

DISICO

Instalación de servicios FreeBSD

Manual

Instalación de Servicios en FreeBSD 6.1

1.- Instalación y configuración de un Servidor – FreeBSD 6.1

1.- Configurando la red

El servidor educnet-clon cuenta con la siguiente configuración:

IP:10.100.6.215 /24

GW:10.100.6.254

DNS:10.50.1.16

Nombre Host: Educnet-Clon

1.1.- Para la ip:

Ifconfig em0 10.100.6.215 netmask 255.255.255.0

1.2.- Para la puerta de enlace (Default gateway)

route add default gw 10.100.6.254

1.3.- Para el DNS

vi /etc/resolv.conf

nombre de host

#vi /etc/hosts

ponemos:

127.0.0.1 localhost.decom-uv.cl localhost

10.100.6.215 clon-educnet.decom-uv.cl clon-educnet

10.100.6.215 clon-educnet.decom-uv.cl.

Luego editamos **/etc/host.conf**, verificando que se encuentre **hosts**, antes que **bind**

#vi /etc/host.conf

hosts

bind

dns

Luego editamos **/etc/rc.conf**, que es el archivo que guarda la configuración con que arranca el equipo. configurando:

```
#vi /etc/rc.conf
ifconfig_fxp0="inet 10.100.6.215 netmask 255.255.255.0"
defaultrouter="10.100.6.254"
hostname="educnet-clon.decom-uv.cl"
nota: para nuestro siguiente acceso al sistema, debería aparecernos en el prompt:
educnet-clon#
```

2.- Instalación de aplicaciones Port

Necesitamos instalar **Diablo** (java para FreeBSD) y **Postgresql 8.1**, para comenzar, es necesario saber con que versión de diablo cuenta nuestro sistema FreeBSD además de la ubicación.

2.1. Buscar el port

```
#cd /usr/ports                                //donde se encuentran todas las aplicaciones
#make search name=diablo                      //diablo es la aplicación a buscar
Port: lsof-4.56.4
Path: /usr/ports/java/diablo-jre13
Info: Lists information about open files (similar to fstat(1))
Maint: obrien@FreeBSD.org
Index: sysutils
B-deps:
R-deps:
```

Nota: Path nos indica la ubicación de diablo, primera aplicación a instalar.

2.2: Entrar a la aplicación

Entramos a **/usr/ports/java/diablo-jre13**
Deben existir los siguientes directorios:

```
Makefile
distinfo
files
pkg-descr
pkg-plist
```

Hacemos **make** y **make install**

Si existe un error sobre vulnerabilidad debemos actualizar la BD de de vulnerabilidades.

Nota: nos damos cuenta que la versión de diablo con la que cuenta nuestro sistema corresponde a la 1.3, lo cual es demasiado antigua, es necesario además de actualizar la lista de vulnerabilidades, actualizar nuestra lista de puertos (ports).

2.3.- Actualización de la BD de vulnerabilidades

Entrar a `/usr/ports/security/portaudit`

Ejecutar
`#make`

y luego
`#makeinstall`

Una vez ejecutado dichos comando, se debe ejecutar:

`#portaudit -F` **//nos descarga la nueva BD de vulnerabilidades**

2.3.- Actualización de la lista de puertos (ports).

Ejecutamos el siguiente comando:

`#portsnap fetch extract`

Antes de empezar a instalar los servicios se debe tener los **Ports** actualizados, estos se pueden descargar de la siguiente dirección <ftp://ftp.freebsd.org/pub/FreeBSD/ports/ports/>

3.- Instalando Bash

Una vez bajados los Ports se pueden dejar en un directorio como ports2 o renombrar el directorio original como ports.ori y el original se deja como ports, eso a gusto de la persona, lo importante es realizar la actualización desde los ports descargados y no los que trae. En este caso se utilizo el ports2

```
cd /ports2/port/shells/bash
```

```
make
```

```
make install
```

