

Mar 10, 14 1:28

aptitude.txt

Page 1/3

```
# Para dar mantenimiento a los paquetes del sistema

# aptitude update

# aptitude safe-upgrade

# aptitude clean


# Para hacer una actualización a todos los paquetes del sistema y tal vez
actualizar la versión

# aptitude full-upgrade


# Para instalar un paquete en particular

# aptitude install paquete


# Para borrar el cache local de paquetes

# aptitude clean


# Para remover completamente un paquete incluyendo los paquetes dependientes

# aptitude purge paquete
# aptitude --purge remove paquete


# Para buscar paquetes que son potencialmente instalables

# aptitude search nombre


# Para conocer las características de un paquete a instalar

# aptitude show nombre


# Para conocer todos los paquetes instalados

# dpkg -l
```

Algunas veces, si se interrumpe el proceso de instalación, es posible corregir este proceso mediante:

```
# dpkg --configure -a
Para completar las instalaciones pendientes usar
# aptitude -f safe-upgrade
```

Configuración del archivo para Debian testing: /etc/apt/sources.list

```
deb http://mmc.geofisica.unam.mx/debian/ testing main contrib non-free
deb-src http://mmc.geofisica.unam.mx/debian/ testing main contrib non-free

deb http://security.debian.org/ testing/updates main contrib non-free
deb-src http://security.debian.org/ testing/updates main contrib non-free
```

Configuración del archivo para Debian estable: /etc/apt/sources.list

```
deb http://mmc.geofisica.unam.mx/debian/ stable main contrib non-free
deb-src http://www.mmc.geofisica.unam.mx/debian/ stable main contrib non-free
```

Mar 10, 14 1:28

aptitude.txt

Page 2/3

```
deb http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
```

Conocer cual es el mirror de Debian que nos de la mejor velocidad de descarga podemos usar apt-spy, se instala:

```
# aptitude install apt-spy
```

Bajar la lista de mirror usando wget

```
$ wget http://www.mmc.geofisica.unam.mx/debian/README.mirrors.txt
```

para luego correrlo usando

```
$ apt-spy -d stable -s mx -m ./README.mirrors.txt -o apt-spy.list
```

el cual generara un archivo apt-spy.list en nuestro home con la configuración óptima además genera una salida indicando la velocidad de acceso, como la siguiente:

```
SERVER: ftp.mx.debian.org
Benchmarking FTP...
Downloaded 7815450 bytes in 0.82 seconds
Download speed: 9255.63 kB/sec

SERVER: debian.unam.mx
Benchmarking FTP...
Downloaded 7815450 bytes in 0.80 seconds
Download speed: 9537.88 kB/sec

SERVER: mmc.geofisica.unam.mx
Benchmarking FTP...
Downloaded 8794225 bytes in 0.20 seconds
Download speed: 42400.16 kB/sec
```

Para conocer las 10 replicas de debian con mejor velocidad de descarga, usar:

```
# aptitude install netselect-apt
# netselect-apt stable
```

En algun momento, ciertos paquetes pueden quedar huérfanos y no ser necesarios en el sistema, para conocer estos, es necesario instalar deborphan

```
# aptitude install deborphan
para luego correrlo mediante
# orphaner
```

donde es posible conocer los paquetes huérfanos y liberarlos.

Usando: netselect-apt

```
# apt-get install netselect-apt
para luego correrlo usando
$ netselect-apt stable
```

el cual generara un archivo sources.list en nuestro home con la configuración óptima

Tasksel

Múltiples configuraciones de paquetes pueden instalarse desde tasksel, tasksel contiene metapaquetes que facilitan la instalación de un sistema Debian, esto es recomendado cuando se tiene poco conocimiento sobre la instalación de

Mar 10, 14 1:28

aptitude.txt

Page 3/3

```
paquetes en Debian, muy usado cuando se hace la instalación de un sistema
Debian con instalación mínima, i.e. sin ambiente gráfico
para hacer uso de este recurso, hacer:
# tasksel
para conocer los metapaquetes instalados y que se pueden instalar, usar:
# tasksel --list-tasks
para conocer los paquetes que contiene un metapaquete
# tasksel --task-packages web-server
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Mar 10, 14 1:28

apt.txt

Page 1/2

```
# Para dar mantenimiento a los paquetes del sistema
# apt-get update
# apt-get check
# apt-get upgrade

# Para hacer una actualización a todos los paquetes del sistema
# apt-get dist-upgrade

# Para instalar un paquete en particular
# apt-get install paquete

# Para borrar el cache local de paquetes
# apt-get clean

# Para remover completamente un paquete incluyendo los paquetes dependientes
# apt-get --purge remove paquete

# Para buscar paquetes que son potencialmente instalables
# apt-cache search nombre
# apt-cache pkgnames

# Para conocer las características de un paquete a instalar
# apt-cache show nombre
# apt-cache showpkg nombre

# Para conocer todos los paquetes instalados
# dpkg -l
```

Algunas veces, si se interrumpe el proceso de instalación, es posible corregir este proceso mediante:

```
# dpkg --configure -a
```

Configuración del archivo para Debian testing: /etc/apt/sources.list

```
deb http://mmc.geofisica.unam.mx/debian/ testing main contrib non-free
deb-src http://mmc.geofisica.unam.mx/debian/ testing main contrib non-free

deb http://security.debian.org/ testing/updates main contrib non-free
deb-src http://security.debian.org/ testing/updates main contrib non-free
```

Configuración del archivo para Debian estable: /etc/apt/sources.list

```
deb http://mmc.geofisica.unam.mx/debian/ stable main contrib non-free
deb-src http://www.mmc.geofisica.unam.mx/debian/ stable main contrib non-free
```

Mar 10, 14 1:28

apt.txt

Page 2/2

```
deb http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
```

Conocer cual es el mirror de Debian que nos de la mejor velocidad de descarga podemos usar apt-spy, se instala:

```
# apt-get install apt-spy
```

para luego correrlo usando

```
$ apt-spy -d stable -s mx -m ./mirror.txt -o apt-spy.list
```

el cual generara un archivo mirror.txt en nuestro home con la configuración óptima además genera una salida indicando la velocidad de acceso, como la siguiente:

```
SERVER: ftp.mx.debian.org
Benchmarking FTP...
Downloaded 7815450 bytes in 0.82 seconds
Download speed: 9255.63 kB/sec
```

```
SERVER: debian.unam.mx
Benchmarking FTP...
Downloaded 7815450 bytes in 0.80 seconds
Download speed: 9537.88 kB/sec
```

```
SERVER: mmc.geofisica.unam.mx
Benchmarking FTP...
Downloaded 7811970 bytes in 0.20 seconds
Download speed: 38442.89 kB/sec
```

Para conocer las 10 replicas de debian con mejor velocidad de descarga, usar:

```
# aptitude install netselect-apt
# netselect-apt stable
```

En algun momento, ciertos paquetes pueden quedar huérfanos y no ser necesarios en el sistema, para conocer estos, es necesario instalar deborphan

```
# apt-get install deborphan
para luego correrlo mediante
# orphaner
```

donde es posible conocer los paquetes huérfanos y liberarlos.

Tasksel

Múltiples configuraciones de paquetes pueden instalarse desde tasksel, tasksel contiene metapaquetes que facilitan la instalación de un sistema Debian, esto es recomendado cuando se tiene poco conocimiento sobre la instalación de paquetes en Debian, muy usado cuando se hace la instalación de un sistema Debian con instalación mínima, i.e. sin ambiente gráfico para hacer uso de este recurso, hacer:

```
# tasksel
para conocer los metapaquetes instalados y que se pueden instalar, usar:
# tasksel --list-tasks
para conocer los paquetes que contiene un metapaquete
# tasksel --task-packages web-server
```

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:08

ayudaLinuxColor.txt

Page 1/1

Algunas veces es necesario usar la ayuda de Linux en línea de comandos y es difícil leerla en monocolor, por eso existe most, que pone la ayuda de comandos como man en colores para facilitar su lectura

Para instalar usar:
aptitude install most

Para actualizar el paginador de ayuda usar
update-alternatives --config pager

Que le mostrara una salida como:

There are 5 choices for the alternative pager (providing /usr/bin/pager).

Selection	Path	Priority	Status
* 0	/bin/less	77	auto mode
1	/bin/less	77	manual mode
2	/bin/more	50	manual mode
3	/usr/bin/most	60	manual mode
4	/usr/bin/pg	10	manual mode
5	/usr/bin/w3m	25	manual mode

Press enter to keep the current choice[*], or type selection number:

Y seleccionar most (3).

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:07

ayuda.txt

Page 1/1

Estas son algunas páginas en español que tienen manuales y páginas de comandos para linux

<http://www.mmc.geofisica.unam.mx/Replicas/LuCAS/>

<http://www.esdebian.org/wiki/linea-comandos>

<http://www.esdebian.org/wiki>

<http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/administracion-avanzada-del-sistema-operativo-gnu-linux/materiales/>

Páginas de Debian en el mundo

<http://www.debian.org>

<http://www.debian-mx.com/>

<http://www.esdebian.org/>

<http://www.debian-news.net/>

<http://news.debian.net/>

<http://www.debiantoday.com/>

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Mar 10, 14 5:12

compactadores.txt

Page 1/2

Para manejar de manera óptima y fácil un conjunto de archivos es necesario agruparlos y/o compactarlos, existen múltiples programas para tal efecto como TAR, GZIP, BZ2, ZIP, LHA, ARJ, ZOO, RAR y su uso básico es el siguiente:

Para instalar los programas más comunes usar:

```
# aptitude install gzip unzip zip pbzip2 bzip2 lbzip2 arj zoo unrar lhasa ncompress \
p7zip p7zip-full p7zip-rar unace unace-nonfree zutils cpio pax unp lzma xz-utils pxz \
pixz kgb rar rarcrack zpaq bsdtar
```

Para respaldar un grupo de archivos y/o directorios al formato TAR

```
$ tar -cvf nombre.tar directorio
```

Para restaurar en formato TAR

```
$ tar -xvf nombre.tar
```

Para respaldar y compactar un grupo de archivos y/o directorios al formato TAR.GZ

```
$ tar -cvf nombre.tar directorio
```

```
$ gzip --best nombre.tar
```

O usar

```
$ tar -zcvf nombre.tar.gz directorio
```

Para restaurar en formato TAR.GZ

```
$ gunzip nombre.tar.gz
```

```
$ tar -xvf nombre.tar
```

O usar

```
$ tar -zxvf nombre.tar.gz
```

Para compactar un archivo o grupos de archivos y/o directorios:

Con GZIP comprimir y descomprimir respectivamente

```
$ gzip fichero
```

```
$ gzip -d fichero.gz
```

Con BZ2 comprimir y descomprimir respectivamente

```
$ bzip fichero
```

```
$ bzip -d fichero.bz2
```

Con ZIP comprimir y descomprimir respectivamente

```
$ zip archivo.zip ficheros
```

```
$ unzip archivo.zip
```

Con LHA comprimir y descomprimir respectivamente

```
$ lha a archivo.lha ficheros
```

```
$ lha x archivo.lha
```

Con ARJ comprimir y descomprimir respectivamente

```
$ arj a archivo.arj ficheros
```

```
$ arj x archivo.arj
```

```
$unarj archivo.arj
```

Con ZOO comprimir y descomprimir respectivamente

```
$ zoo a archivo.zoo ficheros
```

```
$ zoo x archivo.zoo
```

Con RAR comprimir y descomprimir respectivamente

```
$ rar a archivo.rar ficheros
```

```
$ rar x archivo.rar
```

```
$ unrar archivo.rar
```

En algunos casos, archivos RAR de Windows no es posible descomprimirlos correctamente en Linux, para descomprimirlos podemos descargar utilerías GNU para Win32:

<http://unxutils.sourceforge.net/>

entre ellas unrar (es un sólo archivo zip) de la dirección:

<http://sourceforge.net/projects/unxutils/>

ahora usando Wine es posible descomprimir los archivos desde Linux mediante:

Mar 10, 14 5:12

compactadores.txt

Page 2/2

```
$ wine unrar.exe e archivo.rar
```

Descomprimir archivos RAR o ZIP rotos, usar

```
$ unrar e -kb -y nombreArchivo.rar
```

o

```
$ bsdtar xf nombreArchivo.zip
```

Mas opciones de uso de pueden encontrar usando el comando MAN, por ejemplo

```
$ man zip
```

Existe un programa UNP que permite descomprimir de casi cualquier formato:

```
$ unp archivo.compactado
```

Cuando se tiene una lista de archivos de distintas trayectorias o es resultado de una búsqueda, para compactar es preferible usar AFIO. instalar AFIO usando:

```
# aptitude install afio
```

Para compactar digamos todos los archivos *.?pp (fuentes de C++) usar:

```
$ find . -name *.?pp | afio -o -Z fuentes
```

Para descompactarlos, usar:

```
$ afio -i -Z fuentes
```

Si se desea compactar usando GZIP, usar:

```
$ cat lista | afio -o -Z -G 9 fuentes
```

Si se desea ver el listado de archivos que contiene fuentes, usar:

```
$ afio -t fuentes
```

Si se desea compactar y mandar a otra máquina usar:

```
$ find . -name *.?pp | afio -o -Z user@servidor%ssh:fuentes
```

Como el uso de AFIO no necesita extensión en el archivo, para descompactarlo de cualquier formato es recomendable usar UNP, este escoge el mejor método para el archivo en cuestión:

```
$ unp fuentes
```

Una vez compactado el archivo, para transmitirlo de una máquina a otra, se puede hacer uso del programa scp, esta permite transmitir de manera segura archivos y/o directorios entre máquinas usando el protocolo del programa SSH, existen versiones tanto para Windows como para Linux.

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:14

convertirAudioVideo.txt

Page 1/1

```
Para convertir audio o vídeo de un formato a otro, instalar ffmpeg
# aptitude install ffmpeg
por ejemplo, para convertir de AMP a flac
$ ffmpeg -i ach.amr arch.flac
para convertir de flac a MP3
$ ffmpeg -i arch.flac arch.mp3
para convertir de AVI a MP$
$ ffmpeg -i arch.avi arch.mp4
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Oct 21, 15 7:37

crontab.txt

Page 1/2

Cuando se requieren hacer tareas periódicas en Linux, una herramienta indispensable es el CRONTAB, este archivo tiene la siguiente estructura:

```

.----- minuto (0 - 59)
|
| .----- hora (0 - 23)
| |
| | .----- día del mes (1 - 31)
| | |
| | | .----- mes (1 - 12) O jan,feb,mar,apr ... (los meses en ingles)
| | | |
| | | | .----- día de la semana (0 - 6) (Domingo=0 - 7) O sun,mon,tue,wed,thu,fri,sat
| | | | |
| | | | |
* * * * * comando para ser ejecutado

```

Por ejemplo: Se pone a cero el log de error de Apache un minuto después de medianoche (00:01 de cada día del mes, de cada día de la semana)

```
1 0 * * * usr echo -n "" > /www/apache/logs/error_log
```

Otro ejemplo: Se ejecuta el script: /home/user/test.pl cada 5 minutos

```
*/5 * * * * usr /home/user/test.pl
```

Otro ejemplo: Ejecuta la orden who todos los lunes cada media hora y guarda la salida en el fichero quien.tex

```
0,30 * * * 1 usr /usr/bin/who >> /home/quien.tex
```

Hay varios valores predefinidos que se pueden utilizar para sustituir la expresion CRON.

