
- Herramientas para analizar texto.
 - Estadísticas: wc.
 - Ordenamiento de texto: sort, uniq.
- Herramientas para manipular texto:
 - Alterar caracteres: tr
 - Alterar strings: sed
- Caracteres especiales para búsquedas complejas.

7.1. Actividad 1.

7.1.1. cat command

Cree dos archivos con los siguientes nombres:

- ArchivoA.txt con el contenido “Esta es la primer línea del Archivo A”
- ArchivoB.txt con el contenido “Esta es la primer línea del Archivo B”

Dado que el comando cat admite como argumentos una lista de archivos que se pueden enlazar observe que sucede al ejecutar la siguiente línea:

```
# cat ArchivoA.txt ArchivoB.txt

curso@ubuntu:~$ touch ArchivoA.txt ArchivoB.txt

curso@ubuntu:~$ echo "Esta es la primera línea del Archivo A" > ArchivoA.txt
curso@ubuntu:~$ echo "Esta es la primera línea del Archivo B" > ArchivoB.txt

curso@ubuntu:~$ cat ArchivoA.txt
Esta es la primera línea del Archivo A

curso@ubuntu:~$ cat ArchivoB.txt
Esta es la primera línea del Archivo B
curso@ubuntu:~$ cat ArchivoA.txt ArchivoB.txt
Esta es la primera línea del Archivo A
Esta es la primera línea del Archivo B
```

MAN cat

CAT(1)

User Commands

NAME

cat - concatenate files and print on the standard output

SYNOPSIS

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

DESCRIPTION

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

-A, --show-all

equivalent to -vET

-b, --number-nonblank

number nonempty output lines, overrides -n

-e

equivalent to -vE

-E, --show-ends

display \$ at end of each line

-n, --number

number all output lines

-s, --squeeze-blank

suppress repeated empty output lines

-t

equivalent to -vT

-T, --show-tabs

display TAB characters as ^I

7.2. Actividad 2.

7.2.1. less command

Haciendo uso del comando less muestre el contenido del archivo /etc/hosts empezando por la tercera línea.

+ If a command line option begins with +, the remainder of that option is taken to be an initial command to less. For example, +G tells less to start at the end of the file rather than the beginning, and +/xyz tells it to start at the first occurrence of "xyz" in the file. As a special case, +<number> acts like +<number>g; that is, it starts the display at the specified line number (however, see the caveat under the "g" command above). If the option starts with ++, the initial command applies to every file being viewed, not just the first one. The + command described previously may also be used to set (or change) an initial command for every file.

The command to solve the exercise.

```
curso@ubuntu:~/tema8$ less +3 /etc/hosts
```

Then the output starts at the 3rd line:

```
# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
~
~
```

7.3. Actividad 3.

7.3.1. grep & regular expressions

Haciendo uso del comando grep y expresiones regulares muestre las líneas que comiencen por la letra 'd' en el archivo /etc/passwd

Se presentan debajo 2 formas de resolver el enunciado anterior:

```
curso@ubuntu:~/tema8$ grep '^d' /etc/passwd
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
delivery:x:1000:1000:delivery,,,:/home/delivery:/bin/bash

curso@ubuntu:~/tema8$ grep '^[d]' /etc/passwd
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
delivery:x:1000:1000:delivery,,,:/home/delivery:/bin/bash
```

7.4. Actividad 4.

7.4.1. head command

Haciendo uso del comando head obtener las 5 primeras líneas del archivo /etc/group

EAD (1)	User Commands
NAME	head - output the first part of files
SYNOPSIS	head [OPTION]... [FILE]...
DESCRIPTION	Print the first 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. With no FILE, or when FILE is -, read standard input. Mandatory arguments to long options are mandatory for short options too. -c, --bytes=[-]K print the first K bytes of each file; with the leading '-', print

```

        all but the last K bytes of each file

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

-q, --quiet, --silent
    never print headers giving file names

-v, --verbose
    always print headers giving file names

curso@ubuntu:~/tema8$ head -n 5 /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,delivery
curso@ubuntu:~/tema8$

```

7.5. Actividad 5.

7.5.1. sort command

Haciendo uso del comando sort ordenar por la columna de permisos la salida del comando ls -l. Ayuda: Recuerde que debe hacer uso de pipe para combinar dos o mas comandos.

