PRACTICA #15

"SAMBA"

OBJETIVO: Aprender la configuración de un servidor tipo SAMBA para exportar los recursos a las diferentes plataformas que lo soporta, así también aprender a conectarse como cliente samba y obtener recursos de otro servidor vía smb.

INTRODUCCIÓN: ¿Que es SAMBA?

SAMBA es una suite de programas el cual trabaja y aloja clientes accesando archivos e impresoras vía el protocolo SMB (Session Message Block). Inicialmente escrito para UNIX, ahora SAMBA también corre en Netware, AmigaDOS, OS/2, etc.

En la practica, se puede redireccionar discos e impresoras UNIX y que sean clientes tales como WFW3.11, Windows9.x, WindowsNT, Linux y OS/2. Esto es que un cliente genérico UNIX, suple parte de la suite que aloja a usuarios y tienen una interfaz tipo ftp cuando se conecta una maquina UNIX a un servidor SAMBA, accesando a sus recursos vía el servicio smb. Esto obtienen la capacidad de este servidor trabaje mejor que un servidor Windows NT, únicamente teniendo funcionalidad y flexibilidad como está diseñado para hacerlo mas fácil a los administradores.

Los componentes de la suite son:

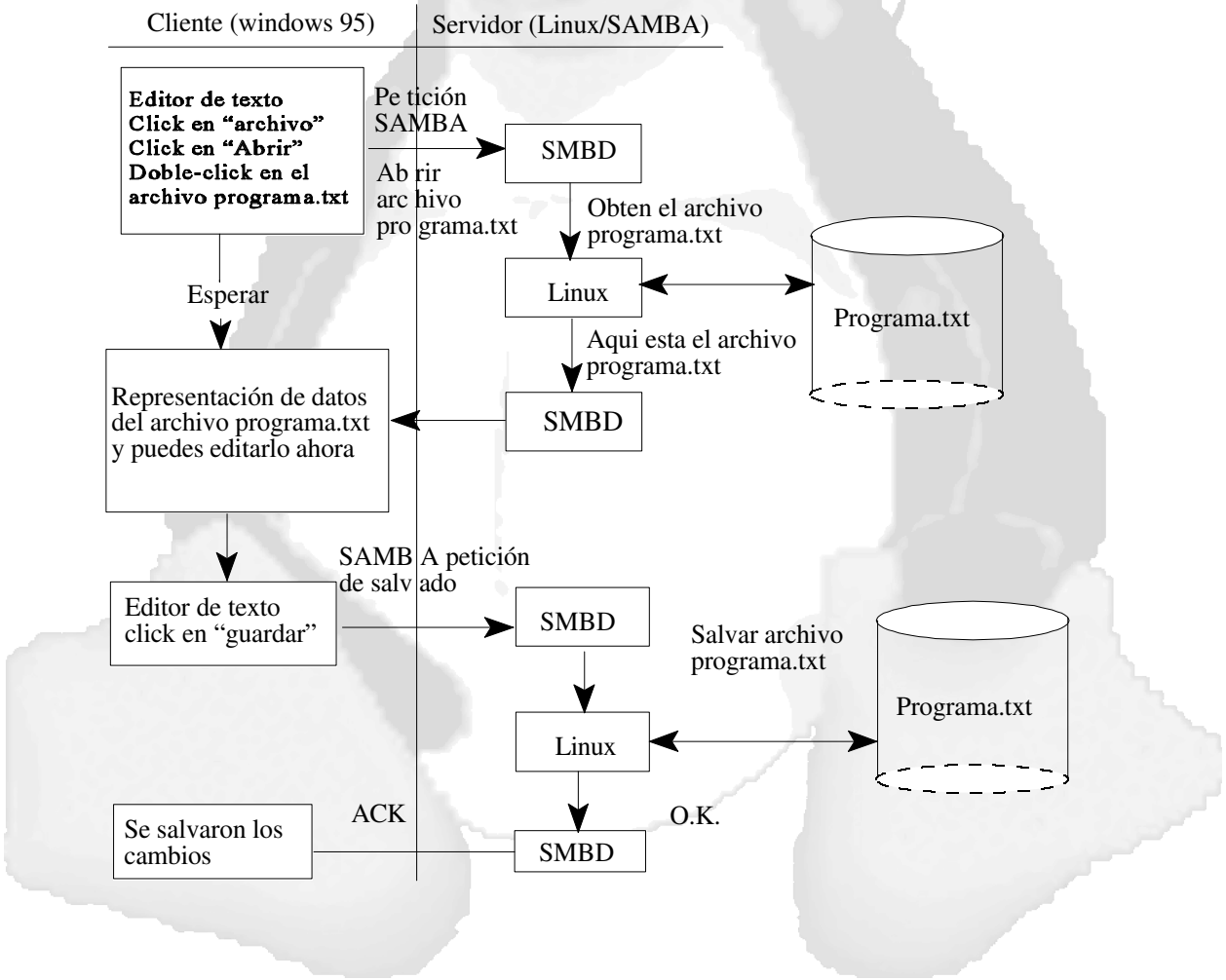
- Smbd, el servidor smb. Este actúa con las conexiones desde clientes, dando todos los archivos, permisos y directorio de trabajo al usuario.
- Nmbd es el servidor de nombre de NetBIOS, el cual ayuda a los clientes a localizar servidores, buscando y manejando dominios.
- Smbclient, el programa cliente del host-UNIX.
- Testprns un programa para hacer un test al servidor que accede a las impresoras.
- Testparms, este programa hace un test a la configuración del archivo samba y verifica que este correcto.
- Smb.conf es el archivo de configuración SAMBA.
- Smbstatus muestra lo compartido (recursos) exportado por SAMBA.