4.- Habilitando ssh

4.1.- Entro al archivo `/etc/rc.conf`

Ahí hay que poner:

```
sshd_enable = "YES"
```

4.2.- Luego ejecuto ssh

```
/etc/rc.d/sshd star
```

4.3.- En caso de no partir ssh, debería agregar estas línea de código al archivo

```
etc/ssh/sshd_config
```

```
IgnoreRhosts yes
```

```
RhostsRSAAuthentication no
```

```
RhostsAuthentication no
```

```
IgnoreUserKnownHosts no
```

```
PrintMotd yes
```

```
StrictModes yes
```

```
RSAAuthentication yes
```

```
PermitRootLogin yes
```

```
PermitEmptyPasswords no
```

```
PasswordAuthentication yes
```

4.4.- Posibles problemas con ssh

Uno de los problemas que se pueden presentar posterior a la instalación por ssh, es que al servidor donde intentamos acceder haya generado nuevamente las claves (trabaja con claves publicas y privadas), es decir, se haya desinstalado ssh y luego instalado.

El problema que se muestra en nuestro ssh (cliente) es el siguiente:

```
alumnos# ssh root@192.168.20.6
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the DSA host key has just been changed.
The fingerprint for the DSA key sent by the remote host is
69:ca:9d:10:03:c9:20:ba:a7:dc:46:d4:b7:7c:e5:44.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending key in /root/.ssh/known_hosts:7
DSA host key for 192.168.20.6 has changed and you have requested strict checking.
Host key verification failed.
```

En este caso se borra `.ssh/known_hosts`. Y se volverá a poder entrar por ssh desde esta cuenta. Este problema se puede presentar en cualquier cuenta de usuario que se tenga en la máquina por ejemplo puede darse con `/home/1998/mferrand/.ssh/known_hosts`. Cabe señalar que el problema es de una cuenta en específico y no de TODAS las cuentas, esto fue probado entrando con otra cuenta mientras persistía el problema desde `root` y `mferrand` que se encontraban en la máquina desde donde se quería acceder al servidor

Problema Host key verification failed. En Jaulas

Este problema se presenta cuando se tienen servidores virtuales y ocurre cuando las interfaces de los servidores virtuales (Jails) son cambiadas, entiendase por interfaces los alias que utilizan para identificarse las distintas jaulas. El problema tiene relación con lo mismo explicado anteriormente, y se soluciona “desinstalando” (parando ssh y comenarlo de rc.conf) ssh y luego instalándolo para que se generen las llaves públicas y privadas nuevamente, en caso que no se pueda ingresar una de las posibilidades es que pase lo señalado en el parrafo anterior, es decir, la cuenta del servidor desde donde se está accediendo mantenga las llaves (publicas y privadas) antiguas.

5.- Instalando Apache 1.3

Para que apache parta la máquina debe tener nombre, para que la máquina tenga nombre deben estar configurados los archivos a continuación

vi /etc/hosts

vi /etc/host.conf

vi /etc/rc.conf

Si los nombre no coinciden en estos archivos no se podrá iniciar apache

Una vez configurados estos archivos se realiza lo siguiente

cd /usr/ports/www/apache13

make install clean

apachectl start //parte apache

otros:

apachectl stop //para apache

apachectl restart //reinicia apache

Para desinstalar una version:

cd /usr/ports/www/apache13

make deinstall //desinstala apache

En el archivo **/etc/rc.conf**

Se agrega

apache_enable = "YES" //con esto cada vez que se reinicie la máquina apache partirá

6- Instalando apache13 (Apache Mysql 5 PHP5)

```
amp# cd /usr/ports/www/apache13
```

```
amp#make
```

```
amp#make install
```

Al ejecutar apache **apachectl start**, es posible que el comando no sea reconocido por lo que habrá que salir de ese Prom. con **exit** y luego ejecutar **apachectl Stara**

Para que el comando que viene a continuación pueda ejecutarse se ingresa por el shell **tcsh**

```
#apachectl start
```

>>Otra forma de poder ejecutar apache es:

```
/usr/local/sbin/apachectl start
```

Luego se agrega **mysql_enable="YES"** al archivo **/etc/rc.conf**

7.- Instalando PHP5

Al instalar PHP5 con Apache13 debo instalar otras aplicaciones como php5-mysql y php5-extensions, todas estas instalaciones quedarán descritas en esta sección

7.1.- Instalando PHP5

```
cd /usr/ports/lang/php5
```

```
make install clean
```

Dar Soporte PHP a Apache13

Al realizar `make install clean` me aparecerá una pantalla azul, en la cual se deberá agregar algunas opciones imprescindibles como `apache` y `cgi` (esta generalmente viene por defecto)