Entrada	Descripcion	Equivale A
@yearly	Se ejecuta una vez al año	0 0 1 1 *
@annually	(igual que @yearly)	0 0 1 1 *
@monthly	Se ejecuta una vez al mes	0 0 1 * *
@weekly	Se ejecuta una vez a la semana	0 0 * * 0
@daily	Se ejecuta una vez al día	0 0 * * *
@midnight	(igual que @daily)	0 0 * * *
@hourly	Se ejecuta una vez cada hora	0 * * * *

También esta disponible @reboot, que permite a un trabajo ejecutarse una vez cada vez que el demonio cron se inicie, que eso típicamente coincidir con el arranque del servidor. Puede ser útil si es necesario levantar un servidor o demonio bajo un usuario en particular o si el usuario no tiene permisos al archivo rc.d/init.d.

A CRON expression is a string comprising 5 or 6 fields separated by white space that represents a set of times, normally as a schedule to execute some routine.

Format

Field name	Mandatory?	Allowed values	Allowed special characters
Minutes	Yes	0-59	* / , -
Hours	Yes	0-23	* / , -
Day of month	Yes	1-31	* / , - ? L W
Month	Yes	1-12 or JAN-DEC	* / , -
Day of week	Yes	0-6 or SUN-SAT	* / , - ? L #
Year	No	1970-2099	* / , -

In some uses of the CRON format there is also a seconds field at the beginning of the pattern.

Special characters

Asterisk (*)

The asterisk indicates that the cron expression will match for all values of the field; e.g., using an asterisk in the 4th field (month) would indicate every month.

Slash (/)

Slashes are used to describe increments of ranges. For example 3-59/15 in the 1st field (minutes) would indicate the 3rd minute of the hour and every 15 minutes thereafter. The form "*/..." is equivalent to the form "first-last/...", that is, an increment over the largest possible range of the field.

Oct 21, 15 7:37

crontab.txt

Page 2/2

Percent (%)

Percent-signs (%) in the command, unless escaped with backslash (\), will be changed into newline characters, and all data after the first % will be sent to the command as standard input.

Comma (,)

Commas are used to separate items of a list. For example, using "MON,WED,FRI" in the 5th field (day of week) would mean Mondays, Wednesdays and Fridays.

Hyphen (-)

Hyphens are used to define ranges. For example, 2000-2010 would indicate every year between 2000 and 2010 CE inclusive.

L

'L' stands for "last". When used in the day-of-week field, it allows you to specify constructs such as "the last Friday" ("5L") of a given month. In the day-of-month field, } it specifies the last day of the month.

W

The 'W' character is allowed for the day-of-month field. This character is used to specify the weekday (Monday-Friday) nearest the given day. As an example, if you were to specify "15W" as the value for the day-of-month field, the meaning is: "the nearest weekday to the 15th of the month". So if the 15th is a Saturday, the trigger will fire on Friday the 14th. If the 15th is a Sunday, the trigger will fire on Monday the 16th. If the 15th is a Tuesday, then it will fire on Tuesday the 15th. However if you specify "1W" as the value for day-of-month, and the 1st is a Saturday, the trigger will fire on Monday the 3rd, as it will not 'jump' over the boundary of a month's days. The 'W' character can be specified only when the day-of-month is a single day, not a range or list of days.

Hash (#)

'#' is allowed for the day-of-week field, and must be followed by a number between one and five. It allows you to specify constructs such as "the second Friday" of a given month.

Question mark (?)

Note: Question mark is a non-standard character and exists only in some cron implementations. It is used instead of '*' for leaving either day-of-month or day-of-week blank.

 Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Jan 20, 14 9:50

cryptsetup.txt

Page 1/3

Cifrado de particiones mediante cryptsetup

Instalar paquetes necesarios usando

```
# aptitude install cryptsetup scrub
```

Respalidar las carpetas de usuario

Nota: estos pasos son necesarios si /home está en la partición que vas a cifrar, si reparticionaste o tienes tu home en una partición distinta a la del cifrado (por ejemplo si instalaste todo el sistema bajo una partición), puedes saltar hasta "Creación de la partición cifrada".

Crear un punto de montaje para /home en tanto ciframos la partición para home

```
# mkdir /opt/home
```

Editar el punto de montaje para /home en las particiones montadas al inicio del sistema

```
# nano /etc/fstab
```

Si hay una línea /home, la comentamos escribiendo un # al inicio. Al final del archivo podemos insertar una nueva línea, sencilla:

```
/opt/home      /home  none      bind
```

El "none" es porque no estamos usando "otro" sistema de archivos, tan sólo queremos "ligar" -bind- el punto /opt/home a /home, mientras ciframos una partición para /home.

Salimos de nano guardando con Control-O (de output) y Control-X

Copiamos todo bajo el /home actual al espacio provisional

```
# cp -a /home /opt/home
```

Reinicia el equipo

Creación de la partición cifrada

(Substituye X por la partición que representa tu futuro home cifrado)

Primero borramos todos los datos previamente existentes

```
# scrub -p random /dev/sdaX
```

Nota: en la mayoría de tutoriales de seguridad encontrarás que el borrado previo al cifrado se hace con dd if=/dev/random of=/dev/sdaX sin embargo scrub es mucho más rápido y además maneja otros algoritmos de borrado seguro como el Gutmann. Si quieres hacer pruebas con ambos y reportarnos cual fue más rápido en tu sistema, sólo anota el resultado de time. Ejemplo: # time scrub -p random /dev/sda5

Ahora inicializamos la partición objetivo

```
# cryptsetup -v -y -c aes-xts-plain -s 512 luksFormat /dev/sdaX
```

Creamos el sistema de archivos en la partición cifrada. Pero primero hay que abrirla con la clave que ingresamos

```
# cryptsetup luksOpen /dev/sdaX home
```

Verificamos que esté presente:

Jan 20, 14 9:50

cryptsetup.txt

Page 2/3

```
# dmsetup ls
```

Creamos el sistema de archivos cifrado y un punto de montaje

```
# mkfs.ext4 /dev/mapper/home
# mkdir /opt/home_crypt
```

Lo montamos y verificamos que está presente

```
# mount /dev/mapper/home /opt/home_crypt
# df -Th | grep crypt
```

Copiamos las carpetas de usuario a la nueva partición cifrada

```
# cp -a /opt/home /opt/home_crypt
```

Desmontar y desactivar el sistema de archivos

```
# umount /opt/home_crypt
# cryptsetup luksClose /dev/sdaX
```

Si se desea montar nuevamente el sistema de archivos encriptado usar

```
# cryptsetup luksOpen /dev/sdaX home
# mount /dev/mapper/home /opt/home_crypt
```

Por último, preparamos el sistema para montar la partición cifrada al reiniciar

Agregue la partición cifrada a /etc/crypttab

```
# nano /etc/crypttab
```

Incluya esta entrada:

```
home /dev/sdaX none luks
```

La entrada en /etc/crypttab hace que su computadora le pida su frase de acceso luks al arrancar

Edite su /etc/fstab

```
# nano /etc/fstab
```

Comenté la entrada actual de /home para referencia insertando un # al inicio de la línea, y agregue la nueva partición cifrada:

```
/dev/mapper/home /home ext4 defaults 1 2
```

Reinicie la máquina y verifique que su nuevo home está montado y su espacio libre

```
# mount /home
# df -Th
```

Limpieza final

Tras reiniciar el sistema si todo funciona correctamente y sus archivos de usuarios son accesibles, ahora es preciso borrar los respaldos antes hechos en claro. No tendría sentido realizar el cifrado de su partición /home si todavía queda su rastro en claro en otra parte.

Borramos el directorio que usamos previamente como respaldo

```
# rm -Rf /opt/home
```

Hacemos un borrado seguro de la información aún presente en disco

```
# scrub -X -p dod /opt/home
```

Jan 20, 14 9:50

cryptsetup.txt

Page 3/3

```
# rm -Rf /opt/home
```

Recuerde: los sistemas operativos normalmente no "borran" un archivo, sólo olvidan preservar el espacio que ocupaban, para que no sean ocupados por otros archivos. Esto permitiría a alguien incluso sin grandes conocimientos técnicos acceder a información sensible.

Si usted prefiere el borrado de mayor seguridad disponible en scrub, aplicable a archivos individuales, pruebe con el patrón Gutmann:

```
scrub -p gutmann archivo_sensible
```

Este borrado es útil en discos magnéticos, pero en dispositivos de estado sólido como USBs y memorias SD acortará en mucho la vida de su dispositivo, sin siquiera garantizar el borrado total. En este caso lo más recomendable es que la información JAMÁS llegue en claro (sin cifrar) a un medio de almacenamiento. Por ello vale la pena pensar de antemano en la necesidad de proteger su información sensible (como claves de acceso, datos personales, etc.)

Nota: Si desea solo montar la información de un disco encriptado en otra máquina o en la misma, sólo es necesario usar:

```
# cryptsetup luksOpen /dev/sdaX home
# mount /dev/mapper/home /opt/home_crypt
```

esto es muy util si se esta accediendo desde una unidad desmontable, como un USB o CDROM.

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:26

documentacionCodigoFuente.txt

Page 1/6

Cuando se crea código fuente, una buena práctica es documentarlo y si esta documentación básica se puede utilizar para generar un manual de referencia del código en HTML, PDF, PS, o XML a partir de los fuentes con unos cuantos comandos de texto en menos de un minuto, pues que mejor.

Existen varias herramientas para ello, una de ellas es DOXYGEN para códigos de java, fortran, C y C++. Para instalar esta herramienta usar:

```
# aptitude install doxygen graphviz
```

una vez instalada, hay que generar el archivo de configuración de DOXYGEN, para ello usar:

```
$ doxygen -g
```

de aquí podemos editar archivo Doxyfile generado según las necesidades de la documentación a generar, un ejemplo de ella para generar la salida en html, latex y XML se pone en el archivo Doxyfile en esta página.

Para generar la documentación de los fuentes en la carpeta donde este el archivo de configuración y los archivos fuentes, usar:

```
$ doxygen
```

La documentación generada con DOXYGEN se mostrara en carpetas separadas para cada una de las salidas seleccionadas por ejemplo: html, latex, xml, etc.

Para ver la documentación generada para html, usar en ambiente texto por ejemplo

```
$ cd html
$ xpdf index.html
```

Para generar la documentación en formato PDF a partir de la salida de

LaTeX usar

```
$ cd latex
$ make pdf
$ xpdf refman.pdf
```

en este caso se supone que se tiene instalado LaTeX en la máquina, en caso contrario instalarlo usando:

```
# aptitude install science-typesetting texlive-science
y adicionalmete si se quiere:
# aptitude install texmaker texmacs texmacs-extra-fonts texlive-latex-base texlive-latex-recommend
ed myspell-en-us myspell-es
```

Hay varios estilos de documentación, aquí pongo una que se me hace fácil de usar para códigos en C++.

```
#ifndef __test__
#define __test__

/// Descripcion breve de la clase.
/**
 * Descripcion detallada de la clase ...
 *
 * @author Antonio Carrillo
 * @date Winter 2010
 * @version 0.0.1
 * @bug No errors detected
 * @warning No warnings detected
 * @todo Exception handling
 */
class test
{
private:

    /// Descripcion breve.
    const char *nmClass;

    /**
     * Descripcion corta.
     *
     * Descripcion larga ...
     */
}
```



```

    * 0 = Dirichlet, 1 = Neumann (or Robin)
    */
    int bdType;

public:

    /**
    * Descripcion breve.
    *
    * Descripcion detallada ...
    *
    * Algo de latex ...
    *
    * \f[
    * |I_2|=\left| \int_{0}^T \psi(t)
    * \left\{
    * u(a,t)-
    * \int_{\gamma(t)}^a
    * \frac{d\theta}{k(\theta,t)}
    * \int_a^{\theta} c(\xi)u_t(\xi,t)\,d\xi
    * \right\} dt
    * \right|
    * \f]
    *
    *
    * @param[out] clas Descripcion del parametro de salida
    * @param[in] fun Descripcion del parametro de entrada
    */
    test(const char *clas, const char *fun)
    {
        nameClassFunct(clas, fun);
    }

    /**
    * Descripcion breve.
    *
    * Descripcion detallada
    *
    * @param nVert Descripcion del parametro
    * @param[in] g Descripcion del parametro
    * @param[in] me Descripcion del parametro
    * @param[out] values Descripcion del parametro
    * @param z Descripcion del parametro
    * @return Descripcion de lo que regresa
    */
    int eval(int nVert, double **g, StdElem *me, double ***values, double *z);

};

/**
* Descripcion breve de la clase.
*
* Descripcion detallada de la clase
*
* Otro parrafo de la descripcion ...
*
* Algo de formulas con latex
*
* \f{eqnarray*}{
* g &=& \frac{Gm_2}{r^2} \\\
* &=& \frac{(6.673 \times 10^{-11})\,\mathrm{m}^3\,\mathrm{kg}^{-1}\,\mathrm{s}^{-2}}{(5.9736 \times 10^{24})\,\mathrm{kg}}\{(6371.01\,\mathrm{km})^2 \\\
* &=& 9.82066032\,\mathrm{m/s}^2
* \f}
*
*
* Documentacion sobre la cual se basa la clase o archivo(s) que hagan una descripcion de la

```

Feb 27, 13 20:26

documentacionCodigoFuente.txt

Page 3/6

```

* misma: Archivo.doc
*
* Descripcion breve del ejemplo de uso de esta clase (este archivo se supone que estara en
* una carpeta de nombre ./Examples en la actual posicion del código)
*
* Algo de latex
*
* The distance between \f$(x_1,y_1)\f$ and \f$(x_2,y_2)\f$ is \f$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}\f$
$.
*
* @example ExampleText.cpp
*/

#endif

```

Adicionalmente es deseable que algunos comportamientos o clases tengan información adicional como son: propósito, entradas, salidas, estructuras de datos usadas en entradas y salidas, dependencia de datos o ejecución, restricciones, etc., usando una estructura como la siguiente:

```

/**
 * Purpose and Method: [include the physics if appropriate]
 * Inputs
 * Outputs
 * TDS Inputs
 * TDS Outputs
 * Dependencies:
 * Restrictions and Caveats: e.g. "This doesn't work off-axis"
 */

```

Para controlar las versiones se podría usar algo como lo siguiente:

```

/**
 * @file release.notes
 * @brief Package TkrRecon
 * @verbatim
 * Coordinator: Leon Rochester
 *
 * v4r4p8 09-Mar-2002 LSR Remove GFxxxxx and SiRecObjs, no longer used
 * v4r4p7 07-Mar-2002 TU Mainly, add a combo vertexing to the TkrRecon sequence
 * @endverbatim
 */

```

Un ejemplo completo puede ser el siguiente:

```

*****
#ifndef __ErrorControl__
#define __ErrorControl__

#include <new>

using namespace std;

#include <stdlib.h>
#include "Printf.hpp"
#ifdef USE_HYPRE
#include <mpi.h>
#endif

/// Error Control, this class handles errors for the system RESSIM
/**
 * @author Antonio Carrillo and Gerardo Cisneros
 * @date Winter 2010
 * @version 0.0.2
 * @verbatim
 * Coordinator: Rober Yates
 * v0.0.1 January 2011 Antonio Carrillo generates the first version of the class
 * v0.0.2 March 2011 Gerardo Cisneros add HYPRE errors control