- Smbfs, es un Sistema de Archivos que trabaja con el protocolo smb. Con esto puedes montar o compartir recursos con computadoras UNIX.

- Smbmount el un comando que monta el smbfs de un sistema UNIX.

El Protocolo SAMBA

El protocolo smb es responsable de la negociación de los archivos remotos y recursos como impresoras. Si hablamos de un ejemplo mas especifico, usaremos un cliente Windows 95 y un servidor Linux, en este servidor tenemos un directorio mapeado a Windows 95 en el drive E. Este directorio tiene un archivo llamado programa.txt. En el sig. diagrama muestra el proceso que ocurre, como el cliente busca y accede al archivo desde el servidor.



Los Demonios SAMBA

El corazón de SAMBA son los programas smbd y nmbd. Usualmente corren como demonios, cual estos procesos corren todo el tiempo. RedHat corre estos demonios por default.

El proceso nmbd habilita al servidor Linux a ser explorados por otras maquinas. El demonio smbd procede los paquetes SMB y estos arriban en la red y negocia con el kernel de Linux a acceder estos recursos y compartirlos. Si un archivo especifico, el recurso es un archivo, una petición de impresión. La forma de exportar es:

Archivos	Impresoras
Linux <-->Linux	Linux<-->Linux
Windows <-->Linux	Windows <-->Linux

El archivo llamado smb.conf tiene toda la configuración e información del uso smbd y nmbd.
Smb.conf [Archivo]

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a
#
# for commentry and a ; for parts of the config file that you# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command "testparm"
# to check that you have not many any basic syntactic errors. #
#====================== Global Settings =====================================
[global]

# workgroup = NT-Domain-Name or Workgroup-Name  workgroup = WORKGROUP
# server string is the equivalent of the NT Description field
server string = Samba Server
# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
;   hosts allow = 192.168.1. 192.168.2. 127.

# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
printcap name = /etc/printcap
load printers = yes

# It should not be necessary to spell out the print system type unless
# yours is non-standard. Currently supported print systems include:
# bsd, sysv, plp, lprng, aix, hpux, qnx;
```

```
printing = bsd

# Uncomment this if you want a guest account, you must add this to /etc/passwd
# otherwise the user "nobody" is used
;  guest account = pcguest

# this tells Samba to use a separate log file for each machine
# that connects
log file = /var/log/samba/log.%m
# Put a capping on the size of the log files (in Kb).
max log size = 50

# Security mode. Most people will want user level security. See
# security_level.txt for details.
security = user
# Use password server option only with security = server
;   password server = <NT-Server-Name>

# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
;  password level = 8

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
;  encrypt passwords = yes
;  smb passwd file = /etc/smbpasswd