MAN sort

```

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.

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

    -b, --ignore-leading-blanks
        ignore leading blanks

    -d, --dictionary-order
        consider only blanks and alphanumeric characters

    -f, --ignore-case
        fold lower case to upper case characters

    -g, --general-numeric-sort
        compare according to general numerical value

```

```

-i, --ignore-nonprinting
    consider only printable characters

-M, --month-sort
    compare (unknown) < 'JAN' < ... < 'DEC'

-h, --human-numeric-sort
    compare human readable numbers (e.g., 2K 1G)

-n, --numeric-sort
    compare according to string numerical value

-R, --random-sort
    sort by random hash of keys

--random-source=FILE
    get random bytes from FILE

-r, --reverse
    reverse the result of comparisons

```

Respuesta:

```

curso@ubuntu:~$ ls
ArchivoA.txt  BarrireroExequiel  Desktop  Downloads  Music  Public
Software  tema8  Videos
ArchivoB.txt  BarrireroExequiel.tar.gz  Documents  group  Pictures  root
tema7  Templates
curso@ubuntu:~$ ls -l | sort
drwxrwxr-x 2 curso curso 4096 Dec 6 18:02 tema7
drwxrwxr-x 2 curso curso 4096 Dec 7 09:12 tema8
drwxrwxr-x 3 curso curso 4096 Dec 5 16:50 Software
drwxrwx--x 3 curso curso 4096 Dec 6 09:23 BarrireroExequiel
drwxrwx--x 5 curso curso 4096 Oct 3 20:57 root
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Documents
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Music
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Public
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Templates
drwxr-x--x 2 curso curso 4096 Sep 21 14:06 Videos
drwxr-x--x 2 curso curso 4096 Sep 21 19:30 Downloads
drwxr-x--x 2 curso curso 4096 Sep 21 20:53 Pictures
drwxr-x--x 3 curso curso 4096 Dec 5 16:52 Desktop
lrwxrwxrwx 1 curso curso 10 Sep 29 19:55 group -> /etc/group
-rw-rw-r-- 1 curso curso 40 Dec 7 09:12 ArchivoA.txt
-rw-rw-r-- 1 curso curso 40 Dec 7 09:12 ArchivoB.txt
-rwxrw---- 1 curso curso 2013003 Sep 21 21:13 BarrireroExequiel.tar.gz
total 2028

```

SORT command:

Sort is a Linux program used for printing lines of input text files and concatenation of all files in sorted order. Sort command takes blank space as field separator and entire Input file as sort key. It is important to notice that sort command don't actually sort the files but only print the sorted output, until your redirect the output.

1 - Sort command without any options

Sorts lines in test file and displays sorted output.

```
$ sort test
aaa
AAA
BBB
ddd
qqq
sss
zzz
```

2 - Perform Numeric Sort using -n option

If we want to sort on numeric value, then we can use -n or --numeric-sort option.

```
$ sort -n test
11 qqq
22 zzz
33 sss
55 BBB
77 aaa
```

3 - Sort Human Readable Numbers using -h option

If we want to sort on human readable numbers (e.g., 2K 1M 1G), then we can use -h or --human-numeric-sort option.

```
$ sort -h test
1K
2K
2M
1G
2G
1T
6T
```

4 - Sort Months of an Year using -M option

If we want to sort in the order of months of year, then we can use -M or --month-sort option. The following sort command sorts lines in test file as per month order. Note, lines in file should contain at least 3 character name of month name at start of line (e.g. jan, feb, mar). If we will give, ja for January or au for August, then sort command would not consider it as month name.

```
$ sort -M test
jan
feb
mar11
apr
aug
sept
oct
```

5 - Check if Content is Already Sorted using -c option

If we want to check data in text file is sorted or not, then we can use -c or --check, --check=diagnose-first option. The following sort command checks whether text file data is sorted or not. If it is not, then it shows first occurrence with line number and disordered value.


```
$ sort -c test
sort: test:3: disorder: 1
```

6 - Reverse the Output and Check for Uniqueness using -r and -u options

If we want to get sorted output in reverse order, then we can use -r or --reverse option. If file contains duplicate lines, then to get unique lines in sorted output, “-u” option can be used.

The following sort command sorts lines in test file in reverse order and removes duplicate lines from sorted output.

```
$ sort -r test
5
4
4
2
2
1

$ sort -r -u test
5
4
2
1
```

7 - Selectively Sort the Content, Customize delimiter, Write output to a file using -k, -t, -o options

If we want to sort on the column or word position in lines of text file, then “-k” option can be used. If we each word in each line of file is separated by delimiter except ‘space’, then we can specify delimiter using “-t” option. We can get sorted output in any specified output file (using “-o” option) instead of displaying output on standard output.

The following sort command sorts lines in test file on the 3rd word of each line and displays sorted output.

```
$ cat test
aa aa zz
aa aa ff
aa aa tt
aa aa kk

$ sort -k3 test
aa aa ff
aa aa kk
aa aa tt
aa aa zz
```

Here, several options are used altogether. In test file, words in each line are separated by delimiter ‘|’. It sorts lines in test file on the 2nd word of each line on the basis of numeric value and stores sorted output into specified output file.

```
$ cat test
aa|5a|zz
aa|2a|ff
aa|1a|tt
aa|3a|kk

$ sort -n -t'|' -k2 test -o outfile

$ cat outfile
aa|1a|tt
aa|2a|ff
aa|3a|kk
aa|5a|zz
```

[Shrestha N., 2013b]

7.6. Actividad 6.

7.6.1. cut command

Haciendo uso del comando cut obtener el 1º, el 5º y todos los caracteres comprendidos entre el 10º y el 20º del archivo /etc/passwd.