```

Feb 27, 13 20:26

documentacionCodigoFuente.txt

Page 4/6

```

Inputs: Name of class and function
Outputs: Exit of program
TDS Inputs: none
TDS Outputs: none
Dependencies: #ifdef USE_HYPRE, MPI package
Restrictions and Caveats: Non exception handling still
@endverbatim
* @bug No errors detected
* @warning No warnings detected
* @todo Exception handling
*/
class ErrorControl {
private:
    /// Name of class
    const char *nmClass;

    /// Name of function generating the error
    const char *nmFunction;

public:
    /**
     * Class Constructor
     */
    ErrorControl(void) {
        nameClassFunct(" ", " ");
    }

    /**
     * Class Constructor
     * @param clas Class name
     */
    ErrorControl(const char *clas) {
        nameClassFunct(clas, " ");
    }

    /**
     * Class Constructor
     * @param clas Class name
     * @param fun Name of function generating the error
     */
    ErrorControl(const char *clas, const char *fun) {
        nameClassFunct(clas, fun);
    }

    /**
     * Name of class and function
     * @param clas Class name
     * @param func Name of function generating the error
     */
    void nameClassFunct(const char * clas, const char *func) {
        nameClass(clas);
        nameFunct(func);
    }

    /**
     * No memory for this request
     * @param var Var name
     */
    void memoryError(const char * var) {
        Afprintf(stderr, "\n\nNo memory for %s request in %s of class %s\n\n", var, nmFunction, nmClass)
;
        fatalError(1);
    }

    /**
     * No memory for this request
     * @param var Var name

```

Feb 27, 13 20:26

documentacionCodigoFuente.txt

Page 5/6

```

    * @param i Index number
    */
    void memoryError(const char * var, int i) {
        Afprintf(stderr, "\n\nNo memory for %s request %d in %s of class %s\n\n", var, i, nmFunction, nm
Class);
        fatalError(1);
    }

    /**
    * No memory for this request
    * @param var Var name
    * @param func Name of function generating the error
    */
    void memoryError(const char * var, const char *func) {
        Afprintf(stderr, "\n\nNo memory for %s request in %s of class %s\n\n", var, func, nmClass);
        fatalError(1);
    }

    /**
    * Fatal error.
    * @param cod Error code
    */
    void fatalError(int cod) {
        Afprintf(stderr, "\nFatal Error\nEnd program\n");
#ifdef USE_HYPRE
        MPI_Abort(MPI_COMM_WORLD, cod);
#else
        exit(cod);
#endif
    }

    /**
    * Fatal error.
    * @param cod Error code
    */
    void fatalError(int cod, const char *txt) {
        Afprintf(stderr, txt);
        Afprintf(stderr, "\nFatal Error\nEnd program\n");
#ifdef USE_HYPRE
        MPI_Abort(MPI_COMM_WORLD, cod);
#else
        exit(cod);
#endif
    }

    /**
    * Set name of class
    * @param clas Class name
    */
    void nameClass(const char *clas) {
        nmClass = clas;
    }

    /**
    * Set name of function
    * @param func Function name
    */
    void nameFunct(const char *func) {
        nmFunction = func;
    }
};

/**
 * Error Control, this class handles errors for the system RESSIM
 *
 * Use of the class ErrorControl for error handling within the system RESSIM,
 * for example in the error control of memory request
 *
 * @example ExampleErrorControl.cpp
 */

```

Feb 27, 13 20:26

documentacionCodigoFuente.txt

Page 6/6

#endif

Más detalles sobre los parámetros en la documentación del código fuente para ser usada por DOXYGEN en:

<http://www.stack.nl/~dimitri/doxygen/commands.html#cmdparam>

<http://www.dealii.org/developer/doxygen/tutorial/index.html>

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:28

ejecucionProgramasWindowsEnLinux.txt

Page 1/1

Algunas veces es necesario correr aplicaciones de Windows en Linux, una forma es usando una máquina virtual. Otra es usar Wine que permite correr directamente el programa.

Instalación de wine
aptitude install wine

Correr aplicaciones, por ejemplo, para visualizar el contenido de un archivo con extensión CHM de ayuda, puede usarse
\$ wine hh.exe archivo.chm

Cuando no encuentra el archivo, es posible indicarle el path
\$ wine hh.exe 'winepath -w /algun/directorio/archivo.chm'

En algunos casos, archivos RAR de Windows no es posible descomprimirlos correctamente en Linux, para descomprimirlos podemos descargar utilerías GNU para Win32:

<http://unxutils.sourceforge.net/>
entre ellas unrar (es un solo archivo ZIP), de la dirección:
<http://sourceforge.net/projects/unxutils/>
ahora usando Wine es posible descomprimir los archivos desde Linux mediante:
\$ wine unrar.exe e archivo.rar

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:28

encfs.txt

Page 1/1

Encriptacion mediante FUSE, instalar

```
# aptitude install encfs afuse
```

Editar /etc/group agregando a los usuarios al grupo fuse

```
# nano /etc/group
fuse:x:119:antonio
```

Activar fuse en el kernel

```
# modprobe fuse
```

Revisar permisos fuse

```
# ls -al /dev/fuse
```

En caso de no tener permisos crw-wr-wr, cambiar usando

```
# chmod 666 /dev/fuse
```

Crear y montar directorio para guardar datos encriptados y los desencriptados

```
$ encfs ~/.encriptados ~/desencriptdos
```

Desmontar

```
$ fusermount -u ~/desencriptdos
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Jun 18, 14 4:19

formaterYmontarHD.txt

Page 1/1

Paquetes a instalar
#

Para conocer las particiones
fdisk -l
o usar
lsblk
también
blkid
o en su defecto
hwinfo --block --short

Ajustar la partición ... por ejemplo /dev/sdb
fdisk /dev/sdb
usar:
n Nueva particion
h Ayuda
w Escribir cambios

formatear la partición
mkfs.ext4 /dev/sdb1

Montar la partición
mkdir /mnt/tmp
mount /dev/sdb1

Desmontar la particion
umount /dev/sdb1

Dejar permanente el montaje de la partición
vi /etc/fstab

Forzar la recarga de particiones
mount -a

Otros comando usuales
fdisk, sfdisk, cfdisk, parted, lsblk, blkid, hwinfo

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:32

git.txt

Page 1/2

Para instalar GIT completo en el servidor o en máquina de trabajo

```
# aptitude install git-all
```

Instalar lo básico de GIT, en caso de no estar instalado

```
# aptitude install git
```

Preparar un directorio para el repositorio ya sea en el servidor o de forma local

```
$ mkdir example.git
```

```
$ cd example.git
```

Inicializar el repositorio

```
$ git --bare init
```

En la máquina de trabajo o en el servidor en cualquier carpeta se generar la estructura del repositorio

Generar el directorio temporal de trabajo para el repositorio

```
$ mkdir tmp
```

```
$ cd tmp
```

```
$ git init
```

Generar la estructura de trabajo para el repositorio y los archivos necesarios

```
$ mkdir branches release trunk
```

```
$ mkdir ...
```

Adicionar todo o cada uno de los archivos y carpetas

```
$ git add .
```

Subir los cambios

```
$ git commit -m "Texto"
```

Mandar al servidor

```
$ git remote add origin ssh://usr@máquina/~trayectoria
```

Mandar a un directorio local

```
$ git remote add origin ~/trayectoria
```

```
$ git push origin +master:refs/heads/master
```

Para usar el repositorio en cualquier otra máquina

Bajar el repositorio por primera vez

Del servidor

```
$ git clone ssh://usr@máquina/trayectoria
```

De carpeta local

```
$ git clone ~/trayectoria
```

Configurar algunos datos usados en el control de cambios

```
$ git config --global usr.name "Nombre"
```

```
$ git config --global usr.email usr@direccion
```

Actualizar del repositorio los cambios

```
$ git pull
```

Para subir los cambios al repositorio

```
$ git commit -a -m "mensaje"
```

```
$ git push
```

Comando usados para el trabajo cotidiano en GIT

Estado de los archivos locales

```
$ git status
```

Generar una nueva rama para trabajar en ella

```
$ git branch MiIdea
```

```
$ git checkout MiIdea
```

Unificar las ramas generadas en el punto anterior

```
$ git checkout master
```

```
$ git merge MiIdea
```

Borrar una rama

```
$ git branch -d MiIdea
```

Listar las ramas

```
$ git branch
```

Para ver los cambios en el repositorio

```
$ git log
```

o verlos en forma acortada

```
$ git log --pretty=oneline
```

Para recuperar un archivo de una actualización anterior

```
$ git show a30ab2ca64d81876c939e16e9dac57c8db6fb103:ruta/al/archivo > ruta/al/archivo.bak
```

Volver a una versión anterior

```
$ git reset --hard 56f8fb550282f8dfaa75cd204d22413fa6081a11
```

Feb 27, 13 20:32

git.txt

Page 2/2

```
para regresar el presente (cuidado con con subir cambios en ramas anteriores)
$ git pull
```

Si en algún momento borramos algo o realizamos cambios en nuestra máquina y necesitamos regresar los archivos como estaban en nuestra última actualización, podemos usar

```
$ git reset --hard HEAD
```

este trabaja con la información de nuestra copia local y no necesita conexión de red para la restitución.

Eventualmente es necesario optimizar la copia local de los archivos en Git, para ello podemos usar

```
$ git gc
```

Visualizador gráfico para GIT

```
# aptitude install gitk
```

Documentación del proyecto GIT puede ser consultada en

```
http://git-scm.com/documentation
```

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

```
http://www.mmc.geofisica.unam.mx/acl/
```

Jul 31, 13 13:57

gobby.txt

Page 1/1

Gobby es un editor colaborativo en tiempo real con licencia GPL que soporta múltiples documentos en una sola sesión multiusuario incluyendo un chat. Este corre sobre Windows, Mac OS X, Linux, etc.

Gobby es un programa muy sencillo de utilizar, en el que el usuario sólo tendrá que instalarlo en su ordenador en cualquier sistema operativo, y en la configuración encontrar el servidor al que está conectado el resto del grupo de usuarios que desean editar uno o más documentos de forma colaborativa.

Gobby no sólo tiene el cuadro de texto para editar colectivamente un documento que se vaya escribiendo a varias manos, si no también una sala de Chat integrada para que el usuario pueda comunicarse con el resto de usuarios involucrados en la redacción del texto.

Cualquier maquina con Gobby puede funjir como servidor, con la única condición que sea visible su IP (local u homologada) por todos los interezados en editar el documento.

Características:

- Es multiplataforma, existen versiones para Windows, Linux y Mac OS X
- A cada usuario se le asigna un color y puede ser cambiado en cualquier momento
- Se puede hacer que cambie automáticamente los tabuladores por espacios
- Resalta los corchetes y paréntesis coincidentes
- Muestra los números de línea
- Letra personalizable.
- Colorea el texto según la sintaxis del mismo
- Chat integrado
- UNDO para todo el texto (sólo en la versión 0.5)

Descargas del programa Windows, Mac OS X y Linux:
<http://gobby.0x539.de/trac/wiki/Download>

Para instalar Gobby como cliente en Linux Debian
aptitude install gobby

Para instalar Gobby como sevidor en Linux Debian
aptitude install gobby infinoted-0.5

En el caso del servidor de Gobby es necesario lanzar INFINOTED para poder controlar la edición y soportar el UNDO (solo en version 0.5 de Gobby). Para ello:

Lanzar el demonio de INFINOTED para crear el certificado (una sola vez)
infinoted --create-key --create-certificate -k /etc/infinote/key.pem -c /etc/infinote/cert.pem
&

Lanzar el demonio de INFINOTED para usar el certificado creado,
(para usarse en la siguiente reinicialización la máquina)
infinoted -k /etc/infinote/key.pem -c /etc/infinote/cert.pem &

Para mayor información ver: <http://gobby.0x539.de/trac/>

Dec 31, 13 18:41

gpg.txt

Page 1/1

```
Instalar el paquete
# apt-get install gnupg
```

```
Generar la llave
$ gpg --gen-key
```

```
Visualizar la llave
$ gpg --list-keys
```

```
Exportar la llave
$ gpg -a --export antonio
```

```
Encriptar archivo
$ gpg -r antonio -ea archivo
```

```
Desencriptar archivo
$ gpg -d archivo.asc > archivo
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Sep 30, 14 7:00

herramientasProgramacion.txt

Page 1/3

Instalar herramientas básicas en Debian

```
# aptitude install g++ valgrind valgrind-mpi cppcheck astyle
```

Compilar fuentes en C++ solicitando que el ejecutable tenga el nombre EJEMP

```
$ g++ -O1 *.cpp -o ejemp
```

para ejecutar el programa ya compilado:

```
$ ./ejemp
```

Compilar usando diversas optimizaciones (-O1, -O2 o -O3) usar:

```
$ g++ -O1 *.cpp
```

para ejecutar el programa ya compilado:

```
$ ./a.out
```

Para compilar y ver todos los avisos usar:

```
$ g++ -pedantic -Wall -Wextra -O *.cpp
```

O

```
$ g++ -Weffc++ *.cpp
```

Para hacer este análisis de rendimiento, hacer:

```
$ g++ -g -pg -O0 *.cpp
```

```
$ ./a.out
```

```
$ gprof -c -z a.out > sal.txt
```

el archivo sal.txt contiene el análisis de rendimiento detallado.

Para hacer depuración del código mediante el depurador gráfico ddd usar:

```
$ g++ -g -O0 *.cpp
```

```
$ ddd ./a.out
```

Puede usarse también los depuradores xxgdb, gdb, kdbg cada uno tiene sus pros y contras, depende del usuario cual es el más adecuado para usar

Para rastreo de problemas con la manipulación de memoria y punteros desbordados:

```
$ g++ -g -O0 *.cpp
```

```
$ valgrind --tool=memcheck --leak-check=yes --show-reachable=yes ./a.out
```

mas opciones en:

```
http://valgrind.org/
```

Existen varios front-end para valgrind, uno de ellos es alleyoop y se usa:

```
$ alleyoop ./a.out -v --arg1=foo
```

otro es kcachevalgrind.

Revisión estática del código en C++ usar

```
$ cppcheck --enable=all *.?pp
```

mostrara los avisos de análisis estático del código indicado

Formateador automático de código fuente para C, C++, C#

```
$ astyle -s3 -p --style=allman --lineend=linux *.?pp
```

para Java, una opción es

```
$ astyle -s2 -p --style=java --lineend=linux *.java
```

más opciones en:

```
http://astyle.sourceforge.net/astyle.html
```

Herramientas para convertir fuentes en HTML

```
$ code2html Fuente Salida.html
```

```
$ java2html Fuentes
```

```
$ c2html Fuente
```

Convertir fuente en PDF

```
$ nl test.cpp | a2ps -1 -l100 -otest.ps ; ps2pdf test.ps
```

```
^
```

```
^
```

```
^
```

```
_____ convierte .PS a .PDF
```

```
_____ Genera del fuente numerado un .PS
```

_____ Numera las líneas del fuente

Si se tiene que ejecutar múltiples programas que son independientes uno de otro se puede usar `parallel` para correr N de ellos al mismo tiempo, por ejemplo si tenemos un archivo: `hacer.bash` con el contenido

```
./a.out 4 5 4 > a1.txt
./a.out 4 5 3 > a2.txt
./a.out 4 5 6 > a3.txt
./a.out 4 5 4 > a4.txt
```

entonces podemos ejecutarlo usando `parallel`, el programa usara el numero máximo de cores disponibles:

```
$ parallel -v < hacer.bash
```

si solo se desea usar una determinada cantidad de cores (por ejemplo 3)

entonces usamos:

```
$ parallel -v -j 2 < hacer.bash
```

Otras herramientas:

DUMA

DUMA helps you detect two common programming bugs:

software that overruns the boundaries of a `malloc()` memory allocation, and software that touches a memory allocation that has been released by `free()`. Unlike other `malloc()` debuggers, DUMA will detect read accesses as well as writes, and it will pinpoint the exact instruction that causes an error. It has been in use at Pixar since 1987, and at many other sites for years.