##Este paso de comprobación de php se puede omitir, o realizar una vez instaladas las otras aplicaciones como php5-extensions y php5-mysql#####

> Para probar php5 agregamos una página hecha en **php** en el directorio **/usr/local/www/data** (para este caso debería agregarse en la jaula que se está instalando)
Dentro del directorio data agregamos una página **pruebaphp.php**

> Ingresamos a la página por ejemplo: **10.100.6.204/pruebaphp.php**

Para verificar que los módulos de PHP4 se encuentren instalados hay que dirigirse a **vi /usr/local/etc/apache/httpd.conf**

En este archivo deberán aparecer

```
LoadModule php5_module    libexec/apache/libphp5.so
```

Y mas abajo aparecerá

AddModule mod_php5.c

Luego se reinicia apache (**apachectl restart**) y al ingresar a la ip del servidor deberá aparecer la página de PHP **PHP Version 5.**

7.2.- Instalando PHP5-Extensions

cd /usr/ports/lang/php5-extensions

make config

Acá aparecerán una serie de opciones que solo algunas se deberán marcar además de dejar las que aparecen por default, las opciones a marcar son las siguientes (estas aparecerán en una pantalla azul)

select type, curl, dom, gd, imap, mbstring, mcrpyt, mysql, mysqli, pcre, posix, session, simplexml, xml, xmlreader, xmlwriter, zlib

make install

make clean

cd /usr/local/etc

cp php.ini-recommended php.ini

7.3.- Instalando PHP5-mysql

Acá se da el soporte de mysql en PHP, se produce la conexión entre ambas aplicaciones, para ver si tengo instalado php5-mysql realizo el siguiente comando

pkg_info|grep php|grep mysql

Si el commando arroja algún resultado quiere decir que la aplicación está instalada, en caso contrario (de no arrojar nada) hay que instalarla.

cd /usr/ports/database/php5-mysql

make

make install

make clean

Observación: Es probable que no inicie php5 por lo que hay que meterse al archivo `vi /usr/local/etc/apache/httpd.conf` y cambiar todos los php4 por php5

8.- Instalando webmin

8.1.- Webmin se encuentra en

```
cd /usr/ports/sysutils/webmin  
make install clean
```

8.2.- Una vez instalado se agrega al archivo

```
vi /etc/rc.conf
```

Se agrega `webmin_enable="YES"`

8.3.- Para ejecutarlo se debe ir a la siguiente ruta

```
cd /usr/local/lib/webmin
```

se ejecuta de la siguiente forma

```
./setup.sh
```

9.- Instalar diablo

Con la nueva lista de puertos, nos damos cuenta que dentro de `/usr/ports/java`, existen diferentes directorios relacionados con diablo, claro que ahora, con la versión 1.5, la más actualizada hasta el momento.

Entramos a `/usr/ports/java/diablo-jdk15/`

Si ejecutamos `make` y/o `make install`, nos entregará un error de la no existencia de un archivo, el cual debe ser descargado.

9.1.- Descarga de Diablo

Descargar **diablo-caffe-freebsd6-i386-1.5.0_07-b01.tar.bz2** y copiarlo dentro de la carpeta **/usr/ports/distfiles** (en caso que se este descargando de un sitio que haya que mandar un “submit” o “aceptar”, esto se puede hacer a través de interfaz grafica de links luego, ir a **/usr/ports/java/diablo-jdk15/**

Se realiza los siguientes comandos en el mismo orden que aparece:

```
make  
lmake install
```

No nos debería presentar mayores problemas, por lo cual tendríamos java instalado.
comprobar, ejecutar:

```
#java -version
```

Debería entregarnos la versión que se encuentra instalada, en nuestro caso, 1.5

10.-Instalación de Postgresql 8.1

Nos dirigimos al puerto que se encuentra **posrgresql 8.1 de la siguiente forma:**

```
cd /usr/ports/databases/postgresql-81-server
```

Si ejecutamos make puede que nos lance un error sobre setear variables Kerberos.
Ejecutamos:

#make config

Marcamos toda la configuración de Server a excepción de Kerberos, la cual a su vez, no es aplicación de Server. Confirmamos con OK ahora si ejecutamos:

#make install

Durante la instalación puede que nos pregunte por un paquete, simplemente confirmar con OK.