MAN cut

```
CUT(1) User Commands

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.

    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

    -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, unless the -s
        option is specified

    -n
        (ignored)

    --complement
        complement the set of selected bytes, characters or fields

    -s, --only-delimited
```

```
do not print lines not containing delimiters

--output-delimiter=STRING
    use STRING as the output delimiter the default is to use the input
    delimiter
```

Respuesta:

```
curso@ubuntu:~$ cut -c1,5,10-20 /etc/passwd
r:0:root:/roo
do1:1:daemon:
bx:bin:/bin:/
sx:sys:/dev:/
s:65534:sync:
gs:60:games:/
mx2:man:/var/
l:lp:/var/spo
m:8:mail:/var
n:9:news:/var
u::10:uucp:/v
py3:13:proxy:
wdx:33:33:www
bu34:34:backu
l::38:Mailing
ix39:ircd:/va
gs1:41:Gnats
nd65534:65534
lu:100:101:/
so101:104:/h
mas:x:102:106
ls:x:103:109:
s:4:65534:/v
dvx:1000:1000
va:999:1:/va
uu105:46:usbm
ai06:113:Avah
cr107:116:col
pe08:117:Puls
rt09:119:Real
lt:110:120:Li
co001:1001:/
curso@ubuntu:~$
```

SORT command:

Linux command cut is used for text processing. You can use this command to extract portion of text from a file by selecting columns.

1 - Select Column of Characters

To extract only a desired column from a file use -c option. The following example displays 2nd character from each line of a file test.txt

```
$ cat test.txt
cat command for file oriented operations.
cp command for copy files or directories.
ls command to list out files and directories with its attributes.
```

```
$ cut -c2 test.txt
a
p
s
```

As seen above, the characters a, p, s are the second character from each line of the test.txt file.

2 - Select Column of Characters using Range

Range of characters can also be extracted from a file by specifying start and end position delimited with -. The following example extracts first 3 characters of each line from a file called test.txt

```
$ cat test.txt
cat command for file oriented operations.
cp command for copy files or directories.
ls command to list out files and directories with its attributes.

$ cut -c1-3 test.txt
cat
cp
ls
```

3 - Select Column of Characters using either Start or End Position

Either start position or end position can be passed to cut command with -c option.

The following specifies only the start position before the '-'. This example extracts from 3rd character to end of each line from test.txt file.

```
$ cut -c3- test.txt
t command for file oriented operations.
command for copy files or directories.
command to list out files and directories with its attributes.
```

The following specifies only the end position after the '-'. This example extracts 8 characters from the beginning of each line from test.txt file.

```
$ cut -c-8 test.txt
cat comm
cp comma
ls comma
```

The entire line would get printed when you don't specify a number before or after the '-' as shown below.

```
$ cut -c- test.txt
cat command for file oriented operations.
cp command for copy files or directories.
ls command to list out files and directories with its attributes.
```

4 - Select a Specific Field from a File

Instead of selecting x number of characters, if you like to extract a whole field, you can combine option -f and -d. The option -f specifies which field you want to extract, and the option -d specifies what is the field delimiter that is used in the input file.

The following example displays only first field of each lines from /etc/passwd file using the field delimiter : (colon). In this case, the 1st field is the username. The file

```
$ cut -d':' -f1 /etc/passwd
root
daemon
bin
sys
sync
games
bala
```

5 - Select Multiple Fields from a File

You can also extract more than one fields from a file or stdout. Below example displays username and home directory of users who has the login shell as “/bin/bash”.

```
$ grep "/bin/bash" /etc/passwd | cut -d':' -f1,6
root:/root
bala:/home/bala
```

To display the range of fields specify start field and end field as shown below. In this example, we are selecting field 1 through 4, 6 and 7

```
$ grep "/bin/bash" /etc/passwd | cut -d':' -f1-4,6,7
root:x:0:0:/root:/bin/bash
bala:x:1000:1000:/home/bala:/bin/bash
```

6 - Select Fields Only When a Line Contains the Delimiter

In our /etc/passwd example, if you pass a different delimiter other than : (colon), cut will just display the whole line.

In the following example, we've specified the delimiter as | (pipe), and cut command simply displays the whole line, even when it doesn't find any line that has | (pipe) as delimiter.

```
$ grep "/bin/bash" /etc/passwd | cut -d'|' -f1
root:x:0:0:root:/root:/bin/bash
bala:x:1000:1000:bala,,,:/home/bala:/bin/bash
```

But, it is possible to filter and display only the lines that contains the specified delimiter using -s option.