# Unix users can map to different SMB User names
;  username map = /etc/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
;   include = /etc/smb.conf.%m

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
socket options = TCP_NODELAY

# Configure Samba to use multiple interfaces
# If you have multiple network interfaces then you must list them
# here. See the man page for details.
;   interfaces = 192.168.12.2/24 192.168.13.2/24

# Configure remote browse list synchronisation here
# request announcement to, or browse list sync from:
# a specific host or from / to a whole subnet (see below)
;   remote browse sync = 192.168.3.25 192.168.5.255
# Cause this host to announce itself to local subnets here
;   remote announce = 192.168.1.255 192.168.2.44

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
;   local master = no
```

```
# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
;   os level = 33

# Domain Master specifies Samba to be the Domain Master Browser. This
# allows Samba to collate browse lists between subnets. Don't use this
# if you already have a Windows NT domain controller doing this job
;   domain master = yes

# Preferred Master causes Samba to force a local browser election on startup
# and gives it a slightly higher chance of winning the election
;   preferred master = yes

# Use only if you have an NT server on your network that has been
# configured at install time to be a primary domain controller.
;   domain controller = <NT-Domain-Controller-SMBName>

# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
;   domain logons = yes
# if you enable domain logons then you may want a per-machine or
# per user logon script
# run a specific logon batch file per workstation (machine)
;   logon script = %m.bat
# run a specific logon batch file per username
;   logon script = %U.bat

# Where to store roving profiles (only for Win95 and WinNT)
# %L substitutes for this servers netbios name, %U is username
# You must uncomment the [Profiles] share below
;   logon path = \\%L\Profiles\%U

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable it's WINS Server
;   wins support = yes
# WINS Server - Tells the NMBD components of Samba to be a WINS Client
#     Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
;   wins server = w.x.y.z

# WINS Proxy - Tells Samba to answer name resolution queries on
# behalf of a non WINS capable client, for this to work there must be
# at least one    WINS Server on the network. The default is NO.
;   wins proxy = yes
# DNS Proxy - tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
# this has been changed in version 1.9.18 to no.  dns proxy = no
# Case Preservation can be handy - system default is _no_
# NOTE: These can be set on a per share basis
;  preserve case = no
;  short preserve case = no

# Default case is normally upper case for all DOS files
;  default case = lower

# Be very careful with case sensitivity - it can break things!
;  case sensitive = no
```

```
#============================ Share Definitions ==============================
[homes]
comment = Home Directories
browseable = no
writable = yes

# Un-comment the following and create the netlogon directory for Domain Logons
; [netlogon]
;  comment = Network Logon Service
;  path = /home/netlogon
;  guest ok = yes
;  writable = no
;   share modes = no

# Un-comment the following to provide a specific roving profile share
# the default is to use the user's home directory;
[Profiles]
;    path = /home/profiles
;    browseable = no
;    guest ok = yes

# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no

# Set public = yes to allow user 'guest account' to print
guest ok = no
writable = no
printable = yes

# This one is useful for people to share files
;[tmp]
;   comment = Temporary file space
;   path = /tmp
;   read only = no
;   public = yes

# A publicly accessible directory, but read only, except for people in
# the "staff" group
;[public]
;   comment = Public Stuff
;   path = /home/samba
;   public = yes
;   writable = yes
;   printable = no
;   write list = @staff

# Other examples.
## A private printer, usable only by fred. Spool data will be placed in fred's
# home directory. Note that fred must have write access to the spool directory,
# wherever it is.
;[fredsprn]
;   comment = Fred's Printer
```

```
;   valid users = fred
;   path = /homes/fred
;   printer = freds_printer
;   public = no
;   writable = no
;   printable = yes

# A private directory, usable only by fred. Note that fred requires write
# access to the directory.
;[fredsdir]
;   comment = Fred's Service
;   path = /usr/somewhere/private
;   valid users = fred
;   public = no
;   writable = yes
;   printable = no

# a service which has a different directory for each machine that connects
# this allows you to tailor configurations to incoming machines. You could
# also use the %u option to tailor it by user name.
# The %m gets replaced with the machine name that is connecting.
;[pchome]
;  comment = PC Directories
;  path = /usr/pc/%m
;  public = no
;  writable = yes