Finalmente ejecutamos:

#make clean

Añadimos postgresql para que se cargue al inicio en el archivo **/etc/rc.conf**

```
#echo 'postgresql_enable="YES"' >> /etc/rc.conf
```

para verificar dicha línea, ejecutar:

```
#cat /etc/rc.conf
```

donde **postgresql_enable= "YES"** debería estar al final.

10.1.- Creando initdb

Debemos crear **initdb** que nos servirá para crear una nueva **"database cluster"**, que no es más que una colección de base de datos que serán manejadas por una única instancia del servidor.
ejecutamos:

```
# mkdir /usr/local/pgsql/data
```

```
# chown postgres:postgres /usr/local/pgsql/data
```

3.

Si todo sale bien ya podemos hacer uso del programa **initdb** de la siguiente forma:

```
# su -l postgres -c "initdb -D /usr/local/pgsql/data -W -A md5"
```

Salida del comando anterior (en español):

Los archivos de este cluster serán de propiedad del usuario «postgres».

Este usuario también debe ser quien ejecute el proceso servidor.

corrigiendo permisos en el directorio existente /usr/local/pgsql/data ... hecho
creando directorio /usr/local/pgsql/data/global ... hecho
creando directorio /usr/local/pgsql/data/pg_xlog ... hecho
creando directorio /usr/local/pgsql/data/pg_xlog/archive_status ... hecho
creando directorio /usr/local/pgsql/data/pg_clog ... hecho
creando directorio /usr/local/pgsql/data/pg_subtrans ... hecho
creando directorio /usr/local/pgsql/data/pg_twophase ... hecho
creando directorio /usr/local/pgsql/data/pg_multixact/members ... hecho
creando directorio /usr/local/pgsql/data/pg_multixact/offsets ... hecho
creando directorio /usr/local/pgsql/data/base ... hecho
creando directorio /usr/local/pgsql/data/base/1 ... hecho
creando directorio /usr/local/pgsql/data/pg_tblspc ... hecho
seleccionando el valor para max_connections ... 40
seleccionando el valor para shared_buffers ... 1000
creando archivos de configuración ... hecho
creando base de datos template1 en /usr/local/pgsql/data/base/1 ... hecho
inicializando pg_authid ... hecho
Ingrese la nueva contraseña del superusuario:
Ingrésela nuevamente:
estableciendo contraseña ... hecho
habilitando tamaño de registro ilimitado para tablas de sistema ... hecho
inicializando dependencias ... hecho
creando las vistas de sistema ... hecho
cargando pg_description ... hecho
creando conversiones ... hecho
estableciendo privilegios en objetos predefinidos ... hecho
creando el esquema de información ... hecho
haciendo vacuum a la base de datos template1 ... hecho
copiando template1 a template0 ... hecho
copiando template1 a postgres ... hecho
Completado. Puede iniciar el servidor de bases de datos usando:
postmaster -D /usr/local/pgsql/data
o pg_ctl -
D
/
usr/local/pgsql/data -
|
archivo_de_registro start
Clave superusuario: escotilla

11.- Instalando Samba 3.0.21a y creando quotas de usuarios

11.1.- Para instalar samba

```
cd /usr/ports/net/samba3  
make  
make install
```

11.2.- en `/etc/rc.conf` agrego `samba_enable="YES"`

11.3.- Para agregar las máquinas que se conectarán al servidor se puede hacer por **webmin** o por **ssh**, en el ejemplo a continuación se mostrará como se hace por **webmin**

Agregando maquinas

Sistemas / nuevo usuario

Nombre del usuario: LAB02PC26\$

Nombre real: LAB02PC26

Shell: /usr/sbin/nologin

Grupo primario: samba

11.4.- En SSH

Acá creo la password con los cuales los pc se conectarán al servidor (samba)

El comando para poner la password es:

Smbpasswd root

New SMB: prueba

Confirmar passwd: prueba

11.5.- En W2K

Se agregan las máquinas al dominio en Windows, para que sean reconocidas en **samba**. Te vas a:

a.- Mi PC / propiedades / nombre de equipo (pestaña) / cambiar (botón)

b.- Desactivar "grupo de trabajo" y Activar "dominio".