GCC

option `-Warray-bounds` up from gcc 4.3

options `-fudflap -fudflapth -fudflapir` up from gcc 4.0

See <http://gcc.gnu.org/>

See http://gcc.gnu.org/wiki/Mudflap_Pointer_Debugging

IBM Stack Smashing Protector aka Pro Police

it is a GCC (Gnu Compiler Collection) extension for protecting applications from stack-smashing attacks. Applications written in C will be protected by the method that automatically inserts protection code into an application at compilation time. The protection is realized by buffer overflow detection and the variable reordering feature to avoid the corruption of pointers. The basic idea of buffer overflow detection comes from StackGuard system.

See <http://www.trl.ibm.com/projects/security/ssp/>

Checkergcc

a modified version of the GNU C Compiler that instruments all memory references, is available on Linux systems and where GCC is used. It performs some of the same tasks as Purify, but only on code that it has compiled.

Valgrind

Valgrind is an award-winning suite of tools for debugging and profiling Linux programs. With the tools that come with Valgrind, you can automatically detect many memory management and threading bugs, avoiding hours of frustrating bug-hunting, making your programs more stable. You can also perform detailed profiling, to speed up and reduce memory use of your programs.

The Valgrind distribution currently includes four tools: a memory error detector, a cache (time) profiler, a call-graph profiler, and a heap (space) profiler. It runs on the following platforms: X86/Linux, AMD64/Linux, PPC32/Linux, PPC64/Linux.

See <http://valgrind.org/>

Pageheap.exe

Another alternative to using duma on Windows (XP, 2000, Server 2003) is to use the built in heap debugging provided by Microsoft. It's not as feature rich as duma, but for complex projects where duma is causing issues it will work with less hassle (it sits in the heap manager itself).

You can control it using the global flags (`gflags`) utility that comes with windbg.

You can enable it by saying: `gflags.exe /p /full /enable MyProgram.exe`

And do unaligned by saying: `gflags.exe /p /full /unaligned /enable MyProgram.exe`

`gflags` will set specific registry keys to enable the pageheap debugging on the executable. You can disable it using the `/disable` flag.

Sep 30, 14 7:00

herramientasProgramacion.txt

Page 3/3

See <http://support.microsoft.com/kb/286470>

MPATROL

The mpatrol library is a powerful debugging tool that attempts to diagnose run-time errors that are caused by the wrong use of dynamically allocated memory. It acts as a malloc() debugger for debugging dynamic memory allocations, although it can also trace and profile calls to malloc() and free() too.

See <http://www.cbmamiga.demon.co.uk/mpatrol/>

Purify

does a much more thorough job than DUMA, and does not have the huge memory overhead.

LibSafe

protects Critical Elements of Stack.

See <http://www.research.avayalabs.com/>

DieHard

helps buggy programs run correctly and protects them from a range of security vulnerabilities.

See <http://www.diehard-software.org/>

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:39

iptables.txt

Page 1/2

Generalmente en los equipos personales no se le pone énfasis al control del acceso a la máquina desde el exterior, pero siempre es bueno cerrar el acceso a la máquina mediante algún cortafuegos, una opción es usar IPTABLES, un ejemplo sencillo de esto es usar el script para permitir salir a la red por cualquier puerto, pero no dejar entra a nuestra máquina por ningún puerto.

```
#!/bin/sh
## SCRIPT de IPTABLE

echo -n Aplicando Reglas de Firewall...

## FLUSH de reglas
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establecemos politica por defecto
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT

## Abre puertos al LOCALHOST
iptables -A INPUT -d localhost -j ACCEPT

## Cierra el resto de los puertos
iptables -A INPUT -p tcp --dport 1:1024 -j DROP
iptables -A INPUT -p udp --dport 1:1024 -j DROP

## Abre puertos de Salida y de Paso
iptables -A OUTPUT -d 0.0.0.0/0 -j ACCEPT
iptables -A FORWARD -d 0.0.0.0/0 -j ACCEPT

## Listado de las reglas
iptables -L -n
iptables -L -t nat
```

A partir de esta configuracion básica, se pueden abrir los puertos necesario para acceso a nuestra máquina desde el exterior que se requieran, por ejemplo abrir el puerto SSH (22) usar:

```
#!/bin/sh
## SCRIPT de IPTABLE

echo -n Aplicando Reglas de Firewall...

## FLUSH de reglas
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## Establecemos politica por defecto
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT

## Abre puertos al LOCALHOST
iptables -A INPUT -d localhost -j ACCEPT

## Abre puertos especificos
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

## Cierra el resto de los puertos
iptables -A INPUT -p tcp --dport 1:1024 -j DROP
iptables -A INPUT -p udp --dport 1:1024 -j DROP

## Abre puertos de Salida y de Paso
iptables -A OUTPUT -d 0.0.0.0/0 -j ACCEPT
iptables -A FORWARD -d 0.0.0.0/0 -j ACCEPT
```


Feb 27, 13 20:39

iptables.txt

Page 2/2

```
## Listado de las reglas
iptables -L -n
iptables -L -t nat
```

Se puede abrir el acceso a determinadas IPs o MACs; redireccionado de puertos y otras opciones avanzadas, sólo es cuestión de agregar esas especificaciones al script básico.

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:39

iso.txt

Page 1/1

```
Instalar
# apt-get install genisoimage

Diversas formas de generar un ISO

Vanilla ISO9660 filesystem
$ genisoimage -o cd.iso cd_dir

Rock Ridge filesystem
$ genisoimage -o cd.iso -R cd_dir

HFS hybrid CD with the Joliet and Rock Ridge
$ genisoimage -o cd.iso -R -J -hfs cd_dir

HFS hybrid CD contains Netatalk Apple/Unix files
$ genisoimage -o cd.iso --netatalk cd_dir

Apple Extensions to ISO9660
$ genisoimage -o cd.iso -apple -magic magic -probe rs cd_dir

Montar y/o revisar ISO
# mkdir /mnt/iso
# mount -t iso9660 -o loop,ro /ruta/imagen.iso /mnt/iso

Desmontar la imagen ISO
# umount -lf /mnt/iso

Para generar un ISO a partir de un CD o DVD
# dd if=/dev/hdc of=recopilacion.iso
o usar
# dd if=/dev/dvd of=dvd.iso # para dvd
# dd if=/dev/cdrom of=cd.iso # para cdrom
# dd if=/dev/scd0 of=cd.iso # si el cdrom es scsi

Para quemar CDs o DVDs se pueden usar los siguientes paquetes
# aptitude install brasero k3b

Más opciones en:
http://www.yolinux.com/TUTORIALS/LinuxTutorialCDBurn.html
```

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:40

kernel.txt

Page 1/1

Algunas veces es necesario instalar una version más actual del kernel de Linux que por omisión no esta disponible en Debian, para ello existe el proyecto liquorix:

```
http://liquorix.net/
```

De este proyecto se puede instalar una versión nueva del kernel en Debian de manera muy sencilla, lo primero es agregar a sources.list la línea:

```
deb http://liquorix.net/debian sid main
```

Ahora actualizamos la lista de paquetes, mediante:

```
# aptitude update
```

Despues instalamos la keyring, mediante

```
# apt-get install '^liquorix-([^-]+-)?keyring.?'
```

Asi, podemos ver los paquetes disponibles usando:

```
# aptitude search liquorix
```

Finanlemente para instalar el nuevo kernel, generalmente usamos:

```
# aptitude install linux-image-2.6-liquorix-686
```

Con esto tendremos instalado una imagen del kernel muy actualizada con respecto a la que se tiene acceso en las diferentes ramas de Debian.

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Jun 29, 12 17:03

makefile_Paralelo.txt

Page 1/3

```

# makefile for parallel programs i.e. MPICH, OpenMPI, etc
#   Author: Antonio Carrillo Ledesma 29/12/2010
#   antonio@mmc.geofisica.unam.mx
#   http://www.mmc.geofisica.unam.mx/acl/
#
# usage:
#   First automatic dependency calculation
#   $ make deps
#
#   Compile and link
#   $ make
#
#   Run
#   $ make run
#
#   Clean
#   $ make clean
#
#   Others
#   target: gz - Store in tar.gz
#   target: bz2 - Store in tar.bz2
#   target: tar - Store in tar
#   target: zip - Store in zip
#   target: val - Debugger with valgrind
#   target: gpro - Analysis of speedup
#   target: svnup - svn update
#   target: svnst - svn status -verbose
#   target: svnci - svn commit
#   target: help - Display callable targets.

# Defining the sources
SOURCES = $(wildcard *.cpp)

# Parameters of executable
PARAMETERS =

# Defining the numbers of CPUs
CPUS = -np 4

# Defining the neme of th executable
EXECUTABLE = test

# Defining the debug Flags
DEBUG = -g -pg -O0      # Debugger
#DEBUG += -Wall         # Warning
#DEBUG += -O2           # Optimization

# Defining the compiler Flags
CFLAGS = $(DEBUG)

# Defining the linker Flags
LDFLAGS = -lm $(DEBUG)

# Defining the compiler:
CC = mpiCC.mpich

#####
# Defining the object files:
OBJECTS = $(SOURCES:.cpp=.o)

# Load the automatic dependency
DEPS = $(OBJECTS:.o=.d)

# target: all - The default rule - compiling our main program:
all: $(SOURCES) $(EXECUTABLE)
    echo all: make complete

$(EXECUTABLE): $(OBJECTS)

```

Jun 29, 12 17:03

makefile_Paralelo.txt

Page 2/3

```

$(CC) $(LDFLAGS) $(OBJECTS) -o $@

# Tell make how to build .o files from .cpp files:
$(OBJECTS):
    $(CC) -MD -c $(CFLAGS) $(@:.o=.cpp) -o $@

#Now make sure that make rebuilds files if included headers change:
-include $(DEPS)

# target: deps - Automatic dependency calculation
deps: $(SOURCES)
    $(CC) -MD -E $(SOURCES) > /dev/null

# The .PHONY rule keeps make from doing something with a file named clean
.PHONY: clean

# target: clean - Removing the executable and the object files
clean:
    rm $(EXECUTABLE) $(OBJECTS) $(DEPS) gmon.out
    echo clean: make complete

# target: run - Run the application
run:
    lamboot -v
    mpirun.mpich $(CPUS) ./$(EXECUTABLE) $(PARAMETERS)
    lamhalt -v

# target: gz - Store in tar.gz
gz:
    tar -zcvpf $(EXECUTABLE)-`date +%F`.tar.gz makefile *.?pp

# target: bz2 - Store in tar.bz2
bz2:
    tar -jcvpf $(EXECUTABLE)-`date +%F`.tar.bz2 makefile *.?pp

# target: tar - Store in tar
tar:
    tar -cvpf $(EXECUTABLE)-`date +%F`.tar makefile *.?pp

# target: zip - Store in zip
zip:
    zip $(EXECUTABLE)-`date +%F`.zip makefile *.?pp

# target: val - Debugger with valgrind, required DEBUG = -g
val:
    lamboot -v
    valgrind --tool=memcheck --leak-check=yes --show-reachable=yes mpirun.mpich $(CPUS) ./$(EXECUTABLE)
    lamhalt -v

# target: gpro - Analysis of speedup, required DEBUG = -g -pg
gpro:
    lamboot -v
    mpirun.mpich $(CPUS) ./$(EXECUTABLE) $(PARAMETERS)
    gprof -c -z $(EXECUTABLE)
    lamhalt -v

# target: svnup - svn update
svnup:
    svn update

```

Jun 29, 12 17:03

makefile_Paralelo.txt

Page 3/3

```
# target: svnst - svn status
svnst:
    svn status --verbose
```

```
# target: svnci - svn commit
svnci:
    svn ci -m "Update"
```

```
# target: help - Display callable targets.
help:
    egrep "^# target:" [Mm]akefile
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Jun 29, 12 17:03

makefile_Secuencial.txt

Page 1/3

```

# makefile for secuential programs i.e. g++, gcc, etc
#   Author: Antonio Carrillo Ledesma 29/12/2010
#   antonio@mmc.geofisica.unam.mx
#   http://www.mmc.geofisica.unam.mx/acl/
#
# usage:
#   First automatic dependency calculation
#     $ make deps
#
#   Compile and link
#     $ make
#
#   Run
#     $ make run
#
#   Clean
#     $ make clean
#
#   Others
#     target: gz - Store in tar.gz
#     target: bz2 - Store in tar.bz2
#     target: tar - Store in tar
#     target: zip - Store in zip
#     target: ddd - Debugger with ddd
#     target: val - Debugger with valgrind
#     target: gpro - Analysis of speedup
#     target: svnup - svn update
#     target: svnst - svn status -verbose
#     target: svnci - svn commit
#     target: help - Display callable targets.

# Defining the sources
SOURCES = $(wildcard *.cpp)

# Parameters of executable
PARAMETERS =

# Defining the neme of th executable
EXECUTABLE = test

# Defining the debug Flags
DEBUG = -g -pg -O0      # Debugger
#DEBUG += -Wall         # Warning
#DEBUG += -O2           # Optimization

# Defining the compiler Flags
CFLAGS = $(DEBUG)

# Defining the linker Flags
LDFLAGS = -lm $(DEBUG)

# Defining the compiler:
CC = g++

#####
# Defining the object files:
OBJECTS = $(SOURCES:.cpp=.o)

# Load the automatic dependency
DEPS = $(OBJECTS:.o=.d)

# target: all - The default rule - compiling our main program:
all: $(SOURCES) $(EXECUTABLE)
    echo all: make complete

$(EXECUTABLE): $(OBJECTS)
    $(CC) $(LDFLAGS) $(OBJECTS) -o $@

```

Jun 29, 12 17:03

makefile_Secuencial.txt

Page 2/3

```

# Tell make how to build .o files from .cpp files:
$(OBJECTS):
    $(CC) -MD -c $(CFLAGS) $(@:.o=.cpp) -o $@

#Now make sure that make rebuilds files if included headers change:
-include $(DEPS)

# target: deps - Automatic dependency calculation
deps: $(SOURCES)
    $(CC) -MD -E $(SOURCES) > /dev/null

# The .PHONY rule keeps make from doing something with a file named clean
.PHONY: clean

# target: clean - Removing the executable and the object files
clean:
    rm $(EXECUTABLE) $(OBJECTS) $(DEPS) gmon.out
    echo clean: make complete

# target: run - Run the application
run:
    ./$(EXECUTABLE) $(PARAMETERS)

# target: gz - Store in tar.gz
gz:
    tar -zcvpf $(EXECUTABLE)-`date +%F`.tar.gz makefile *.?pp

# target: bz2 - Store in tar.bz2
bz2:
    tar -jcvpf $(EXECUTABLE)-`date +%F`.tar.bz2 makefile *.?pp

# target: tar - Store in tar
tar:
    tar -cvpf $(EXECUTABLE)-`date +%F`.tar makefile *.?pp

# target: zip - Store in zip
zip:
    zip $(EXECUTABLE)-`date +%F`.zip makefile *.?pp

# target: ddd - Debugger with ddd, required DEBUG = -g
ddd:
    ddd $(EXECUTABLE)

# target: val - Debugger with valgrind, required DEBUG = -g
val:
    valgrind --tool=memcheck --leak-check=yes --show-reachable=yes ./$(EXECUTABLE) $(PARAMETERS)

# target: gpro - Analysis of speedup, required DEBUG = -g -pg
gpro:
    ./$(EXECUTABLE) $(PARAMETERS)
    gprof -c -z $(EXECUTABLE)

