PRACTICA No. 13

"CONFIGURACIÓN DE NFS"

OBJETIVO: Conocer el tipo de transmisión que ofrece NFS, como el protocolo y la configuración para la exportación del volúmenes y poder compartirlo.

INTRODUCCIÓN:

NFS (Networking File System o Sistemas de Archivo en Red) fue desarrollado y llevado a la mercadotecnia por Sun MicroSystems Inc. en 1984. Sun tiene la filosofía de distribución y sistemas abiertos. Así, desde el comienzo NFS fue diseñado para permitir la conexión de computadoras de diferentes sistemas operativos corriendo en ellas en redes homogéneas. Sin embargo, la realidad de NFS pudo ser usado en redes heterogéneas y fue probablemente el gran avance sobre otros productos como el Remote File System de AT&T (RFS) o el Andrew File System (AFS).

El NFS es el servicio mas complejo de los que ofrecen usando RPC. Permite acceder a los archivos remotos exactamente igual que si fueran locales. NFS ofrece numerosas ventajas:

- Los datos accedidos por todo tipo de usuarios pueden mantenerse en un nodo central, con clientes que montan los directorios en el momento de arrancar, esto es transparente a los usuarios. Si además se instala NIS, los usuarios podrían entrar y trabajar de forma transparente en cualquiera de las máquinas.

- Los datos que consumen grandes cantidades de espacio en disco pueden mantenerse en un solo nodo.

- Los datos de administración también puede mantenerse en un solo nodo. No será necesario usar el rcp para instalar el mismo archivo.

- Un aspecto de la transparencia es la velocidad de acceso de datos sobre la red, cual debe ser alta no es notificable a diferencia del acceso a disco local. La meta original del desarrollo de NFS es el 80% de los datos con respecto al disco local.

El NFS de Linux principalmente fue obra de Rick Sladkey apoyandose en programas de Mark Shand. Donde la implementación del cliente será en el nuevo VFS (Virtual File Systems) y no necesita del código biod . NFS esta basado en un protocolo acordado cual largo corresponde al acordado definido con el modelo OSI:

Aplicación		mount, NFS, yp, bind
Presentación		XDR
Sesión		RPC
Transporte		TCP, UDP
Red		IP
Liga		ETHERNET
Físico		

Llamada de Procedimiento Remoto

La llamada de procedimiento remoto (RPC) es la capa de sesión de protocolo. La ejecución de un RPC consiste en los sig. pasos:

1) Activación por el programa cliente. Los parámetros de solicitud son empaquetados dentro de un paquete de datos.

2)Mandando solicitudes y desempaquetamiento de los parámetros en el programa del servidor.

3)Ejecución de la solicitud (el procedimiento) en el servidor.

4)Empaquetar y retornar de los resultados del cliente.

5)Desempaquetar los resultados por el cliente y continuación de los programas normales de ejecución.

Port Mapper

El servicio portmapper opera semejante a un servicio, cual esencialmente controla una tabla de mapeo de relación de programas RPC and/or números de versiones locales de TCP ó UDP de números de puertos, como cual es un servicio y disponibilidad. EL portmapper ocupa el puerto 111 en ambos TCP y UDP.

Si este principio definido al número de puerto cual tiene que ser usado por el programa RPC.

Protocolo de cabecera RPC

El protocolo RPC es el chasis de transporte solicitado al servidor, cuando el flujo esta dentro del procedimiento. La programación puede definir un número arbitrario de procedimiento entre el cliente y servidor, con arbitraria opciones de parámetro y resultados. Un funcionamiento de grupo coherente de procedimiento es llamado un servicio de RPC. Cada servicio es asignado un llamado del numero de programa.

Protocolo PortMapper

Cliente y servidor ambos comunican con el portmapper vía RPC. Los procedimientos RPC define en el protocolo portmapper. No es apropiado obtener la descripción exacta de los procedimientos y el resultado de este punto.

Protocolo de NFS

El protocolo NFS consiste de un número de procedimientos RPC, construye en el mismo principio como el protocolo portmapper. La especificación del número de puerto de 2049. Cuando alguien accede a un archivo remoto, el núcleo manda una llamada RPC al programa nfsd (demonio de NFS) del nodo remoto. Esta llamada incluye el descriptor de archivo, el nombre del archivo a acceder y los identificadores de usuario y de grupo del demandante. Estos identificadores se usan para checar los permisos de acceso en la maquina remota, con lo que los usuarios de ambas maquinas deberían ser los mismos.