c.- Poner nombre al dominio ej: LABDECOM

d.- Aparecerá

login:

passwd:

login:root

contraseña: prueba

e.- Aún no tenemos los usuarios creados por lo que en caso de haber un gran número de usuarios resultaría agotador llenar todos los datos de usuarios, por lo que se hace es cargar un **archivo txt** en **webmin**. Donde se carga este archivo es **Sistema / usuarios y grupos / crear y modificar usuarios desde archivos por lotes/**

Ejecutar Archivo por Lotes

Este formulario le permite crear, modificar o borrar varios usuarios de una vez desde un archivo de texto cargado o local. Cada línea del archivo especifica una acción a realizar, dependiendo de su primer campo. Los formatos de línea son:

```
crear: username:passwd:uid:gid:realname:homedir:shell:class:change:expire  
modificar: oldusername:username:passwd:uid:gid:realname:homedir:shell:class:change:expire  
borrar: username
```

En las líneas **create** (**crear**), si el campo de uid se deja vacío, Webmin asignará automáticamente una UID. Si el campo de gid se deja vacío, Webmin creará un nuevo grupo con el mismo nombre que el usuario. Los campos username (nombre de usuario), homedir (directorio inicial) y shell deben de ser suministrados para cada usuario - todos los otros campos se permite que estén vacíos. Si el campo de passwd (contraseña) se deja en blanco, no se asigna contraseña al usuario. Si contiene sólo la letra x, la cuenta se bloqueará. En caso contrario, el texto en el campo es tomado como contraseña en texto plano, y posteriormente encriptada.
En las líneas **modify** (**modificar**), un campo vacío indica que el correspondiente atributo de usuario no ha de ser modificado.

Fuente de datos de lotes

Archivo cargado

Archivo en servidor

Texto en la caja inferior

El archivo que se carga mantiene esta estructura, se puede hacer de forma automática con funciones de excel

```
create:aalvarez:620037:5005:1000:"Alejandro Alvarez":/home/2007/aalvarez:/usr/local/bin/bash:default::  
create:aamaya:986679:5006:1000:"Amanda Amaya":/home/2007/aamaya:/usr/local/bin/bash:default::  
create:acordova:505258:5020:1000:"Angelo Cordova":/home/2007/acordova:/usr/local/bin/bash:default::  
create:agardell:328955:5031:1000:"Angelo Gardella":/home/2007/agardell:/usr/local/bin/bash:default::  
create:ahernand:135485:5039:1000:"Alexis Hernandez":/home/2007/ahernand:/usr/local/bin/bash:default::  
create:amaturan:599493:5053:1000:"Alex Maturana":/home/2007/amaturan:/usr/local/bin/bash:default::
```

Una vez cargado el archivo aparecerá los siguiente:

Creado usuario aalvarez
Creado usuario aamaya
Creado usuario acordova
Creado usuario agardell
Creado usuario ahernand

En caso de existir duplicaciones **webmin** mostrará cual es el usuario que esta duplicado

De esta forma tengo todos los usuarios creados solo falta agregarlos a samba. Para agregarlos a samba se puede hacer por ssh. Utilizando el siguiente comando se agregará los usuarios a samba, esto quiere decir que cuando cada persona se conecte con su PC (que ya esta registrada la máquina) se estará logeando al servidor.

11.6.- Creación de cuotas

1º Se debe añadir en el archivo de configuración del Kernel: **options QUOTA**

Esta opción se agrega en

cd /usr/src/sys/i386/conf

vi ALUMNOS //En este archive agregamos la “options QUOTA”

En el caso de nuestro servidor el Kernel se respaldo estando dentro del directorio **cd /usr/src/sys/i386/conf** de la siguiente forma

cp GENERIC GENERIC.resp

Esto se realiza en caso de cualquier problema tenga el kernel. En nuestro caso en específico se renombro el kernel **GENERIC** por **ALUMNOS**, se realizó de la siguiente forma

mv GENERIC ALUMNOS

2º En:

vi /etc/rc.conf

enable_quotas="YES"

check_quotas="YES"

3º Una vez realizadas estas modificaciones se compila el Kernel

- Dentro del directorio **usr/src/sys/i386/conf** Se pone **/usr/sbin/config ALUMNOS**
- **cd ../compile/ALUMNOS**
- **#make depend //compilando el kernel**
- **#make**
- **#make install //Instalando el nuevo kernel ALUMNOS**