# target: svnup - svn update
svnup:
    svn update

# target: svnst - svn status
svnst:
    svn status --verbose

```


Jun 29, 12 17:03

makefile_Secuencial.txt

Page 3/3

```
# target: svnci - svn commit
svnci:
    svn ci -m "Update"
```

```
# target: help - Display callable targets.
help:
    egrep "^# target:" [Mm]akefile
```

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:44

make.txt

Page 1/4

Cundo iniciamos un proyecto de programación es común que nos baste con un simple g++ *.cpp para compilar nuestro código de forma rápida, pero conforme el número de fuentes crece es necesario usar algo que nos permita recompilar sólo el código que se ve afectado con los cambios a los fuentes realizados y no todos los fuentes, esto se logra usando make y el archivo makefile.

Por lo general al ejecutar make solo hace uso de un core en las compilaciones, si se cuenta con un equipo con varios cores se pueden usar todos ellos para hacer más rápida la compilación, usar

```
$ make -j n
```

donde el valor óptimo de n se puede encontrar variando N; para medir el tiempo de compilación usar el comando:

```
$ time make -j 2
```

Este nos permite tener las interdependencias de nuestro código especificadas para que sólo recompile aquello que sea requerido, su estructura básica es:

```
=====
# Inicio del makefile

# Indicar el compilador
CC=g++
# Indicar las opciones del compilador
CFLAGS=-c -Wall

all: test

# indicar que se liga y el nombre del ejecutable
test: ejem.o A.o B.o
    $(CC) ejem.o A.o B.o -o test

# opciones de compilación por cada fuente
ejem.o: ejem.cpp
    $(CC) $(CFLAGS) ejem.cpp

A.o: A.cpp
    $(CC) $(CFLAGS) A.cpp

B.o: B.cpp
    $(CC) $(CFLAGS) B.cpp

# Permite borrar los archivos generados
clean:
    rm -rf *o test

# fin de makefile
=====
```

Uso básico de Make:

Para compilar
\$ make

Para borrar los archivos generados por la compilación
\$ make clean

Nota: Para que funcione el makefile después de una regla debe de usarse tabulador para indicar la acción, no funciona si sólo se usan espacios.

Pero que pasa si nuestro proyecto crece y crece el número de archivos fuente, entonces usar este esquema puede ser engorroso de mantener, una opción es usar un makefile genérico, y que sea este mismo makefile el encargado de llevar la cuenta de los fuentes y las dependencias entre los mismos.

Para ello podemos usar:

```
=====
# Inicio de makefile

# Defining the sources
```

Feb 27, 13 20:44

make.txt

Page 2/4

```

SOURCES = $(wildcard *.cpp)

# Parameters of executable
PARAMETERS = file test

# Defining the name of the executable
EXECUTABLE = ressim

# Defining the debug flags
DEBUG = -g -pg -Wall      # Debugger

# Defining the compiler flags
CFLAGS = -MD -c $(DEBUG)

# Defining the linker flags
LDFLAGS = -lm $(DEBUG)

# Defining the compiler:
CC = g++

# Defining the object files:
OBJECTS = $(SOURCES:.cpp=.o)

# Load the automatic dependency
DEPS = $(patsubst %.o,%.d,$(OBJECTS))

# target: all - The default rule - compiling our main program:
all: $(SOURCES) $(EXECUTABLE)
    echo all: make complete

$(EXECUTABLE): $(OBJECTS)
    $(CC) $(LDFLAGS) $(OBJECTS) -o $@

# Tell make how to build .o files from .cpp files:
$(OBJECTS):
    $(CC) $(CFLAGS) $(@:.o=.cpp) -o $@

# Now make sure that make rebuilds files if included headers change:
-include $(DEPS)

# target: deps - Automatic dependency calculation
deps: $(SOURCES)
    $(CC) -MD -E $(SOURCES) > /dev/null

# The .PHONY rule keeps make from doing something with a file named clean
.PHONY: clean

# target: clean - Removing the executable and the object files
clean:
    rm $(EXECUTABLE) $(OBJECTS) $(DEPS) gmon.out
    echo clean: make complete

# target: run - Run the application
run:
    ./$(EXECUTABLE) $(PARAMETERS)

# Fin de makefile
=====

Uso basico de Make:

Para generar dependencias de los fuentes
$ make deps

Para compilar
$ make

Para ejecutar
$ make run

Para borrar los archivos generados por la compilación

```

Feb 27, 13 20:44

make.txt

Page 3/4

```
$ make clean
```

Este makefile soporta una amplia configuracion y es reusable en proyectos de C y C++, ademas se puede extender para soportar varias actividades rutinarias en el manejo de nuestro proyecto, como son compactar en distintos formatos (como tar, tar.gz, tar.bz2, zip), usar subversion o depurar con ddd y valgrind, etc. Angunas opciones son:

```
# target: gz - Store in tar.gz
```

```
gz:
    tar -zcvpf $(EXECUTABLE)-`date +%F`.tar.gz makefile *.?pp
```

```
# target: bz2 - Store in tar.bz2
```

```
bz2:
    tar -jcvpf $(EXECUTABLE)-`date +%F`.tar.bz2 makefile *.?pp
```

```
# target: tar - Store in tar
```

```
tar:
    tar -cvpf $(EXECUTABLE)-`date +%F`.tar makefile *.?pp
```

```
# target: zip - Store in zip
```

```
zip:
    zip $(EXECUTABLE)-`date +%F`.zip makefile *.?pp
```

```
# target: ddd - Debugger with ddd, required DEBUG = -g
```

```
ddd:
    ddd $(EXECUTABLE)
```

```
# target: val - Debugger with valgrind, required DEBUG = -g
```

```
val:
    valgrind --tool=memcheck --leak-check=yes --show-reachable=yes \
        ./${EXECUTABLE} $(PARAMETERS)
```

```
# target: gpro - Analysis of speedup, required DEBUG = -g -pg
```

```
gpro:
    ./${EXECUTABLE} $(PARAMETERS)
    gprof -c -z $(EXECUTABLE)
```

```
# target: svnup - svn update
```

```
svnup:
    svn update
```

```
# target: svnst - svn status
```

```
svnst:
    svn status --verbose
```

```
# target: svnci - svn commit
```

```
svnci:
    svn ci -m "Update"
```

De esta forma es posible tener un makefile flexible y poderoso para usarse en múltiples proyectos de desarrollo de software con un mínimo de cambios

Nota: Una versión de makefile para programación secuencial y otro paralela se pone en este directorio, pero recuerde que para el funcionamiento de make, es necesario respetar los caracteres tabulador en las acciones del makefile.

Feb 27, 13 20:44

make.txt

Page 4/4

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Mar 04, 13 19:31

maquinasVirtuales.txt

Page 1/5

Revisión del tipo de virtualización soportado por la máquina

Primeramente hay que revisar si hay soporte en Hardware para la virtualización, usando para ello:

```
$ egrep "vmx|svm" /proc/cpuinfo
```

si se soporta la virtualización por Hardware aparecerá la bandera

```
Procesadores INTEL: vmx
Procesadores AMD:  svm
```

Instalar y Usar máquinas virtuales

Por omisión los equipos de tecnología de bajo desempeño no soportan la virtualización a nivel de Hardware, por ello es común el uso de QEMU. Si la computadora soporta virtualización a nivel de Hardware es posible usar KVM, estos tienen la misma sintaxis de uso, y sólo es necesario reemplazar qemu por kvm y siempre se usara qemu-img para ambos paquetes.

```
Instalación de KVM (recomendado para virtualización por Hardware)
# aptitude install kvm
```

```
Instalación de QEMU
# aptitude install qemu
```

Nota: El desempeño de QEMU versus KVM es de varios ordenes de magnitud menor, pero una imagen creada con cualquiera de ellos es usable con los otros virtualizadores (en el caso de Windows sólo hay que usar un Hardware parecido al de qemu, en caso contrario marca que es necesario registrar el sistema operativo, para ello se usa la bandera -cpu). Por ejemplo:

```
Usar máquina virtual de Windows de QEMU en KVM:
$ kvm -localtime -m 400 -boot c -hda Windows.img -cpu qemu32
```

Suponiendo que se instalo QEMU, entonces podemos usar la máquina virtual con un archivo ISO, en este ejemplo no supondremos que tenemos disco duro, sólo cdrom

```
$ qemu -cdrom knoppix.iso -m 700
```

Un ejemplo completo de instalación y uso de una máquina virtual para Windows XP, necesitamos:

```
Crear la disco virtual, por ejemplo de 10 GB
$ qemu-img create -f qcow2 WindowsXP.img 10G
```

Hacer la instalación básica de Windows XP a partir, por ejemplo del ISO

```
$ qemu -no-reboot -cdrom es_winxp_pro_with_sp2.iso -boot d -hda WindowsXP.img -m 400 -localtime
```

Concluir la instalación de Windows XP

```
$ qemu -no-reboot -boot c -hda WindowsXP.img -cdrom es_winxp_pro_with_sp2.iso -m 400 -localtime
```

Después de la instalación, es conveniente compactar y desfragmentar la imagen usando

```
$ qemu-img convert -c WindowsXP.img -O qcow2 Windows.img
```

Uso de máquina virtual de Windows XP

```
$ qemu -boot c -hda Windows.img -m 400 -localtime
```

Un ejemplo completo de instalación y uso de una máquina virtual para Windows 7, necesitamos:

```
Crear la disco virtual, por ejemplo de 15 GB
$ qemu-img create -f qcow2 Windows7.img 15G
```

Hacer la instalación básica de Windows 7 a partir, por ejemplo del DVD

```
$ kvm -no-reboot -cdrom /dev/cdrom -boot d -hda Windows7.img -m 500 -localtime
```

Concluir la instalación de Windows 7

```
$ kvm -no-reboot -boot c -hda Windows7.img -cdrom /dev/cdrom -m 500 -localtime
```

Después de la instalación, es conveniente compactar y desfragmentar la imagen usando

```
$ qemu-img convert -c Windows7.img -O qcow2 Windows.img
```

Uso de máquina virtual de Windows 7

Mar 04, 13 19:31

maquinas Virtuales.txt

Page 2/5

```
$ kvm -boot c -hda Windows.img -m 500 -localtime
```

Una vez que se cuenta con una imagen de Windows, podemos instalar por ejemplo Office

Instalar Windows Office 2003, aquí suponemos que tenemos un ISO de Office

```
$ qemu -localtime -m 300 -boot c -hda Windows.img -cdrom Office-2003.iso
```

Si se tiene el cdrom, entonces podemos usar

```
$ qemu -localtime -m 300 -boot c -hda Windows.img -cdrom /dev/cdrom/
```

Otro ejemplo, lo tenemos en la instalación de Debian estable a partir del archivo ISO bajado de la red

Generar un disco virtual, por ejemplo de 10 GB

```
$ qemu-img create -f qcow2 debianStable.img 10G
```

Instalar la imagen de Debian estable en un disco virtual

```
$ qemu -no-reboot -boot d -cdrom debian-503-i386-netinst.iso -hda debianStable.img -m 300
```

Usar Debian estable

```
$ qemu -hda debianStable.img -m 400
```

También podemos manipular la máquina virtual al usar la combinación de teclas:

```
[Ctrl] + [Alt] + [2]
```

ya en ella, por ejemplo podemos detener y grabar el estado de la máquina virtual:

```
savevm test.vm
```

```
quit
```

Para que en otro momento, podamos restaurar la máquina virtual tal como estaba cuando esta se detuvo:

```
$ qemu -boot c -hda .img -m 400 -localtime -loadvm test.vm
```

Optimización de imágenes

Las imágenes de disco de QEMU después de generarlas (al instalar algún sistema operativo) son archivos dispersos, para optimizar su rendimiento es recomendable convertir la imagen dispersa en una que no tenga esta propiedad, usar:

```
$ qemu-img convert disk-sparse.img -O qcow2 disk.img
```

o puede ser compactada, usar:

```
$ qemu-img convert -c disk-sparse.img -O qcow2 disk.img
```

la cual decrecerá el tamaño, para descompactar una imagen e incrementar la velocidad de uso, usar:

```
$ qemu-img convert disk-compact.img -O qcow2 disk.img
```

Trabajar con una imagen virtual sin que se altere esta

Es muy deseable al trabajar con una máquina virtual, el dejar la información de la máquina virtual base intacta y guardar los cambios que se requieran en otro archivo, una forma es hacer una copia y trabajar con la copia de esta o crear un archivo que almacene por separado los cambios a la imagen, para esto último usar:

```
$ qemu-img create -b debianStable.img -f qcow2 debian.img
```

y trabajar con la imagen que resultante (para este ejemplo debian.img)

Comunicación entre la máquina virtual y el sistema anfitrión

Que hacer para tener comunicación entre la máquina virtual y el sistema anfitrión, hay varias maneras de hacer esto:

Mar 04, 13 19:31

maquinas Virtuales.txt

Page 3/5

- 1) Lo más sencillo es que la máquina virtual se conecte a un servidor en red del tipo samba, este puede ser una máquina Windows que comparta una impresora y/o disco o la máquina anfitrión tenga instalado SAMBA y comparta uno o más servicios como son discos o impresoras. Para ver la instalación y configuración de un servidor de SAMBA cheque el texto samba.txt
- 2) Conectarse a un servidor de SSH mediante los programas SSH Server, así la máquina virtual puede acceder mediante SFTP a los archivos en el servidor. Es posible instalar el servidor de SSH en la máquina anfitrión y así poder prestar el servicio de SSH y FTP a la máquina virtual. Para ver la instalación y configuración de un servidor de SSH cheque el texto ssh.txt
- 3) Leer un dispositivo USB montado en el sistema anfitrión desde la máquina virtual, para ello el dispositivo USB deberá estar conectado en la máquina anfitrión y deberá ser accesado directamente en la máquina virtual. QEMU necesita parámetros adicionales, el parámetro `-usb` activa el soporte en la máquina virtual de dispositivos USB. La emulación de Intel SB82371 UHCI-Controller tiene 8-puertos en el USB hub. Si se busca tener acceso a uno de los dispositivos físicos, se requiere encontrar los parámetros Vendor-ID and Product-ID. Esta información se obtiene examinando la salida del comando:


```
# /sbin/lsub
```

```
$ cat /proc/bus/usb/devices
```

Entonces es posible decirle a QEMU los datos de VendorID y ProductID a través de la línea de comandos

```
$ qemu -usb -usbdevice host:<VendorID>:<ProductID> <otros parámetros>
```

o iniciar QEMU con soporte para dispositivos USB activados mediante

```
$ qemu -usb <otros parámetros>
```

después de iniciar la máquina virtual, cambiar al sistema de monitoreo de la máquina virtual presionando `Ctrl+Alt+2` e introducir el siguiente comando

```
usb_add host:<VendorID>:<ProductID>
```

cuando se retorne al ambiente gráfico al teclear `[Ctrl]+[Alt]+[1]` se vera el mensaje de reconocimiento del dispositivo USB.

Por ejemplo si se tiene una impresora HP Scanjet 3300C conectada en el puerto USB de la computadora, la salida del comando `lsub` es:

```
# lsub
Bus 003 Device 002: ID 03f0:0205 ScanJet 3300C
```

así, el comando en QEMU para dejar accesible el dispositivo es:

```
$ qemu -usb -usbdevice host:03f0:0205 <otros parametros>
```
- 4) Usar la impresora conectada en el puerto paralelo, para ello al invocar la ejecución de la máquina virtual usar:


```
$ qemu -parallel /dev/parport0 <otros parámetros>
```
- 5) Montar el contenido de un disco virtual y poder intercambiar información entre la máquina virtual y la huésped, primero convertir el disco a formato accesible a Linux


```
$ qemu-img convert disco.img -O raw tmp.img
```

montar la imagen en Linux como root

```
# mkdir disk
# mount -o loop,offset=32256 tmp.img disk
```

trabajar con la imagen montada y al terminar desmontar esta