The following example doesn't display any output, as the cut command didn't find any lines that has | (pipe) as delimiter in the /etc/passwd file.

```
$ grep "/bin/bash" /etc/passwd | cut -d'|' -s -f1
```

7 - Select All Fields Except the Specified Fields

In order to complement the selection field list use option `–complement`.

The following example displays all the fields from `/etc/passwd` file except field

7

```
$ grep "/bin/bash" /etc/passwd | cut -d':' --complement -s -f7
root:x:0:0:root:/root
bala:x:1000:1000:bala,,,:/home/bala
```

8 - Change Output Delimiter for Display

By default the output delimiter is same as input delimiter that we specify in the `cut -d` option.

To change the output delimiter use the option `–output-delimiter` as shown below. In this example, the input delimiter is `:` (colon), but the output delimiter is `#` (hash).

```
$ grep "/bin/bash" /etc/passwd | cut -d':' -s -f1,6,7 --output-delimiter='#'
root#/root#/bin/bash
bala#/home/bala#/bin/bash
```

9 - Change Output Delimiter to Newline

In this example, each and every field of the `cut` command output is displayed in a separate line. We still used `–output-delimiter`, but the value is `$'\n'` which indicates that we should add a newline as the output delimiter.

```
$ grep bala /etc/passwd | cut -d':' -f1,6,7 --output-delimiter=$'\n'
bala
/home/bala
/bin/bash
```

[Shrestha N., 2013a]

7.7. Actividad 7.

7.7.1. sed command

Haciendo uso del comando `sed` elimine todas las lineas del archivo que comiencen con `#` (comentarios).

Script.txt

```
#!/bin/bash
```

```
#Mi primer programa
```

```
echo "Hola Mundo"
```

MAN sed

```
SED(1) User Commands

NAME
    sed - stream editor for filtering and transforming text

SYNOPSIS
    sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION
    Sed is a stream editor. A stream editor is used to perform basic text
    transformations on an input stream (a file or input from a pipeline). While in some ways similar to an editor
    which permits scripted edits (such as ed), sed works by making only one pass over the input(s), and is
    consequently more efficient. But it is sed's ability to filter text in a pipeline which particularly distinguishes it
    from other types of editors.
```

Respuesta:

```
curso@ubuntu:~/tema8$ pwd
/home/curso/tema8
curso@ubuntu:~/tema8$ ls
actividad7_T8.sh
curso@ubuntu:~/tema8$ cat actividad7_T8.sh
#!/bin/bash

#Haciendo uso del comando sed elimine todas las lineas del archivo que
#comiencen con \# (comentarios).

#Mi primer programa

echo "Hola Mundo"curso@ubuntu:~/tema8$ sed 's/^#.*//g' actividad7_T8.sh

echo "Hola Mundo"curso@ubuntu:~/tema8$
```

SED command:

The 's' (or sed) command attempts to match the pattern space against the supplied REGEXP; if the match is successful, then that portion of the pattern space which was matched is replaced with REPLACEMENT.

Syntax:

```
#sed 'ADDRESSs/REGEXP/REPLACEMENT/FLAGS' filename
#sed 'PATTERNs/REGEXP/REPLACEMENT/FLAGS' filename
```

- s is substitute command
- / is a delimiter

- **REGEXP** is regular expression to match
- **REPLACEMENT** is a value to replace

FLAGS can be any of the following

- **g** Replace all the instance of REGEXP with REPLACEMENT
- **n** Could be any number, replace nth instance of the REGEXP with REPLACEMENT.
- **p** If substitution was made, then prints the new pattern space.
- **i** match REGEXP in a case-insensitive manner.
- **w** file If substitution was made, write out the result to the given file.
- We can use different delimiters (one of @ % ; :) instead of /

1 - Substitute Word “Linux” to “Linux-Unix” Using sed s//

In the example below, in the output line “1. Linux-Unix Sysadmin, Linux Scripting etc” only first Linux is replaced by Linux-Unix. If no flags are specified the first match of line is replaced.

```
$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed 's/Linux/Linux-Unix/' example.txt
# Instruction Guides
1. Linux-Unix Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux-Unix
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.
```

2 - Substitute all Appearances of a Word Using sed s//g

The below sed command replaces all occurrences of Linux to Linux-Unix using global substitution flag “g”.

```
$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
```



```
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed 's/Linux/Linux-Unix/g' example.txt
# Instruction Guides
1. Linux-Unix Sysadmin, Linux-Unix Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux-Unix
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.
```

3 - Substitute Only 2nd Occurrence of a Word Using sed s//2

In the example below, in the output line “1. Linux Sysadmin, Linux-Unix Scripting etc.” only 2nd occurrence of Linux is replaced by Linux-Unix.

```
$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed 's/Linux/Linux-Unix/2' example.txt
# Instruction Guides
1. Linux Sysadmin, Linux-Unix Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.
```