4º Asignar cuotas a particiones **/etc/fstab**

En este caso (alumnos) se debe asignar las cuotas a sus discos virtuales (conectados a través de samba) y as vez asignar cuotas al correo. En este caso las cuotas de los alumnos quedaron en el directorio **Home** y las de correo en **var**

Para las cuotas del home se asignan 50 MB

Para las cuotas del var se asignan 15 MB

En el caso para el Home quedaría de la siguiente forma

/dev/da0s1d /home ufs rw, userquota

En el caso para el var quedaría de la siguiente forma

/dev/da0s1d /var ufs rw, userquota

5º Luego asigno por ssh el tamaño de ambas cuotas (samba y correo) al usuario:

quota -u mferrand

/home: kbytes in use: 6482, limits (soft = 50001, hard = 50001)

inodes in use: 246, limits (soft = 0, hard = 0)

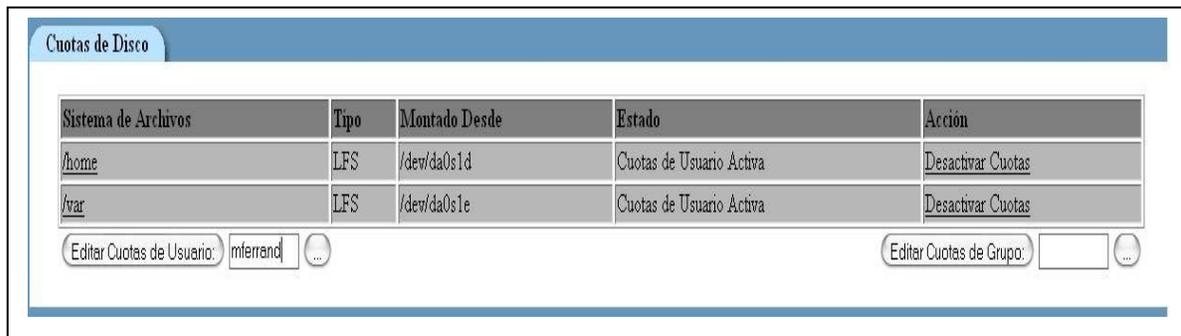
/var: kbytes in use: 154, limits (soft = 15000, hard = 150000)

inodes in use: 1, limits (soft = 0, hard = 0)

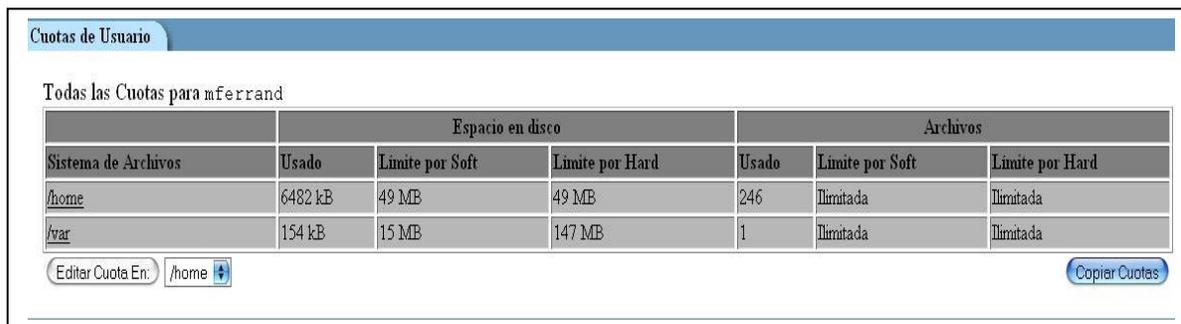
6.- Por **ssh** y **webmin** se pueden copiar cuotas de un usuario a un grupo determinado, en este caso se copió por webmin a todo el grupo **alm** el cual contiene a los usuarios que se quiere copiar la cuota.