```
# umount ./disk
```

y puede ser regresada al formato original mediante

```
$ qemu-img convert -c tmp.img -O qcow2 disco.img
```

Más información en la página:

<http://www.qemu.org/user-doc.html>

Algunos problemas comunes con la red

Por lo general las máquinas virtuales detectan correctamente la red, pero en el caso de Windows esto no siempre pasa, por ello es común emular una tarjeta de red lo más generica posible, esta puede ser RTl8139, para ello es necesario que al lanzar la máquina virtual se indique `-net nic,model=rtl8139 -net user`, por ejemplo:

Mar 04, 13 19:31

maquinasVirtuales.txt

Page 4/5

```
$ qemu -boot c -hda WindowsXP.img -m 400 -localtime -net nic,model=rtl8139 -net user
algunas de las otras opciones para la red son: NE2000 PCI, RTL8139, PCNET y NE2000 ISA.
```

Algunos problemas con KVM

Si se detectan las banderas para virtualización por Hardware y al tratarlo de usar marca:

```
> open /dev/kvm: Permission denied
> Could not initialize KVM, will disable KVM support
Solo hay que agregar, el login del usuario al grupo kvm en el archivo /etc/group

> open /dev/kvm: No such file or directory
> Could not initialize KVM, will disable KVM support
Solo hay que activar en el BIOS la virtualización por Hardware
```

Aumento de desempeño

La virtualización normalmente es rápida, pero en algunas circunstancias se hace lenta generalmente esto es ajeno a QEMU o KVM y generalmente es por la constante grabación de datos al disco duro por parte de la máquina virtual. Por optimizar el desempeño de la máquina virtual es posible pedirle a QEMU o KVM que trate de usar un cache y baje lo menos posible a disco la información, esto aumentara notablemente el desempeño de la máquina virtual.

Para aumentar el desempeño, en lugar de usar:

```
$ qemu -boot c -hda WindowsXP.img -m 400 -localtime -net nic,model=rtl8139 -net user
```

usar en QEMU:

```
$ qemu -drive file=WindowsXP.img,cache=writeback,media=disk -m 400 -localtime -net nic,model=rtl8139 -net user
```

usar en KVM:

```
$ kvm -drive file=WindowsXP.img,cache=writeback,media=disk -m 400 -localtime -net nic,model=rtl8139 -net user
```

En el caso de usar un iso, usar

```
$ kvm -m 512 -drive file=fedora.iso,cache=writeback,media=cdrom
```

Instalando UBUNTU 11.10 mediante KVM

En este caso generaremos un disco de 10 GB, se carga Ubuntu y dentro de el procedemos a instalar Ubuntu en el disco recién generado, una vez terminada la instalación, podemos hacer uso de nuestra nueva imagen.

```
$ qemu-img create -f qcow2 disco.img 10G
$ kvm -no-reboot -boot d -drive file=ubuntu-11.10-desktop-i386.iso,cache=writeback,media=cdrom -drive file=disco.img,cache=writeback,media=disk -m 500 &
$ kvm -drive file=disco.img,cache=writeback,media=disk -m 500
```

aquí, se usa el cache para acelerar el desempeño de KVM.

Mejorando el desempeño del vídeo de la máquina virtual

Por omisión se tiene un tarjeta gráfica de pobre desempeño en la máquina virtual, si se necesita mayor resolución en la salida gráfica, una opción es usar la opción -VGA, donde dos de sus posibilidades es STD o VMWARE usándose como:

```
$ kvm -vga std -m 512 -drive file=fedora.iso,cache=writeback,media=cdrom
```

o

```
$ kvm -vga vmware -m 512 -drive file=fedora.iso,cache=writeback,media=cdrom
```

Mar 04, 13 19:31

maquinasVirtuales.txt

Page 5/5

Usando QEMU o KVM con USB live

Para el caso de tener un USB live y se quiera correr su contenido desde una máquina virtual con QEMU o KVM solo es necesario montar el USB, conocer el dispositivo mediante

```
$ df
```

y usar ese dispositivo en

```
$ kvm -usb /dev/sddx
```

Direcciones de red usadas en QEMU o KVM

Gateway/DHCP/TFTP server: 10.0.2.2

DNS server: 10.0.2.3

Samba server: 10.0.2.4

Netmask: 255.255.255.0

Guest IP: any address above 10.0.2.15

Uso de tarjeta de sonido dentro de QEMU o KVM

Por omisión el uso de la tarjeta de audio no esta habilitada, para habilitarla usar en la linea de comandos: -soundhw sb16,es1370,adlib

por ejemplo

```
$ qemu -boot c -hda Windows.img -m 400 -localtime -soundhw sb16,es1370,adlib
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:44

ntfs.txt

Page 1/1

```
Para tener acceso a particiones NTFS en Linux, instalar ntfs-3g
# aptitude install ntfs-3g

Para montar una partición NTFS /dev/sda1 en un directorio por
ejemplo /mnt/windows usar en linea de comandos
# ntfs-3g /dev/sda1 /mnt/windows
para desmontar el disco usar
# umount /mnt/windows

Para dejar permanente el acceso a la partición, editar como root
# nano /etc/fstab
agregando la linea
/dev/sda1 /mnt/windows ntfs-3g defaults 0 0
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 27, 13 20:49

openvpn.txt

Page 1/2

En estas líneas se explica brevemente como configurar VPN usando OpenVPN, tanto de la parte del servidor como del cliente. Se entenderá como servidor un equipo que es accesible por el cliente en internet y que generalmente cuenta con una IP homologada; el cliente es un equipo conectado a cualquier proveedor de acceso a internet que generalmente proporciona el acceso vía su intranet y el IP es generado dinámicamente por DHCP.

La idea detrás de una red virtual privada es permitir la interconexión en ambos sentidos, para poder acceder a datos y programas que se encuentran en el cliente con sólo prenderlo y ponerlo en red, VPN permite que desde el servidor o cualquier equipo que pueda acceder al servidor, tener acceso a los datos y programas del cliente como si el cliente se encontrara en la misma red que el servidor.

En el servidor, como root

1. Instalar el paquete openvpn mediante
aptitude install openvpn
2. En el servidor generar la clave estática
cd /etc/openvpn
openvpn --genkey --secret static.key
3. Copiar la clave estática (static.key) al cliente en el directorio /etc/openvpn usando scp o sftp.
4. En el servidor, crear el archivo /etc/openvpn/tun0.conf y agregar:
dev tun0
ifconfig 10.9.8.1 10.9.8.2
secret /etc/openvpn/static.key
5. Iniciar OpenVPN manualmente mediante el comando:
openvpn --config /etc/openvpn/tun0.conf --verb 6
la opción verb 6, proporciona información abundante de las conexiones y tráfico generado

Nota: 10.9.8.x es la VPN subnetwork, 10.9.8.1 es el IP del servidor, 10.9.8.2 es el IP del cliente.

En el cliente, como root

1. Instalar el paquete openvpn mediante
aptitude install openvpn
2. Copiar la clave estática generada por el servidor (static.key) en el directorio /etc/openvpn
3. Crear el archivo /etc/openvpn/tun0.conf y agregar:
remote your-server.org
dev tun0
ifconfig 10.9.8.2 10.9.8.1
secret /etc/openvpn/static.key
4. Iniciar OpenVPN manualmente mediante el comando:
openvpn --config /etc/openvpn/tun0.conf --verb 6

Por omisión, el servicio de openvpn se inicializa al arrancar la máquina, por ello no se requiere la inicialización manual de OpenVPN, si se quiere detener o inicializar se puede usar:

```
# /etc/init.d/openvpn stop
o
# /etc/init.d/openvpn start
```

Ahora ya está activo el túnel, se puede ver la tunelización mediante

```
# ifconfig -a
o
# route -n
```

Para revisar la tunelización como usuario normal, en el cliente

```
$ ping 10.9.8.1
en el servidor
$ ping 10.9.8.2
```

Nota: En este caso no se usa el IP del cliente o servidor local, se usa el creado para la tunelización

Para conectarse usando ssh en el cliente como usuario normal

```
$ ssh -l usr 10.8.9.1
en el servidor
```

Feb 27, 13 20:49

openvpn.txt

Page 2/2

```
$ ssh -l usr 10.8.9.2
```

```
-----  
---  
Algo de linux: http://www.mmc.geofisica.unam.mx/acl/linux/
```

```
Antonio Carrillo Ledesma  
http://www.mmc.geofisica.unam.mx/acl/
```

Feb 27, 13 20:50

papeleraParaLineaDeComandos.txt

Page 1/1

¿Qué ocurre cuando hacemos un `rm` en la consola para borrar un archivo o un directorio y de repente nos damos cuenta de que nos hemos equivocado? Tendremos que buscar un software de recuperación. ¿No habría sido mejor que se moviera a una papelera, como ocurre con el equivalente gráfico? Buenas noticias, se puede hacer, y muy fácilmente.

En Debian tendrías que ejecutar como usuario `root`:

```
# aptitude install libtrash
```

y añadir en tu fichero de usuario `.bashrc` una nueva variable de entorno `LD_PRELOAD` (que carga en memoria la librería dinámica que le indiquemos) con la ruta a la librería

```
$ echo "export LD_PRELOAD=/usr/lib/libtrash/libtrash.so.2.4" >> ~/.bashrc
```

Ahora puedes cerrar y abrir la consola de nuevo para grabar los cambios o ejecutar

```
$ source ~/.bashrc
```

Ahora vamos a probar nuestros nuevos superpoderes. Creamos un archivo cualquiera

```
$ echo "hola" > hola.txt
```

y ahora lo borramos

```
$ rm hola.txt
```

el archivo habrá ido a parar al directorio `Trash` en nuestro directorio `HOME`, que es el que nos hace de papelera.

Pero esto puede mejorar aún más: `libtrash` tiene un archivo de configuración que nos permite indica el directorio a utilizar como papelera, entre otras cosas. Si le indicamos `.Trash`, que es el archivo que usa `Gnome` como papelera, ¡cuando borremos un archivo este ira a parar a la papelera de `Gnome` y podremos borrarlo desde allí de forma gráfica!

```
$ echo "TRASH_CAN = .Trash" > ~/.libtrash
```

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Oct 27, 15 9:01

paquetes.txt

Page 1/6

Por omisión en Debian sólo se instalan paquetes libres, pero hay otro tipo de paquetes útiles que no son libres o que tienen licencia distinta a la usada por Linux Debian, para poder tener acceso a ello hay que modificar el archivo /etc/apt/sources.list, mediante:

```
# nano /etc/apt/sources.list
```

en México hay dos replicas de paquetes, ambas en la UNAM, una de ellas en el Instituto de Geofísica, para acceder a ella agregar:

```
deb http://www.mmc.geofisica.unam.mx/debian/ testing main contrib non-free
deb-src http://www.mmc.geofisica.unam.mx/debian/ testing main contrib non-free
```

y otra en la Facultad de Ciencias, para acceder a ella agregar:

```
deb http://nisamox.fciencias.unam.mx/debian/ testing main contrib non-free
deb-src http://nisamox.fciencias.unam.mx/debian/ testing main contrib non-free
```

Esto permite tener acceso a paquetes libres (main) y no libres o con licencias distintas a GLP de Linux Debian (contrib y non-free). Una vez agregadas las replicas hay que actualizar las definiciones de paquetes en Debian, usando:

```
# aptitude update
# aptitude safe-upgrade
# aptitude clean
```

Ahora el sistema esta listo para poder instalar los paquetes que el usuario de la máquina requiere, estos son sólo algunos de los mas de 39,000 disponibles. Más información de los paquetes y lo que se puede hacer con ello revise los archivos apt.txt y aptitude.txt en: <http://132.248.182.159/acl/linux/>

Algunas veces es necesario instalar en la versión estable algún paquete de la versión testing pero que no rompa la estabilidad del sistema, para ello se desarrollo Backports. Para usarlo, lo primero es agregar a sources.list la línea:

```
deb http://ftp.debian.org/debian/ wheezy-backports main
```

también existen para los paquetes contrib y non-free, después de agregar la línea, es necesario correr:

```
# aptitude update
```

Para instalar algun paquete, usar:

```
# apt-get -t wheezy-backports install "package"
```

```
o
```

```
# aptitude -t wheezy-backports install "package"
```

Paquete para eliminar las versiones locales de documentación no usadas

```
# aptitude install localepurge
# localepurge
```

Paquetes de administración y utilerias

```
# aptitude install rcconf mc ntpdate netstat-nat sysstat htop atop deborphan powertop \
iotop iftop ifstat procinfo lsof lshw dvtm screen dstat sysbench dnsutils traceroute \
sysstat glances nmap iptraf wget most fslint gcp jigdo-file grsync tasque clusterssh \
pwgen nmon hwinfo parallel wipe bleachbit dfc tcpdump gddrescue lftp gcp conky-all \
bmon iptables iptables-persistent
# aptitude install tilda yakuake terminator
# aptitude install firmware-linux firmware-linux-free firmware-linux-nonfree
# aptitude install sinfo
# aptitude install rednotebook
```

Paquetes para encontrar archivos duplicados

```
# aptitude install fslint fdupes
```

Paquetes de navegadores de red

```
# aptitude install arora kazehakase flashplugin-nonfree chimera2 chromium conkeror dillo \
dwb edbrowse elinks elvis epiphany-browser iceape iceweasel konqueror links links2 lynx-cur \
midori netrik netsurf netsurf-fb netsurf-gtk surf uzbl w3m
```

Instalar paquetes de KDE

```
# aptitude install kde-full kdevelop kadmin kdeartwork kdegames junior-kde \
kdemultimedia calligra kdesdk kdewebdev kdenetwork kdesdk kdeutils kdevelop \
kdetools education-desktop-kde
```

Oct 27, 15 9:01

paquetes.txt

Page 2/6

Manejo de particiones NTFS

aptitude install ntfs-3g ntfsprogs scrounge-ntfs

Manejo de SAMBA

aptitude install samba smbclient cifs-utils

Manejo de particiones

aptitude install parted partimage gparted testdisk genisoimage

Servidor y cliente de SSH

aptitude install openssh-server openssh-client filezilla

Utilerias de compactación

aptitude install gzip unzip zip pbzip2 bzip2 lbzip2 arj zoo unrar lhasa ncompress \
p7zip p7zip-full p7zip-rar unace unace-nonfree zutils cpio pax unp lzma xz-utils pxz \
pixz kgb rar rarcrack zpaq bsdtar

Manejo de documentos

aptitude install science-typesetting texlive-science libreoffice calligra texstudio
aptitude install texmaker texmacs kile gummi texstudio myspell-es scribus enchant \
texlive-latex-base texlive-latex-recommended myspell-en-us myspell-es
aptitude install pdf-viewer pdfedit msttcorefonts djview4 okular gv zathura diffpdf \
pdf-presenter-console epdfview evince xpdf okular viewpdf.app calibre
aptitude install pdftk pdfgrep xpdf-utils pdfcrack qpdf pdfsam pdfshuffler \
pdfposter pdfchain pdf2djvu gpdfstext catdoc chktex cxref cxref-doc latex2rtf \
antiword unoconv pdfjam a2ps
aptitude install xchm chm2pdf kchmviewer archmage

Editores de gráficos

aptitude install gpaint gimp inkscape imagemagick dia xfig

Messenger

aptitude install pidgin pidgin-guifications pidgin-plugin-pack pidgin-themes finch skype4pidgin
aptitude install kopete kmess empathy

Multimedia y conversión de formatos de audio y video

aptitude install vlc amarok mplayer xbmc ffmpeg mpg123 clementine
aptitude install libxine1-bin libxine1-ffmpeg libxine1-x libxine1-plugins

Paquetes matemáticos

aptitude install science-statistics science-mathematics education-mathematics science-viewing
\
science-numericalcomputation xabacus euler geogebra-kde gretl lybniz mathomatic pari-gp qliss3d
\
rkward xaos yacas bliss geogebra mumps-test nauty netgen netgen-doc eukleides genius pspp \
pdl yorick gnuplot freemat octave qt octave scilab maxima wxmaxima xmaxima mathomatic-primers \
open-axiom julia
aptitude install ipython python-matplotlib python-numpy python-scipy python-scitools \
ipython-notebook python-pandas python-sympy python-nose python-dev spyder \
python-rpy2 python-rpy python-sympy python-nose python-axiom

Lenguajes de programación

aptitude install gfortran gfortran-doc fortran77-compiler fortran95-compiler fortran-compiler cf
ortran
aptitude install mono-complete
aptitude install c++-compiler c-compiler g++ cppcheck splint
aptitude install manpages-dev glibc-doc glibc-doc-reference gcc-doc-base gcc-doc c++-annotations
-pdf

Lenguaje de programación JAVA

aptitude install openjdk-8-jre openjdk-8-jdk openjdk-8-doc

Actualizar la versión por omisión de JAVA

\$ update-java-alternatives -l
update-java-alternatives -s java-1.8.0-openjdk-amd64

Oct 27, 15 9:01

paquetes.txt

Page 3/6

Editores, IDEs, Debuggers y herramientas para programación