4 - Write Changes to a File and Print the Changes Using sed s//gpw

The example below has substitution with three flags. It substitutes all the occurrence of Linux to Linux-Unix and prints the substituted output as well as written the same to the given file.

```
$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed -n 's/Linux/Linux-Unix/gpw output' example.txt
1. Linux-Unix Sysadmin, Linux-Unix Scripting etc.
4. Storage in Linux-Unix
$ cat output
1. Linux-Unix Sysadmin, Linux-Unix Scripting etc.
```

4. Storage in Linux-Unix

5 - Substitute Only When the Line Matches with the Pattern Using sed

In this example, if the line matches with the pattern “-“, then it replaces all the characters from “-” with the empty.

```
$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, mySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed 's/-.*//g' example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows
```

6 - Delete Last X Number of Characters From Each Line Using sed

This sed example deletes last 3 characters from each line.

```
$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, mySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed 's/...$//' example.txt
# Instruction Gui
1. Linux Sysadmin, Linux Scripting e
2. Databases - Oracle, mySQL e
3. Security (Firewall, Network, Online Security e
4. Storage in Li
5. Productivity (Too many technologies to explore, not much time availab
# Additional F
6. Windows- Sysadmin, reboot e
```

7 - Eliminate Comments Using sed

Delete all the comment lines from a file as shown below using sed command.

```
$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
```

```

2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed -e 's/#.*//' example.txt

1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)

6. Windows- Sysadmin, reboot etc.

```

8 - liminate Comments and Empty Lines Using sed

In this example, there are two commands seperated by ';'.

- First command replaces the lines starting with the # to the blank lines
- Second command deletes the empty lines.

```

$ cat example.txt
# Instruction Guides
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
# Additional FAQs
6. Windows- Sysadmin, reboot etc.

$ sed -e 's/#.*//; /^$/d' example.txt
1. Linux Sysadmin, Linux Scripting etc.
2. Databases - Oracle, MySQL etc.
3. Security (Firewall, Network, Online Security etc)
4. Storage in Linux
5. Productivity (Too many technologies to explore, not much time available)
6. Windows- Sysadmin, reboot etc.

```

9 - Convert DOS newlines (CR/LF) to Unix format Using sed

Copy the DOS file to Unix, you could find `\r\n` in the end of each line.

This example converts the DOS file format to Unix file format using sed command.

```
$sed 's/.$//' filename
```

10 - Eliminate HTML Tags from file Using sed In this example, the regular expression given in the sed command matches the html tags and replaces with the empty.

```
$ sed -e 's/<[^>]*>//g'  
This <b> is </b> an <i>example</i>.  
This is an example.
```

[Shrestha N., 2009]

8. Ejercicios Tema 10: Shell scripting intermedio

- Agrupación de comandos.
- Interpretación del estado de salida de un proceso.
- Operadores de ejecución condicional: ||, &&.
- Evaluación booleana de sentencias: test.
- Expresiones condicionales: if.
- Interpretación de parámetros de un comando.
- Lectura de entrada como argumento de un parámetro: read.

8.1. Actividad 1.

8.1.1. tar - gzip - date format output

Realizar un script que dado un directorio, cree un archivo tar comprimido con gzip y con nombre igual a la fecha en formato yyyy-mm-dd seguido del nombre del directorio.tar.gz Por ejemplo: aplicado al directorio tmp obtendríamos 2012-07-11tmp.tar.gz.

```
#!/bin/bash

#Realizar un script que dado un directorio, cree un archivo tar comprimido con
#gzip y con nombre igual a la fecha en formato #yyyy-mm-dd seguido del nombre del directorio.tar.gz
#Por ejemplo: aplicado al directorio tmp obtendríamos 2012-07-11tmp.tar.gz.

#tar -zcvf archive-name.tar.gz directory-name
#Where,
#-z : Compress archive using gzip program
#-c : Create archive
#-v : Verbose i.e display progress while creating archive
#-f : Archive File name

Date=$(date +"%Y-%m-%d")
echo "#####"
echo "##### Date: $Date #####"
echo "#####"

read -p "Input absolute path DIR to Compersss: " Dir
echo "#####"
echo "# Directory to Compress: $Dir      #"
echo "#####"

Dir_name=$(echo "$Dir" | tr '/' '-')
echo "#####"
echo "## The dir name is: $Dir_name      "
echo "#####"

tar -zcvf $Date$Dir_name.tar.gz "$Dir"
compress_f=$(ls -la | grep tar.gz | cut -d' ' -f8-12)
```

```
echo "#####"
echo "##### Your backups: #####"
echo "$compress_f"
echo "#####"
```