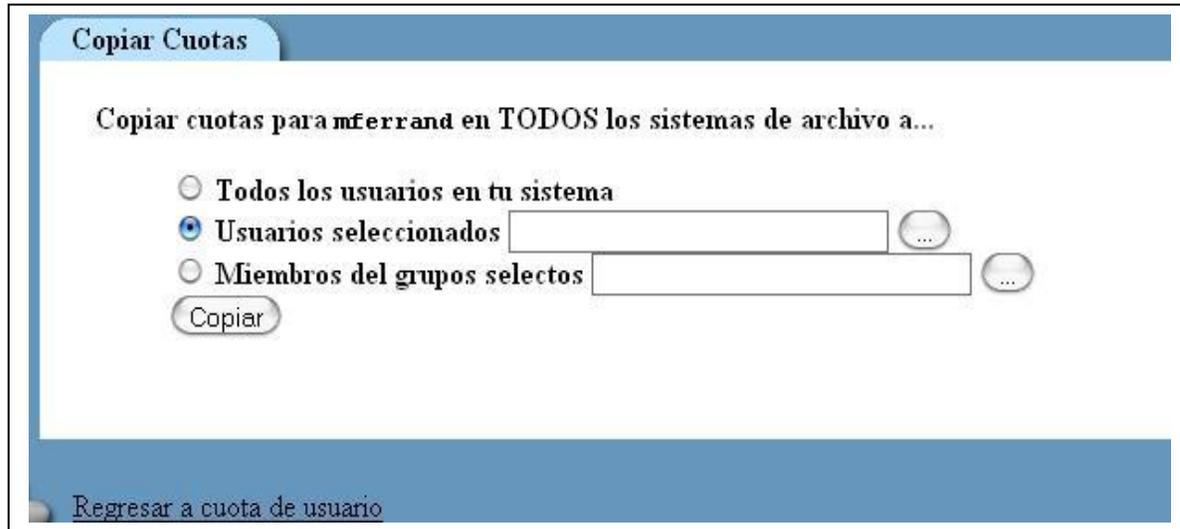
Sistemas/Cuotas de disco

En editar cuotas de usurita se agrega el usuario que replicará su cuota al resto del grupo



Luego se elige la opción **copiar cuotas**





12.- configuración de UPS - OPTIMUS

```
cd /usr/ports/syutils/apcupsd
make install clean
```

Agregar en el archive **rc.conf**
apcupsd_enable="YES"

Luego en el siguiente archivo se debe de configurar

```
vi /usr/local/etc/apcupsd/apcupsd.conf
```

//Se configura

```
UPSCable      "número cable"      940-0024c
UPStype       APSSmart          //modelo
DEVICE
```

Luego se ejecuta

```
/usr/local/etc/rv.d/apcupsd start
```

```
/usr/local/sbin/apcaccess status
```

Luego de configurar la UPS aparece el siguiente error:

```
FATAL ERROR in apcaccess.c at line 252
Tcp_open: cannot connect to server local host on port 3551
ERR-Connection refused
```

El error se produjo por el cable, se cambio el cable

- Luego dirigirse a

`/usr/local/etc/apcupsd`

Se copia el directorio **cgi** completo a `/usr/local/www/cgi-bin`

- Luego editar el archivo **hosts.conf** que se encuentra en `/usr/local/etc/apcupsd`
En este archivo agrego la ip del servidor en este caso 192.168.1.10 y agrego el nombre "alumnos".
- En esta ruta `/usr/local/www/cgi-bin/cgi` cambiar los permisos a todos los archivos
`Chmod 777`
- **IMPORTANTE:** Reiniciar
`/usr/local/etc/rv.d/apcupsd restart` luego comenzar con el servicio `/usr/local/etc/rv.d/apcupsd start`

13.- Crear usuarios

13.1.- Primero se crea el grupo

```
[root@optimus ~]# pw groupadd decom
```

Se visualiza con

```
[root@optimus ~]# cat /etc/group
```

13.2.- Agregar usuarios

```
[root@optimus ~]# adduser -v
```

Para borrar un usuario

```
rmuser nombredelusuario
```

IMPORTANTE: Para que los usuarios creados puedan utilizar **su** para poder cambiar a root , los usuarios al ser creados deben ser agregados al grupo **wheel**

14.- Configurando ssh en jaulas de tesis

Para que los usuarios puedan entrar por ssh a sus jaulas se pasa primero por el servidor de alumnos, y luego se salta a la jaula correspondiente, esto se realiza de la siguiente forma.