```
# aptitude install scite jedit kate gedit nedit emacs medit kscope geany geany-plugins editra qtc
eator \
    anjuta anjuta-extras codelite codelite-plugins tea vim-gtk
# aptitude install eclipse eclipse-cdt netbeans-platform netbeans-ide codeblocks
# aptitude install fte fte-console fte-terminal nano joe vim-gtk
# aptitude install kdiff3 meld diffuse dirdiff kompare numdiff
# aptitude install ddd-doc ddd gdb xxgdb kdbg valgrind
# aptitude install alleyoop astyle c2html java2html code2html c2html autodid
```

Editor GOBBY de texto plano colaborativo multiplataforma cliente

```
# aptitude install gobby
```

Servidor GOBBY

```
# aptitude install gobby infinoted-0.5
```

Lanzar demonio de INFINOTED para crear el certificado (una sola vez)

```
# infinoted --create-key --create-certificate -k /etc/infinote/key.pem -c /etc/infinote/cert.p
em
```

Lanzar demonio de INFINOTED para usar el certificado creado

```
# infinoted -k /etc/infinote/key.pem -c /etc/infinote/cert.pem
```

Resplazo de grep para proyectos grandes ack-grep

```
# aptitude install ack-grep
# dpkg-divert --local --divert /usr/bin/ack --rename --add /usr/bin/ack-grep
```

Control de cambios en un proyecto

```
# aptitude install git git-all gitk gitg
# aptitude install mercurial
# aptitude install subversion rapidsvn
# aptitude install cvs
```

Genrador de documentación

```
# aptitude install doxygen graphviz
```

Manejo de UML

```
# aptitude install umbrello dia
```

Librerías matemáticas

```
# aptitude install gsl-bin libgsl0-dev gsl-ref-html
```

Computo en paralelo usando MPI (es necesario instalar ssh cliente y servidor)

```
# aptitude install lam-runtime xmpi libmpich2-dev mpich2 mpi-default-dev mpi-default-bin openmpi-b
in valgrind-mpi
```

En Debian testing primero quitar estos paquetes:

```
# aptitude --purge remove openmpi-bin openmpi-common libopenmpi1.3
```

Para que en la ejecucion de MPI no pida la clave de usuario:

```
$ ssh-keygen -t rsa
```

En cada pregunta responder con ENTER, para después copiar usando

```
$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

Nota: Si continua pidiendo clave es que también esta instalado rsh o lsh

Cómputo distribuido

```
# aptitude install pvm pvm-dev
# aptitude install perl python-mpi
```

OpenDx Visualization Data Explorer

```
# aptitude install dx dx-dox dxsample
```

HDF5 is a data model, library, and file format for storing and managing data

```
# aptitude install hdfview hdf5-tools
```

Oct 27, 15 9:01

paquetes.txt

Page 4/6

Generador de mallas en 3D
 # aptitude install gmesh

Servidor de web Apache, PHP, Mysql y postgresql
 # aptitude install apache2 libapache2-mod-evasive apachetop
 # aptitude install php5 libapache2-mod-php5 php5-mysql php5-gd phpmyadmin
 # aptitude install mysql-common mysql-client mysql-server mytop mysql-admin ferret mysql-workbench
 mysql-workbench-data
 # aptitude install postgresql postgresql-client postgresql-doc postgresql-contrib

Paquetes para notebook
 # aptitude install wireless-tools acpi-support cpufrequtils acpi wpasupplicant powertop vbetool ac
 pid \
 wicd-curses apmd pcmciautils pm-utils radeontool anacron avahi-autoipd bluetooth iban kmilo
 \
 laptop-detect laptop-mode-tools laptop-net guessnet noflushd bluez-utils bluez-pcmcia-support

Máquinas virtuales
 # aptitude install qemu
 # aptitude install kvm

Paquetes para wireless
 # aptitude install wireless-tools wireshark kismet kwavecontrol wicd-curses
 # aptitude install aircrack-ng macchanger prismstumbler swscanner wavemon wmmwave ifrename
 # aptitude install ndisgtk ndiswrapper-common ndiswrapper-dkms

Construccion de CDlive
 # aptitude install live_helper mbr syslinux

Instalar distribuciones linux/BSD en USB drives
 # aptitude install gnome-disk-utility unetbootin unetbootin-translations
 # aptitude install gksu testdisk

Poder correr aplicaciones de Windows mediante WINE
 # aptitude install wine

Lector de correo tipo PINE
 # aptitude install alpine

Paquetes para intalar impresoras locales y remotas
 # aptitude install system-config-printer system-config-printer-udev cups-pk-helper cups

Paquetes para generar videos a partir de imagenes
 # aptitude install openshot imagination photofilmstrip

Paquetes para tomar videos o imagenes del escritorio
 # aptitude install gtk-recordmydesktop shutter

Paquetes para conocer la resoluciones soportadas de los monitores
 # aptitude install xrandr

FEniCS The FEniCS book, titled Automated Solution of Differential Equations by the Finite Element Me
 thod
 # aptitude install fenics

FreeFem ++ is an implementation of a language dedicated to the finite element method.
 It enables you to solve Partial Differential Equations (PDE) easily.

Oct 27, 15 9:01

paquetes.txt

Page 5/6

```

Es necesario instalar primero los compiladores de Fortran y C, ademas de
# aptitude install freeglut3-dev glutg3
Descargar la ultima version del paquete de la direccion http://www.freefem.org/ff++/
Preparar el paquete como usuario
$ tar -zxvf freefem++-....
$ cd freefem++-...
$ ./configure
$ make
Instalar como root
# make install
Manual en
$ xpdf /usr/local/share/freefem++/freefem++doc.pdf
Ejemplos en
$ cd /usr/local/sare/freefem++/3.8/
Programas instalados
FreeFem++-nw bamg  cvmsh2 drawbdmesh FreeFem++-glx  FreeFem++-x11  FreeFem++-mpi fmedit ffglut

```

GMT (The Generic Mapping Tool)

```

# aptitude install gmt gmt-coastline-data
# aptitude install gmt-examples gmt-manpages gmt-doc-pdf gmt-tutorial-pdf
Tambien hay documentacion disponible en HTML y PS

```

Para usar un escaner

```

# aptitude install gscan2pdf tesseract-ocr tesseract-ocr-spa tesseract-ocr-eng
# aptitude install xsane simple-scan eikazo

```

Generar CDs y DVDs de linux debian

Para generar los CDs y DVDs de debian usando jigdo, instalar

```

# aptitude install jigdo-file
Para generar todos los CDs usar
# jigdo-lite http://cdimage.debian.org/debian-cd/current/amd64/jigdo-cd/debian-6.0.4-amd64-CD-{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31}.jigdo
Para Generar todos los DVDs usar
# jigdo-lite http://cdimage.debian.org/debian-cd/current/amd64/jigdo-dvd/debian-6.0.4-amd64-DVD-{1,2,3,4,5,6,7,8}.jigdo

```

Poner a la hora el equipo con respecto a algun servidor

```

# ntpdate -s -u 1.debian.pool.ntp.org

```

Instalar QGIS y GRASS

Agregar repositorios

```

deb http://qgis.org/debian wheezy main

```

```

deb-src http://qgis.org/debian wheezy main

```

Instalar

```

#gpg --recv-key 47765B75

```

```

#gpg --export --armor 47765B75 | sudo apt-key add -

```

```

# apt-get update

```

```

# aptitude install qgis python-qgis qgis-plugin-grass

```

```

# aptitude install grass grass-dev grass-doc

```

Instalar multimedia

Adicionar la replica:

```

deb http://www.deb-multimedia.org testing main non-free

```

Adicionar la llave

```

# aptitude install deb-multimedia-keyring

```

Instalar para 32 bits

```

# aptitude install libfaad2 libmp4v2-2 libfaac0 alsa-utils lame libmp3lame0 libdvnav4 libdvread
4 libdvdcss2 w32codecs ffmpeg

```

Instalar para 64 bits

```

# aptitude install libfaad2 libmp4v2-2 libfaac0 alsa-utils lame libmp3lame0 libdvnav4 libdvread
4 libdvdcss2 w64codecs ffmpeg

```

Oct 27, 15 9:01

paquetes.txt

Page 6/6

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 28, 13 17:36

prompt.txt

Page 1/1

```
Visualizar configuración
set | grep PS1
```

```
Restaurar el Prompt
```

```
PS1='\[\e]0;\u@\h: \w\a\]\${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
```

```
Prompt [hora] $
```

```
PS1="[\t]$ "
```

```
Poner $ como prompt
```

```
PS1="$ "
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Feb 28, 13 17:38

red-dhcp-estatica.txt

Page 1/2

Para conocer la configuración actual de la red usar:

```
# ifconfig -a
```

Para cambiar la configuración de red a DHCP o IP fijo, como superusuario modificar archivo /etc/network/interface, mediante:

```
# nano /etc/network/interface
```

en lo que sigue, supondremos que se tiene una sola tarjeta de red, cuyo nombre es ETH0.

Para usar DHCP, el contenido del archivo debe quedar:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

Para usar IP fijo, el contenido del archivo debe quedar:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 192.168.10.169
    netmask 255.255.255.0
    network 192.168.10.0
    broadcast 192.168.10.255
    gateway 192.168.10.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.254.205
    dns-search fciencias.unam.mx
```

Si se tienen dos direcciones de red, puede ser una homologada ETH0 y la otra de una intranet ETH1, el contenido del archivo debe quedar:

```
# The loopback network interface
auto lo

iface lo inet loopback

# The primary network interface
allow-hotplug eth0

iface eth0 inet static
    address 132.233.134.159
    netmask 255.255.255.0
    network 132.233.134.0
    broadcast 132.233.134.255
    gateway 132.233.134.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 132.248.204.1 132.248.10.2
    dns-search xxxx.unam.mx

# The secondary network interface
auto eth1
iface eth4 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    broadcast 192.168.1.255
    network 192.168.1.0
```

Feb 28, 13 17:38

red-dhcp-estatica.txt

Page 2/2

```
Para desactivar la red
# /etc/init.d/networking stop
```

```
Para activar la red
# /etc/init.d/networking start
```

En algunos casos es necesario desactivar/activar solo una tarjeta de red de las multiples que con que cuenta el equipo:

```
Desactivar la tarjeta eth0
# ifdown eth0
```

```
Activar la tarjeta eth0
# ifup eth0
```

En el caso que se quiera tener redundancia en la red y si el equipo tiene mas de una tarjeta de red conectada preferentemente a distinto switch, entonces para cambiar la configuración de red de forma redundante, como root modificar archivo /etc/network/interface, mediante:

```
# nano /etc/network/interface
en lo que sigue, supondremos que se tienen dos tarjetas de red, cuyo
nombre es ETH0 y ETH1

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto bond0
iface bond0 inet static
    address 192.168.10.169
    netmask 255.255.255.0
    network 192.168.10.0
    broadcast 192.168.13.255
    gateway 192.168.10.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.254.205
    dns-search fciencias.unam.mx
    slaves eth0 eth1
```

Ahora, ya se puede hacer pruebas conectando y desconectando cables de red de Eth0 y Eth1, manteniendo la conexión de red en el proceso si hay al menos una tarjeta de red conectada

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Mar 05, 14 1:46

samba.txt

Page 1/2

Para instalar samba usamos:

```
# aptitude install smb-client samba smbfs samba-common
```

Para compartir carpetas en linux, podemos hacerlo de varias maneras, una de las cuales y aconsejable es editando el archivo de configuración:

```
/etc/samba/smb.conf
```

y añadir unas líneas que sirven para compartir las carpetas, ahora veremos un ejemplo:

```
;Mi primer Fichero de Configuración de Samba
```

```
[global]
```

```
workgroup = miggrupo
```

```
encrypt passwords = yes
```

```
[test]
```

```
path = /home/gigi/prueba
```

```
read only = no
```

```
guest ok = yes
```

```
[test1]
```

```
path = /home/gigi/prueba2
```

```
read only = yes
```

```
guest ok = yes
```

Que significan estas líneas (respetar las mayúsculas y minúsculas):

- [global]: Es la etiqueta donde ira toda la información que usara todo el mundo, viene a ser las variables globales en programación
- workgroup: grupo de trabajo con el que se comunica la red de windows, en este caso, configuraremos la red de Windows con este grupo
- encrypt passwords: se pone para Windows 98 y Windows NT en adelante
- [test] y [test1]: es la etiqueta que tendrá el directorio con el que accederemos desde Windows a nuestro servidor de Linux
- path: el directorio dentro de nuestro Linux que queremos compartir en este caso tendrá que existir el directorio 'prueba' dándole permisos de lectura y escritura para probar nuestro Samba
- read only: 'yes' significa que los directorios sólo serán en modo lectura. 'no' significa que podrás escribir en los directorios
- guest ok: Permitimos que un usuario invitado entre en el servidor con estos parámetros

Para poder identificar a un usuario de la red, lo tenemos que dar de alta, para eso usamos el programa smbpasswd que nos pedirá el usuario/contraseña de aquellos usuarios que deberían tener capacidad de acceder a los recursos compartidos del Servidor. Recuerda que este usuario debe estar dado de alta como usuario local del sistema.

```
# smbpasswd -a gigi
```

Para hacer un test del fichero de configuración de Samba y comprobar que no nos hemos equivocado

```
# testparm
```

y reinicializar el servicio

```
# /etc/init.d/samba restart
```

Una vez configurado el Cliente de Windows, toca ver si todo funciona bien. Para eso basta con abrir la parte de red y buscar el equipo, o hacer un acceso directo de la forma:

```
\\P90\test (iremos al directorio Prueba)
```

```
\\P90\test1 (iremos al directorio Prueba2)
```

Con esto será suficiente y podremos entrar en el servidor

De todas formas podemos hacer un test para ver las conexiones producidas ejecutando:

```
# smbstatus -a
```

```
Samba version 2.2.2debian-2
```

```
Service      uid      gid      pid      machine
```

```
-----
```

```
IPC$         gigi     gigi     814      k350     (192.168.0.2) Mon Jan  7 22:19:29 2002
```

```
test         gigi     gigi     814      k350     (192.168.0.2) Mon Jan  7 22:19:24 2002
```

Programas asociados con Samba

- smbd: Un demonio que permite compartición de archivos e impresoras sobre una red SMB y proporciona autenticación y autorización de acceso para clientes

Mar 05, 14 1:46

samba.txt

Page 2/2

SMB.