Then the result:

```
curso@ubuntu:~/tema10$ ./actividad1.sh
#####
##### Date: 2015-12-08 #####
#####
Input absolute path DIR to Comperss: /tmp
#####
# Directory to Compress: /tmp      #
#####
#####
## The dir name is: -tmp
#####
tar: Removing leading '/' from member names
/tmp/
/tmp/config-err-epPb49
/tmp/.org.chromium.Chromium.JkwRvY/
tar: /tmp/.org.chromium.Chromium.JkwRvY/SingletonSocket: socket ignored
/tmp/.org.chromium.Chromium.JkwRvY/SingletonCookie
/tmp/.ICE-unix/
tar: /tmp/.ICE-unix/1633: socket ignored
/tmp/.X0-lock
/tmp/orbit-curso/
/tmp/.X11-unix/
tar: /tmp/.X11-unix/X0: socket ignored
#####
##### Your backups: #####
Dec  8 11:31 2015-12-08-tmp.tar.gz
#####
curso@ubuntu:~/tema10$
```

8.2. Actividad 2.

8.2.1. script : "divisible por 2"

**Realizar un script que, dado un número, indique si es o no divisible por 2.
Si no se ingresa un número debe decir como usar el programa.**

```
#!/bin/bash

#Realizar un script que, dado un número, indique si es o no divisible por 2.
#Si no se ingresa un número debe decir como usar el programa.

divisor=2

read -t 3 -p "Input a Number to check if it's divisible by 2 : " Number || echo -e "\nPlase enter"
#echo $(( $Number / $divisor ))
#expr $Number / $divisor
#if [ $(( $number % 5 )) -eq 0 ]

# Use the && (and) and || (or) operators:
```

```

if [[ expression ]] && [[ expression ]] || [[ expression ]] ; then

#They can also be used within a single [[ ]]:
#if [[ expression && expression || expression ]] ; then

#And, finally, you can group them to ensure order of evaluation:
#if [[ expression && ( expression || expression ) ]] ; then

if [[ "$Number" -ne 0 ]] && [[ -n $Number ]]
then
if (($Number % $divisor == 0))
then
    echo "Your number is divisible by 2"
else
    echo "Your number is not divisible by 2"
fi
else
echo "You must enter a number dif from 0 (cero) to check if it's divisible by 2"
fi

```

Then the OUTPUT:

```

curso@ubuntu:~/tema10$ ./actividad2.sh
Input a Number to check if it's divisible by 2 :
Plase enter a number during the 1st 3 secs
You must enter a number dif from 0 (cero) to check if it's divisible by 2
curso@ubuntu:~/tema10$ ./actividad2.sh
Input a Number to check if it's divisible by 2 : 0
You must enter a number dif from 0 (cero) to check if it's divisible by 2
curso@ubuntu:~/tema10$ ./actividad2.sh
Input a Number to check if it's divisible by 2 : 1
Your number is not divisible by 2
curso@ubuntu:~/tema10$ ./actividad2.sh
Input a Number to check if it's divisible by 2 : 2
Your number is divisible by 2
curso@ubuntu:~/tema10$ ./actividad2.sh
Input a Number to check if it's divisible by 2 : 3
Your number is not divisible by 2
curso@ubuntu:~/tema10$ ./actividad2.sh
Input a Number to check if it's divisible by 2 : 4
Your number is divisible by 2
curso@ubuntu:~/tema10$

```

8.3. Actividad 3.

8.3.1. script: tar - gzip - date format output

Realizar un script que dado una lista de directorios, cree un archivo tar comprimido con gzip con nombre igual a la fecha en formato yyyy-mm-dd.tar.gz. Además se debe generar un fichero yyyy-mm-dd.lst con los nombres de los directorios contenidos en el archivo tar, uno por linea usando un bucle. Si el fichero lst existe, mostrar un error y terminar el programa. Si alguno de los elementos no es un directorio, mostrar un error y finalizar el programa.

```
#!/bin/bash
```

```

#Realizar un script que dado una lista de directorios, cree un archivo tar
#comprimido con gzip con nombre igual a la fecha en formato yyyy-mm-dd.tar.gz.
#Además se debe generar un fichero yyyy-mm-dd.lst con los nombres de los di-
#rectorios contenidos en el archivo tar, uno por linea usando un bucle. Si el
#fichero lst existe, mostrar un error y terminar el programa. Si alguno de los
#elementos no es un directorio, mostrar un error y finalizar el programa.

# 1386 ls -d /home/curso/*
# 1387 ls -x /home/curso/* | grep '^d'
# 1388 ls -x -R /home/curso/
read -p "Input absolute path DIR to get a List of
subdirs in a .txt and compress this file list: " Dir

echo "#####"
echo "# List of Dirs to Compress: $Dir  #"
echo "#####"

Dir_name=$(echo "$Dir" | tr '/' '-')
echo "#####"
echo "## The dir name replazing / for - is: $Dir_name  "
echo "#####"

#curso@ubuntu:~/temal0$ ls -lR /home/curso/ | grep '^d' | cut -d' ' -f9-13 | cut -d':' -f2
ls -ld $Dir* | cut -d' ' -f12-16 | cut -d'/' -f2-6 | sed '/^$/d' > Dir_list.txt

Date=$(date +%Y-%m-%d)
echo "#####"
echo "##### Date: $Date #####"
echo "#####"

tar -zcvf $Date.tar.gz Dir_list.txt
compress_f=$(ls -la | grep tar.gz | cut -d' ' -f8-12)
echo "#####"
echo "##### Your backups: #####"
echo "$Backup List"
echo "#####"

if [ test -f ./ext/Dir_list_ext.lst ]
then
echo "El fichero .lst YA EXISTE"
else
mkdir ext
tar -xzvf $Date.tar.gz -C ./ext
echo -e "Se creará fichero lst en base a la lista de Dirs del file en \n archivo /ext/Dir_list_ext
while IFS=' ' read -r line || [[ -n "$line" ]]; do
echo "$line," >> ./ext/Dir_list_ext.lst
done < Dir_list.txt
fi