En el **servidor de alumnos**, se crea un script, esto se realiza en el directorio root:

```
vi /root/dramirez  
#!/bin/sh  
ssh root@192.168.20.8
```

El usuario dramirez que se encuentra creado en el servidor de alumnos pasa a su jaula que se encuentra con la ip 192.168.20.8.

Este script se encuentra en **/root** y tiene como permiso **755**

Luego en el servidor de alumnos se modifica el master.passwd sin antes haberlo respaldado, se ejecuta vipw y se cambia la forma de ingresar

Se tenía

```
azamora:$1$Xu3Rd/fQ$0P2ADbu2/RUN19/CYkKtA0:1377:1000::0:0:Amador Zamora
Nunez:/home/2005/azamora:/usr/local/bin/bash
```

Se cambia a

```
azamora:$1$Xu3Rd/fQ$0P2ADbu2/RUN19/CYkKtA0:1377:1000::0:0:Amador Zamora
Nunez:/home/2005/azamora:/root/azamora-ssh
```

15.- Cambiar Zona horaria

Es importante verificar la zona horaria de la jaula, ya que nos encontramos con aplicaciones que trabajan con la hora. Para realizar esto se debe realizar lo siguiente

Dirigirse hasta

```
amp3#cd /usr/share/zoneinfo/America
```

Luego copiar el archivo Santiago que aparece dentro del directorio America.

```
amp3#cp Santiago /etc/localtime
```

Luego se verifica que la hora este correcta con el comando date

```
amp3# date
Wed Jun 11 14:56:46 CLT 2008
```

Para cambiar la hora a través de comandos se realiza lo siguiente

```
amp3# date año mes día hora minutos
```

ejemplo

```
amp3# date 0808211000
```

Esto equivaldría a 21/08/2008 10 rct

16.- Instalando bind9 para DNS (Para este caso DNS Externo)

```

Primero se instala el bind9
jail2# cd /usr/ports/dns/bind9
jail2#make
jail2#make install

```

En el `/etc/namedb/named.conf` de `scox` aparecen las siguientes líneas que deben ser copiadas en el nuevo `scox (Ultraman)` en `named.conf` para poder realizar el **DNS Externo**

>> En `/etc/namedb/named.conf`

```

zone "decom-uv.cl"{
    type master;
    file "/etc/namedb/master/decom-uv.cl";
};

```

Luego en el directorio `/etc/namedb/master` creo el archivo `decom-uv.cl` y agrego las siguientes líneas:

En `/etc/namedb/master`

```

$ttl 38400
decom-uv.cl. IN SOA          behemoth.decom-uv.cl. root.localhost. (
                1071154044
                10800
                3600
                604800
                38400 )
decom-uv.cl. IN      NS      scox.decom-uv.cl.
www.decom-uv.cl. IN  A      200.14.68.2
scox.decom-uv.cl.  IN  A      200.14.68.2
decom-uv.cl. IN     MX      10 200.14.68.3
alumnos.decom-uv.cl. IN A      200.14.68.3
alumnos.decom-uv.cl. IN MX    10 alumnos.decom-uv.cl.
siga.decom-uv.cl.  IN  A      200.14.68.107
director.decom-uv.cl. IN A      200.14.68.107
asterisk1.decom-uv.cl. IN A      10.100.6.166

```

>> Agregando ip donde va a escuchar el DNS (En este caso es la de la jaula)

En el archivo **/etc/namedb/named.conf** debo agregar la ip donde va a escuchar el **DNS**, en el caso de la jaula que se está configurando será la 10.100.6.181 Esto sería en:

```
jail2# vi /etc/namedb/named.conf
```

>>Debería aparecer algo así:

```
// If named is being used only as a local resolver, this is a safe default.  
// For named to be accessible to the network, comment this option, specify  
// the proper IP address, or delete this option.  
listen-on { 127.0.0.1;10.100.6.170 };
```

>> Agregar lineas rc.conf

Agregar en el **/etc/rc.conf** de la jaula

```
named_enable="YES"
```

>>Partiendo el named (DNS)

```
jail3# /etc/rc.d/named start
```

>>Comprobando su configuración

Cambiar el DNS que se tiene actualmente ya sea en Win o Linux y poner el DNS perteneciente a la jaula donde se configuró el DNS, y comprobar si hay navegabilidad