- nmbd: Un demonio que busca a través del Windows Internet Name Service (WINS), y ayuda mediante un visualizador.
- smbclient: Un cliente tipo FTP Unix que puede ser usado para conectar a recursos compartidos por Samba.
- smbtar: Un programa para realizar copias de seguridad de datos sitos en los recursos compartidos, similar al comando Unix "tar".
- nmblookup: Un programa que proporciona búsquedas de nombres NetBIOS sobre TCP/IP.
- smbpasswd: Un programa que permite a un administrador cambiar las passwords encriptadas usadas por Samba.
- smbstatus: Un programa para reportar las conexiones de red actuales hacia los recursos compartidos por el servidor Samba.
- testparm: Un simple programa para validar el fichero de configuración de Samba.
- testprns: Un programa que testea si varias impresoras son reconocidas por el demonio smbd.

Para usar en linux existen varias opciones, segun lo que se requiera:

```
# mount -t smbfs -o username=el_necesario,password=el_requerido //alguna_maquina/algñ_volumen /
punto/de/montaje/
```

```
# mount -t smbfs -o guest //LINUX/FTP //var/ftp
```

```
$ smbclient //alguna_maquina/recurso -U usuario
```

```
$ smbclient //LINUX/FTP -U jbarrios
```

```
$ smbclient -N -L alguna_maquina
```

```
$ smbmount //alguna_maquina/algñ_volumen /punto/de/montaje/
```

```
$ smb://nombre_del_otro_pc o bien smb://ip_privada_del_otro_pc
```

```
$ smbclient -U usuario -L alguna_maquina
```

```
# mount -t cifs //10.0.2.2/windows ./dir -o user=windows,pass=windows
```

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Jun 27, 14 2:52

scp.txt

Page 1/2

Para transmitir archivos y/o directorios de una máquina a otra es recomendable usar el comando SCP de Linux que trabaja con SSH, este tiene una sintaxis similar al del comando cp, con la salvedad que es necesario indicar el usuario, la máquina y el subdirectorío de trabajo del archivo y/o directorio para el destino, fuente o ambos, por ejemplo, si se desea transmitir un archivo por ejemplo de una maquina HOST estando en sesión en otra, se usa la siguiente sintaxis:

```
$ scp archivo.dat usr@HOST:~/Datos/
```

Si se desea transmitir un subdirectorío, por ejemplo de la máquina MMC estando en sesión otra, se usa la siguiente sintaxis:

```
$ scp -r Directorio usr@HOST:.
```

Si se desea copiar un archivo en una máquina remota a nuestra máquina

```
$ scp user@HOST:/home/user/file_name /home/local-username/file-name
```

o

```
$ scp user@HOST:/home/user/file_name .
```

Si se desea copiar de una máquina remota a otra máquina remota

```
$ scp user1@HOST1:/home/user1/file_name user2@HOST2:/home/user2/
```

Si se desea transferir multiples archivos podemos tener

```
$ scp file1.txt file2.txt user@HOST:/home/user/
```

o

```
$ scp user@host:/home/user/{file1.txt,file2.txt} .
```

En el caso que se quiera limitar el ancho de banda en la transmisión de archivos por scp, usar

```
$ scp -l 400 user@server:/home/user/* .
```

En el caso de que se desee usar otro puerto distinto al de imisión (22) usar

```
$ scp -P 4455 file.txt user@HOST:/home/user/file.txt
```

En el caso de querer incrementar la velocidad de transferencia en el uso de scp, la opción más viable es el cambiar la encriptación usada por omisión (AES-128) por otras como BLOWFISH o ARCFOUR mediante

```
$ scp -c blowfish user@server:/home/user/file .
```

o

```
$ scp -c arcfour user@HOST:/home/user/file .
```

Si se desea que no se muestre información de la transferencia de los archivos al usar scp usar

```
$ scp -q SourceFile user@HOST:/home/user/TargetFile
```

o si desea ver más información en la transferencia usar

```
$ scp -v SourceFile user@HOST:/home/user/TargetFile
```

Más información usar

```
$ man scp
```

Si se trabaja en varias máquinas y se desea tener sincronizada una o más carpetas, una opción es usar rsync conjuntamente con ssh, este permite trasmitir solo aquello que ha cambiado, minimizando las comunicaciones, esto se puede usar mediante el comando ssh como se muestra:

Suponiendo que se esta en la máquina y quiere tener sincronizada la carpeta <carpeta> con mas de un equipo, mediante SSH usando un puerto <puerto>, en la máquina y usuario, usar:

```
$ rsync --partial --recursive --links --hard-links --times --verbose --delete --stats <carpeta>
-e 'ssh -p <puerto>' usr@maquina:.
```

por supuesto esto puede hacerse en cualquier dirección, i.e. de la máquina remota a nuestra máquina o viceversa. Además de poder usar rsync de manera

Jun 27, 14 2:52

scp.txt

Page 2/2

local, por ejemplo para hacer respaldos de una parte o todos nuestros archivos en nuestra cuenta y solo actualizar aquellos archivos que han cambiado desde la última actualización.

```
$ rsync --partial --recursive --links --hard-links --times --verbose --delete --stats <carpeta> .
```

en este caso se sincronizaria el contenido de la carpeta <carpeta> en nuestra actual trayectoria.

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 28, 13 17:53

screen.txt

Page 1/1

Supóngase que se quiere ayudar a algún amigo en su computadora y que todas las operaciones que yo realice en ella de forma remota el las vea en su maquina en tiempo real, una forma fácil y ligera es hacerlo con SCREEN en una consola en formato texto.

Para ello necesitamos instalar el programa SCREEN:

```
# aptitude install screen
```

Ya sea como usuario o como super-usuario, ambos usuarios deben hacer login con el mismo nombre de usuario, el usuario que controlara la sesion teclee:

```
$ screen -S mision
```

donde <mision> es un nombre, que el otro usuario debe conocer para tener acceso a la sesión compartida, una forma de conocerlo es usar:

```
$ screen -ls
```

El otro usuario tecleara para tener acceso a la sesión:

```
$ screen -x mision
```

de esta forma, ambos usuarios compartirán la pantalla y verán la misma información así como lo que cada uno de ellos teclea.

Para salir de la sesión, solo teclear Ctrl-A-D, para retomar la sesión usar:

```
$ screen -x mision
```

si desea concluir la sesión solo teclear:

```
$ exit
```

Es posible hacer uso de screen en dos o mas sesiones simultáneamente y en todas ellas se tendrá acceso a la misma visualización, tanto en texto como la salida gráfica generada por la sesión si esta fue inicializada con soporte a X11 y el servidor lo soporta, mediante:

```
$ ssh -X -l usuario maquina
```

P.D. Esto es especialmente útil si se ingresa como usuario y dentro de la sesión se cambia a super-usuario, para permitir enseñar como se instalar, configurar o arreglar algún problema en una máquina personal o servidor. Si el equipo esta en casa conectado con algún proveedor de Internet, es posible hacer uso de OpenVPN para tener acceso a el y mediante SCREEN poder tener acceso al equipo desde alguna máquina en Internet y compartir la visualización de lo que se hace en dicho equipo.

Otras variantes gráficas que se pueden usar si se cuenta con una red rápida es Virtual Network Computing (VNC), la cual tiene múltiples versiones y entre otras cosas permiten compartir escritorios remotos de forma gráfica de Windows, Linux o Unix a cualquier otro ambiente.

 Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Jun 29, 12 17:03

sshfs.txt

Page 1/1

Montar cuentas por medio de SSH como si fueran directorios locales, instalar

```
# aptitude install sshfs afuse
```

Editar /etc/group agregando a los usuarios al grupo fuse

```
# nano /etc/group
fuse:x:119:antonio
```

Activar fuse en el kernel

```
# modprobe fuse
```

Revisar permisos fuse

```
# ls -al /dev/fuse
```

En caso de no tener permisos crw-wr-wr, cambiar usando

```
# chmod 666 /dev/fuse
```

Crear directorio de montaje

```
$ mkdir directoriolocal
```

Montar mediante sshfs

```
$ sshfs usuario@maquina:./directorio directoriolocal
```

Desmontar

```
$ fusermount -u directoriolocal
```


Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma

<http://www.mmc.geofisica.unam.mx/acl/>

Feb 28, 13 17:55

ssh.txt

Page 1/1

Primeramente hay que instalar ssh server, para ello usamos:

```
# apt-get install openssh-server
```

cuando se termina de configurar por default tiene una serie de parámetros predeterminados pero podemos cambiarlos si deseamos así que editamos el archivo, por ejemplo mediante:

```
# nano /etc/ssh/sshd_config
```

el archivo es algo extenso, sólo mencionare las secciones que son de gran importancia:

```
PermitRootLogin no  
X11Forwarding YES
```

esto bloqueara el acceso como root a nuestra máquina por ssh y permitirá la tuteización por X11 a nuestra máquina.

Por último reiniciamos el servicio:

```
# /etc/init.d/ssh restart
```

Para checar si podemos acceder a nuestra máquina con el mismo usuario con el que accesamos, usar:

```
$ ssh mimáquina
```

para acceder a un equipo usando otro nombre de usuario distinto al usuario que con el que accesamos a la máquina, usar:

```
$ ssh -l usuario máquina
```

Para acceder a un equipo en el cual se tiene activado la tunelización de ssh por X11, hay que conectarse mediante:

```
$ ssh -X -l usuario máquina
```

Nota: Si utilizamos cortafuegos, hay que permitir el acceso al puerto de SSH.

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>

Feb 28, 13 17:59

subversion.txt

Page 1/3

Para controlar los cambios en un proyecto (no necesariamente de programación) podemos usar subversion (svn), el nos permite manipular el control de cambios tanto de un proyecto remoto como en uno local.

Para instalar subversion si no viene preinstalado (svn) usar:
`# aptitude install subversion`

En caso de requerir un manejador gráfico instalar rapidsvn, este paquete no tiene editor, comparador o merge por defecto, para que funcione el paquete hay que instalar por ejemplo scite y kdiff3. Para ello instalar
`# aptitude install rapidsvn kdiff3 scite`
 para conocer la ruta de los paquetes auxiliares, por ejemplo de scite usar
`$ whereis scite`
 entonces podemos configurar rapidsvn con los programas necesarios en el menú:
 View==> Preferences==> Programs
 hay que poner la trayectoria completa de por ejemplo el editor (scite), visualizador de web (mozilla), diff tool (kdiff3), Merge tool (kdiff3).

Para hacer el control de cambios en un proyecto local, primeramente debemos construir un árbol para de el importar los archivos iniciales al control de cambios, usamos por ejemplo:

Crear estructura de trabajo para un proyecto
`$ mkdir ~/Proyecto`
`$ cd ~/Proyecto`
`$ mkdir branches tags trunk`
 Agregar fuentes al directorio trunk
`$ cp fuentes ~/Proyecto/trunk/`

Crear un repositorio por ejemplo ProyectoSubversion (uno por cada proyecto)
`$ svnadmin create /home/svn/SVNProgramacion`

Importar la estructura básica creada con el nombre Proyecto
`$ svn import ~/Proyecto file:///home/svn/SVNProgramacion -m "Importacion inicial"`
 una vez creada esta estructura podemos olvidarnos de ella (podemos borrarla).

Los diferentes metodos de acceder al repositorio son:

<code>file:///</code>	Repositorio en el mismo servidor
<code>http://</code>	Repositorio accedendo por http
<code>https://</code>	Repositorio accedendo por https
<code>svn://</code>	Repositorio accedendo por svn
<code>svn+ssh://</code>	Repositorio accedendo por svn tunelizando por ssh

Obtener copia de trabajo (que es donde realmente trabajaremos)
`$ svn checkout file:///home/svn/SVNProgramacion Trabajo`

así, ya estamos listos para empezar a trabajar con nuestro proyecto.

Adicionar un archivo
`$ svn add <arch>`

Ver diferencia de archivos
`$ svn diff`

Ver diferencia de el <archivo>
`$ svn diff <arch>`

Borrar archivo
`$ svn delete <arch>`

Copia de un archivo
`$ svn copy <arch> <arch>`

Feb 28, 13 17:59

subversion.txt

Page 2/3

```

Mover el archivo
$ svn move <arch> <arch>

Crear un directorio dentro de la copia local
$ svn mkdir <dir>

Revertir los cambios del archivo <arch>
$ svn revert <arch>

Mantener la copia local actualizada
$ svn update

Mandar una actualización al control de cambios
$ svn commit -m "Actualizacion"
$ svn commit <arch> -m "Actualizacion"
$ svn commit <directorio> -m "Actualizacion"

en caso de encontrar conflictos en la actualización, por cada archivo en
conflicto, se generaran tres archivos auxiliares con extension .mine, .r<n>,
.r<m>. Despues de hacer los ajustes pertinentes hay que avisar al control de
versiones mediante
$ svn resolved <arch>
una vez actualizados los cambios seran borrados los archivos auxiliares creados
por el control de versiones, algunos de los estados de los archivos y carpetas
pueden ser:
(U) Actualizado, (A) Añadido, (D) Borrado, (R) Remplazado, (G) Fusionado cambios,
(C) Conflictos

Ver el logs de cambios
$ svn log
$ svn log <arch>
$ svn log --revision <n>
$ svn log <arch> --revision <n>:<m>

Ver el log de cambios y a que archivos se aplicaron
$ svn log --verbose

Ver el log de cambios de una determinada revisión hasta ahora
$ svn log --verbose --revision 2273:HEAD

Ver un archivo de una determinada versión
$ svn cat <archivo> --revision <n> <arch>

ver el Estado de todos los componentes del control de cambios
$ svn status --verbose

Mostrar el estatus local versus el repositorio
$ svn status --show-updates --verbose

Mostrar las diferencias con respecto al control de cambios
$ svn diff

Mostrar las diferencias con respecto a una revisión previa en el control de cambios
$ svn diff --revision <n> <archivo>

Ver información de un directorio o archivo
$ svn info <achiv | dir>

Descargar una copia local de una version particular, por ejemplo usando
$ svn checkout --revision {<n>} file:///home/svn/SVNProgramacion Trabajo
la cual es accesible tambien por fecha, hora o combinacion de ellas

Si se trabaja en branches y se quiere actualizar con una parte de trunk se puede
usar MERGE para mantener estas actualizadas, primeramente posicionarse en branches
y:

Conocer lo que es suceptible para hacer el merge
$ svn merge --dry-run file:///var/svn/ressim/trunk/src/

Para hacer el merge
$ svn merge file:///var/svn/ressim/trunk/src/

Para que los cambios se queden permamente en el control de cambios
$ svn commit -m "Update branches"

```


Feb 28, 13 17:59

subversion.txt

Page 3/3

En el caso que se use SSH para acceder a subversion en algun servidor y se quiere que no se solicite la clave de acceso del usuario al aplicar cambios, entonces en el servidor hacer:

```
$ ssh-keygen -t rsa
contestar por omisión a todas las preguntas con ENTER, despues copiar usando
$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
los archivos id_rsa* generados en el servidor, se deberan de copiar a la máquina
de trabajo usando por ejemplo:
$ scp usuario@servidor:~/.ssh/id_rsa* ~/.ssh/
para probar que todo esta correcto hacer un ssh para comprobar que no se solicita
la clave de acceso
$ ssh -l usuario servidor
```

mas informacion en: <http://subversion.tigris.org/>

Algo de linux: <http://www.mmc.geofisica.unam.mx/acl/linux/>

Antonio Carrillo Ledesma
<http://www.mmc.geofisica.unam.mx/acl/>