```

Then the OUTPUT:

```

curso@ubuntu:~/temal0$ ls
2015-12-08-tmp.tar.gz  actividad1.sh  actividad2.sh  actividad3.sh
curso@ubuntu:~/temal0$ ./actividad3.sh
Input absolute path DIR to get a List of
subdirs in a .txt and compress this file list: /home/curso/
#####
# List of Dirs to Compress: /home/curso/  #
#####
#####
## The dir name replazing / for - is: -home-curso-

```



```
#####
#####
##### Date: 2015-12-08 #####
#####
Dir_list.txt
#####
##### Your backups: #####
List
#####
./actividad3.sh: line 41: [: -f: binary operator expected
Dir_list.txt
Se creará fichero lst en base a la lista de Dirs del file en
archivo /ext/Dir_list_ext.lst
curso@ubuntu:~/tema10$ ls
2015-12-08.tar.gz 2015-12-08-tmp.tar.gz actividad1.sh actividad2.sh actividad3.sh Dir_list.tx
curso@ubuntu:~/tema10$ cd ext/
curso@ubuntu:~/tema10/ext$ ls
Dir_list_ext.lst Dir_list.txt
curso@ubuntu:~/tema10/ext$ cat Dir_list.txt
home/curso/ArchivoA.txt
home/curso/ArchivoB.txt
home/curso/BarrireroExequiel
home/curso/dead.letter
home/curso/Desktop
home/curso/Documents
home/curso/Downloads
home/curso/group -> /etc/group
home/curso/Music
home/curso/Pictures
home/curso/Public
home/curso/root
home/curso/Software
home/curso/tema10
home/curso/tema7
home/curso/tema8
home/curso/Templates
home/curso/Videos
curso@ubuntu:~/tema10/ext$ cat Dir_list_ext.lst
home/curso/ArchivoA.txt,
home/curso/ArchivoB.txt,
home/curso/BarrireroExequiel,
home/curso/dead.letter,
home/curso/Desktop,
home/curso/Documents,
home/curso/Downloads,
home/curso/group -> /etc/group,
home/curso/Music,
home/curso/Pictures,
home/curso/Public,
home/curso/root,
home/curso/Software,
home/curso/tema10,
home/curso/tema7,
home/curso/tema8,
home/curso/Templates,
home/curso/Videos,
curso@ubuntu:~/tema10/ext$
```

8.4. Actividad 4.

8.4.1. head command

Explique la siguiente línea de comando:

```
if [ -e $HOME/.configuracion -a ! -e SinConfiguracion]
```

8.5. Actividad 5 (ejercitación evaluación).

8.5.1. script: find directories by name - for - if - test - ls

Crear un script en bash que imprima en un archivo llamado, lista_directorios, el nombre de aquellos archivos que solo sean directorios. Utilizar un ls que recorra el directorio /home/usuario (cualquier usuario). Ayuda: comandos a utilizar for, if, test y ls.

8.6. Actividad 6 (ejercitación evaluación).

8.6.1. script: True or False - for - if - test - ls

Crear un script en bash que indique "Verdadero o Falso" si la cantidad de dispositivos de caracteres especiales es mayor que la cantidad de dispositivos de bloques existentes en el directorio /dev. Ayuda: comandos a utilizar for, if, test y ls.

9. Ejercicios Tema 11: Configuración de Bash

- Variables de bash.
- Variables de entorno.
- Alias.
- Cómo expande bash una línea de comandos.
- Prevenir la expansión.
- Diferencias entre shell con y sin login.
- Tareas de ejecución al inicio de un bash.
- Tareas de ejecución al final de un bash.

9.1. Actividad 1.

9.1.1. Variable de entorno PS1

Explore la utilidad de la variable de entorno PS1 y escriba en un archivo de texto el uso básico de esa variable con 2 ejemplos.

9.2. Actividad 2.

9.2.1. Limpiar History y /tmp

Configure su usuario de sistema para que limpie el historial y el contenido de la carpeta /tmp al salir del sistema.

9.3. Actividad 3.

9.3.1. init.d and processes

Configure su usuario de sistema para que al inicio imprima en la terminal los 3 procesos que más ocupan memoria.