# A publicly accessible directory, read/write to all users. Note that all files
# created in the directory by users will be owned by the default user, so
# any user with access can delete any other user's files. Obviously this
# directory must be writable by the default user. Another user could of course
# be specified, in which case all files would be owned by that user instead.
;[public]
;   path = /usr/somewhere/else/public
;   public = yes
;   only guest = yes
;   writable = yes
;   printable = no

# The following two entries demonstrate how to share a directory so that two
# users can place files there that will be owned by the specific users. In this
# setup, the directory should be writable by both users and should have the
# sticky bit set on it to prevent abuse. Obviously this could be extended to
# as many users as required.
;[myshare]
;   comment = Mary's and Fred's stuff
;   path = /usr/somewhere/shared
;   valid users = mary fred
;   public = no
;   writable = yes
;   printable = no
;   create mask = 0765
```

El archivo smb.conf tiene una estructura simple como se lista:

- El archivo esta dividido en secciones. Cada sección contiene parámetros, estos define que comparte SAMBA para exporta.

- Una sección global define los parámetros que tienen control general de las características de SAMBA.

- otra sección que comienzan con un nombre encerrado entre corchetes (ejemplo [home]) y continua hasta la sig. sección

- Especifica un parámetro con la siguiente sintaxis

```
Nombre = valor
```

El nombre puede estar hecho de una o mas palabras separadas por espacio. El valor es booleano (falos o verdadero, si o no, 1 o 0); numérico o una cadena de caracteres.

- Los comentarios son precedidos de punto y coma (;) y debe poner en cada línea.

- Las líneas pueden ser continuadas con un slash (\ ) y luego un <enter> y con ello podrá continuar la línea.

Este archivo tiene 3 grandes partes:

- Parámetros globales.
- La sección del directorio compartido.
- La sección de las impresoras compartidas.

Los parámetros globales, puede compartir, ponen las normas de tu sistema. La sección [home] y [printer] son instancias especiales de servicio. El termino servicios en la terminología de SAMBA de los directorios e impresoras son compartidas o exportar como clientes de la red. La configuración SAMBA usa un nivel de seguridad de usuario y está especificado en el /etc/smb.conf por el parámetro `security = user`. Esta seguridad es requerida, ya que el login y contraseña de Windows es muy fácil de suplantar.

DESARROLLO

Veremos la configuración del Servidor SAMBA, exportar un directorio público exclusivo por SAMBA.

1) Entra a la computadora e inicia sesión con Linux.

2) Haz una copia de respaldo.

```
#cp /etc/smb.conf /etc/smb.comf.orig
```

Editamos el archivo /etc/smb.conf y nos colocamos en la sección [public]:

```
;[public]
;   comment = Public Stuff
;   path = /home/samba
;   public = yes
;   writable = yes
;   printable = no
```

Removemos los comentarios y luego salvamos el archivo y debe quedar como sigue:

```
[public]
   comment = Public Stuff
   path = /home/samba
   public = yes
   writable = yes
   printable = no
```

Ahora que tenemos las modificaciones del archivo, pero necesitamos que los servicios vean estos cambios ya se leyendo los anteriores y los tiene cargados usados por el puerto 139. Hay varias formas de que los servicios lean la nueva configuración.
i)Reiniciar la maquina (no recomendable, pero inicialmente la usan mucho).

ii)Usando el comando kill

iii)Usando el script de los servicios para SAMBA.

3)Ahora usaremos el comando `kill`:

```
#kill -HUP `ps ax | grep smbd | awk ´{print $1}´`
#kill -HUP `ps ax | grep nmbd | awk ´{print $1}´`
```

Con ello haremos que el demonio vuelva a leer el archivo /etc/smb.conf y lee los cambios.

4) Usando el script de SAMBA ubicado en la ruta /etc/rc.d/init.d para detener, reiniciando los servicios SAMBA y lee las nuevas modificaciones del archivo /etc/smb.conf.

```
#/etc/rc.d/init.d/smb stop
#/etc/rc.d/init.d/smb start
```

Exportar tu directorio de casa del servidor Linux y puedes verlo en Windows 95.

5) Editar el archivo /etc/smb.conf y localizar la sección [homes]

```
[homes]
comment = Home Directories
browseable = no
writable = yes
```

Con ello podemos darnos cuenta que exportaremos los homes de los usuarios. Reiniciar los servicios si hubo cambios:

```
#/etc/rc.d/init.d/smb stop
#/etc/rc.d/init.d/smb start
```

Modificar el archivo /etc/smb.conf para exportar el CDROM y puedan accesar las maquinas clientes (Windows ó Linux).

6)Editar el archivo /etc/smb.conf y agregar las nuevas líneas para exportar el CDROM (formato de solo lectura).

```
[cdrom]
comment = Espacio temporal dado al CDROM
path = /mnt/cdrom
read only = yes
public = yes
```

Después de salvar los cambios, reiniciar los servicios y además de montar el CDROM en formato de lectura:

```
#mount -r -t iso9660 /dev/cdrom /mnt/cdrom
```

7) Exportando un directorio de 2 o mas usuarios. Aqui veremos que hay una forma de agregar 2 ó mas usuarios con el parametro “valid users”, aquí veremos un ejemplo que compartan el mismo directorio con 3 usuarios que esten registrados en el sistema (ejemplo: falcon, kyo y mena).

8)Editemos el archivo /etc/smb.conf

Agregemos estas líneas:

```
[compartir]
comment = Dir. compartido para el staff
path = /home/staff
valid users = falcon kyo mena
writable = yes
public = no
printable = no
create mask = 0765
```

salvamos y tendremos directorio común para esta 3 personas. Reiniciamos servicios de SAMBA.

9)A veces podemos equivocarnos en escribir una mal sintaxis en el archivo /etc/smb.conf, donde una forma de verificarlo es editarlo y buscar error en el archivo (algo tedioso), pero hay un comando que

hace esto por nosotros llamado testparms.Usa el sig. comando:

```
#testparms

Observa y anota el resultado
```

10)También necesitamos saber que usuarios y de donde se están conectando y tenemos el sig. comando:

```
#smbstatus                    (Usarlo cunado hay conexión con clientes)

Observa el resultado y anotalo
```

El archivo /etc/smb.conf contiene macros para la versatilidad y flexibilidad del mismo. SAMBA puede sustituir macros de los parametros de servicios. Para diseñar un macro es crear un simbolo del porcentaje (%) como el primer caracter de alguno de los nombres predefinidos. Mostraremos las macros:

MACRO	DESCRIPCIÓN
%S	Servicio común o compartido si existe (si existe alguno)
%P	Directorio de root del servicio común o compartido (si existe alguno)
%u	Nombre del usuario del servicio común o compartido (si existe alguno)
%g	Grupo primario del nombre de %u
%U	Sesión del usuario de la respuesta, pero no necesariamente la primera respuesta
%G	Grupo primario del nombre del %U
%H	El directorio de casa del usuario obtenido por %u
%v	Numero de versión de SAMBA
%h	Nombre del host del servidor de SAMBA
%m	Nombre NetBIOS de la computadora cliente
%L	Nombre NetBIOS del servidor SAMBA
%M	Nombre de Internet de la computadora cliente
%d	Proceso ID del servidor común de procesos
%a	Arquitectura del servidor remoto
%I	Dirección IP de la computadora cliente
%T	Dato y tiempo común

11) Ahora si tenemos una maquina con plataforma Windows (3.11, 9x, ó NT) cliente, ademas del protocolo NetBeui activado, usaremos las propiedades de Windows para buscar el dominio correspondiente a la maquina Linux y poder accesar a los recursos compartidos (ya que veremos 4 servicios activados: home directory, impresoras, cdrom y temporal).

Haz un resumen de que cliente utilizaste (preferentemente Windows) y como haces la conexión para iniciar sus recursos en el servidor SAMBA. Pero hacemos una pregunta.
¿Si tengo una maquina Linux (cliente) puedo conectarme al servidor SAMBA (con Linux) vía el protocolo SAMBA?

La respuesta es sí

Conexión del cliente (Linux) con el servidor vía SAMBA

Queremos ver la información que exportamos, en vez de un cliente Windows ahora un cliente Linux, para esto usaremos el comando smbclient que usa una breve sintaxis:

```
#smbclient [opciones] servidor
```

```
Ejemplo:

#smbclient –L nombre_del_servidor
```

Anota el resultado

El comando smbclient tiene las sig. banderas:

-p port	Conecta al puerto especifico
-d debuglevel	El nivel de debughlevel y este tiene niveles del 0 al 5
-l logbasename	Nombrebase de los archivos logdebug
-n nebios name	Usa este nombre como mi nombre de Netbios

-P	Conecta el servicio como una impresora
-M host	Manda un mensaje winpopup al host
-L host	Obtiene una lista de objetos compartidos disponible en un host
-I dest IP	Usa esta IP para conectarse
-U username	Pone el nuevo username en la maquina destino
-W workgroup	Pone el nuevo workgroup en la maquina destino
-t codigo de terminal	termina la E/S en codigo
-T <c | x>IXgbnA	Comando en línea de tar
-D directory	Inicia desde el directorio

Queremos ver el directorio temporal que estoy exportando (public) y es algo similar a la conexión tipo Windows que utiliza el protocolo Netbios:

\\servidor\recurso_compartido

13) Esta conexíon será:

```
#smbclient \\\\servidor\\recurso_compartido
```

Se pone otro (\) en Linux ya que este simbolo es un caracter especial y necesitamos que lo lea como un caracter asi que tenemos que protegerlo (y para esto el caracter especial es el “\”).

```
#smbclient \\\\servidor\\public
```

Despues te preguntará el passwd y te dará un prompt de este tipo:

```
smb\>
```

Podemos usar comandos mixtos tanto los de MS-DOS (dir, copy, md, etc.) como Linux (ls, cp, mkdir, etc.), también podemos hacer transferencia de archivos como el ftp (mencionado en la introducción) y otros comandos mixtos, ya que nos podemos conectar bajo el protocolo netbios a maquinas Windows y UNIX.

Otra utilería importante es la forma de montaje vía SAMBA con los directorios compartidos y la sintaxis es la siguiente:

```
smbmount //servidor/servicio /punto de montaje [opciones]
```

si usamos el comando smbmount -h veremos unas opciones como:

-p puerto	Te conectas al puerto
-s nombre_del_servidor	Nombre del servidor NetBios
-c nombre_del_cliente	Nombre del cliente NetBios
-I nombre de la maquina	El hostname de la maquina
-U username	El nombre del usuario
-P password	Usa este password
-h	Imprime ayuda

```
#smbmount //cerbero/public /mnt –c cliente
```

Si no marca error por el password o porque no esta compartido entonces veremos como resultado la siguiente salida usando el comando mount:

```
/dev/hda3 on / type ext2 (rw,usrquota)
none on /proc type proc (rw)
/dev/hdb on /mnt/cdrom type iso9660 (ro)
//cerbero/public on /mnt type smbfs (0)
```

14)O también con el comando smbstatus:

```
#smbstatus
```

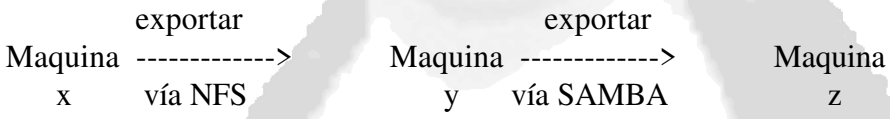
```
Anota el resultado
```


15) Para desmontar directorios vía SAMBA se usará el comando smbumount:

```
#smbumount /punto_de_montaje
```

CUESTIONARIO:

1.- ¿Que es SAMBA?
2.- ¿Que protocolo utiliza SAMBA para transferir los archivos e impresoras?
3.- ¿Cuales plataformas soportan el protocolo NetBIOS?
4.- Con ayuda de diagramas cliente-servidor cuales son las opciones de transferir archivos y compartir impresoras en plataforma Windows y Linux.
5.- ¿Cuales y cuantos comandos son del paquete SAMBA y cuales usamos?
6.- ¿Un servidor SAMBA (Linux) puede ser también cliente? Si ó no ¿Porque?
7.-¿Que puerto maneja el protocolo NetBIOS?
8.- Si tenemos directorio compartido de una maquina "x" a nuestro servidor "y" vía NFS, podemos exportarlo vía SAMBA a la maquina "z"



¿Se puede hacer esta transferencia? Si ó no ¿porque?

CONCLUSIONES