9.4. Actividad 4.

9.4.1. backup script - tar gz

Cree un comando de bash llamado "backup_personal", que realice un tar gz de la carpeta home del usuario con la fecha actual e imprima la salida en el archivo /tmp/backup.log

9.5. Actividad 5.

9.5.1. Alias for grep command

Cree un alias para el comando grep que permita por defecto pintar las ocurrencias exitosas de grep , al ejecutar el comando grep.

9.6. Actividad 6.

9.6.1. script: given a value to a variable

Realize un script que imprima en una misma linea la siguiente frase, teniendo en cuenta que la palabra producto proviene de la variable producto=parrilla. "Precio de lista de producto \$100"

Bibliografía

Asociación Cultural U.T.L.A.I. (Cantabria España) (2009). Enlaces de ficheros: enlaces duros y enlaces simbólicos. <http://www.nodo50.org/utlai/linux/27010712.htm>. [03-10-2015].

Debian org (2014). sudo. <https://wiki.debian.org/sudo>. [20-09-2015].

DigitalOcean™ Inc. (2015). How to use find and locate to search for files on a linux vps. <https://www.digitalocean.com/community/tutorials/how-to-use-find-and-locate-to-search-for-files-on-a-linux-vps>. [03-10-2015].

Hogbin E., Komarinski M., M. D. (2005a). The linux documentation project: A brief introduction to regular expressions. <http://tldp.org/LDP/abs/html/x17129.html#FTN.AEN17134>. [01-11-2015].

Hogbin E., Komarinski M., M. D. (2005b). The linux documentation project: The command-line history. <http://www.tldp.org/LDP/GNU-Linux-Tools-Summary/html/x1712.htm>. [02-11-2015].

Hosting News LLC (2010). Dedicated server school: Using the linux “history” command. <http://www.serverschool.com/dedicated-servers/using-the-linux-history-command/>. [02-11-2015].

Linux Kernel Organization Inc. (2014a). The linux kernel archives. <https://www.kernel.org>. [16-09-2015].

Linux Kernel Organization Inc. (2014b). The linux kernel archives: Frequently asked questions. <https://www.kernel.org/category/faq.html>. [16-09-2015].

Mirizio E. (2015a). Guía teórica - diplomatura linux modulo 1. <http://www.di.artec.unc.edu.ar/course/view.php?id=4§ion=2>. [20-09-2015].

Mirizio E. (2015b). Guía teórica - diplomatura linux modulo 7. http://www.di.artec.unc.edu.ar/pluginfile.php/8003/mod_resource/content/1/Tips_Scripts.txt. [06-12-2015].

Natarajan R. (2015). Tecmint: The power of linux “history command” in bash shell. <http://www.tecmint.com/history-command-examples/>. [02-11-2015].

Negus, C. (2015). *Linux Bible*. Wiley, 9 edition.

Osamu Aoki (2013). Debian reference - chapter 1. gnu/linux tutorials - 1.1.6. virtual consoles. https://www.debian.org/doc/manuals/debian-reference/ch01.en.html#_virtual_consoles. [19-09-2015].

Red Hat, I. (2015). Fedora project wiki - what is the fedora project? https://fedoraproject.org/wiki/Fedora_Project_Wiki. [16-09-2015].

Scheifler B., G. J. xhost man page. <http://www.x.org/archive/X11R7.5/doc/man/man1/xhost.1.html>. [19-09-2015].

Shrestha N. (2009). Unix sed tutorial: Find and replace text inside a file using regex. <http://www.thegeekstuff.com/2009/09/unix-sed-tutorial-replace-text-inside-a-file-using-substitute-command/>. [07-12-2015].

Shrestha N. (2013a). 10 practical linux cut command examples to select file columns. <http://www.thegeekstuff.com/2013/06/cut-command-examples/>. [07-12-2015].

Shrestha N. (2013b). How to sort files in linux using sort command. <http://www.thegeekstuff.com/2013/04/sort-files/>. [07-12-2015].

Vivek G. (2006). How do i compress a whole linux or unix directory. <http://www.cyberciti.biz/faq/how-do-i-compress-a-whole-linux-or-unix-directory>. [04-10-2015].

Wikipedia Project (2015a). Filesystem hierarchy standard. https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard. [20-09-2015].

Wikipedia Project (2015b). Linux kernel. https://en.wikipedia.org/wiki/Linux_kernel#2.6.x.y_releases. [16-09-2015].

Wikipedia Project (2015c). What is the fedora project? [https://es.wikipedia.org/wiki/N%C3%BAcleo_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)). [16-09-2015].

Siglas y Acrónimos

cd change directory

EOL End of Life

GNU GNU's Not Unix!

Gnome GNU Network Object Model Environment

GTK Graphical user interface toolkit

GUI Grafic User Interface

KDE K desktop environment

sudo Super-user do