En varias implementaciones de UNIX, las funcionalidades del clientes y servidor NFS, se implementan como demonios de nivel del núcleo que se arrancan desde el espacio de usuario al arrancar la maquina. Se trata del programa biod (Block I/O Daemon) en el cliente. En cuanto a Linux el programa biod no se utiliza porque este se implementa en el VFS.

Demonios de NFS

Si se desea proporcionar un servicio NFS a otras maquinas, deberá ejecutar en el servidor los programas nfsd y mountd. Son programas basados en RPC, por los que no son arrancados por el demonios inetd, sino lanzados como demonios de tiempo de arranque y registrados en el mapeador de puertos de RPC.

DESARROLLO:

Para esta practica necesitamos 2 maquinas Linux para hace la interpretación de cliente-servidor, por esto necesitaremos las direcciones IP correspondientes, si no las sabes usa el comando ifconfig y correlo tanto en el cliente como en el servidor.

1) Probaremos si el cliente y el servidor tenga el filesystem de NFS en el kernel, esto se ve a través del sig. archivo:

```
#cat /proc/filesystems
minix
ext2
msdos
nodev proc
nodev nfs
```

Si no tenemos la palabra de nfs entonces recompilaremos el kernel.

Configuración del Servidor

Aquí haremos modificaciones hacia un archivo y manipular 2 archivos ejecutables. Para poder compartir directorios en red, necesitamos decirles a los demonios de NFS cuales queremos y estos demonios leen el archivo /etc/exports.

2) El archivo exports tiene el sig. aspecto:

```
/directorio    nombre_de_la_maquina_ó_dir._IP(opciones)

Ejemplo:
#cat /etc/exports
/home     maquina1(rw) 192.168.2.10(rw) 192.168.2.54(rw)
/tmp      192.168.2.150(ro)
/usr/tmp  (ro)
```

Cada línea especifica un directorio, la lista de la(s) maquina(s) que pueden acceder a el por medio de NFS. En la lista de la maquina puede ser con el nombre completo de la maquina o el numero IP. Aunque se pueden usar los comodines (* y ?) que tiene el shell. Si no lleva el nombre de la maquina (como en el caso de /usr/tmp) entonces se exportará a todas las maquinas de Internet.

Tras el nombre de la maquina autorizada, se encerrará entre paréntesis el conjunto de opciones separadas por comas. Dichas opciones son:

insecure	Permitir acceso no autentificado desde ese nodo.
unix-rpc	Requerir autentificación RPC del dominio UNIX para este nodo. Se trata simplemente de que las peticiones se originen en un puerto reservado (es decir inferior a 1024). Esta opción está activa por defecto.
kerberos	Requerir autentificación Kerberos. Tampoco se ha implementado aun.
root_squash	Se trata de una opción de seguridad que niega acceso a archivos del superusuario a las peticiones con uid 0 o 65535 (este último se le asocia al usuario nobody).
no_root_squash	Evita la restricción anterior.
ro	Monta la jerarquía de archivos en solo lectura.
rw	Monta la jerarquía con permisos para leer y escribir.
link_relative	Convierte enlaces simbólicos absolutos (que comienza con un barra de directorio, "/") en enlaces relativos colocados los prefijos ../ que sean necesarios para hacer que apunten a la raíz del servidor. Esta opción solo tiene sentido cuando se monta un sistemas de archivos completo y no solo un directorio.
link_absolute	Deja los enlaces absolutos como estaban.
map_identity	Esta opción indica al servidor que asuma que el cliente utiliza el mismo mapa de uids y gids que el servidor.
map_daemon	Esta opción indica al servidor NFS que no comparte el mapa de usuarios con el del cliente. Con ello, las comparaciones de uids y gids se harán mediante una lista de mapeado entre ambos que se construirá llamado al demonio ugidd del cliente.

3)Ahora modificamos ese archivo para que tenga el siguiente formato:

```
#vi /etc/exports

/tmp            direccion_IP_cliente(rw,no_root_squash)
```

```
/mnt/floppy    direccion_IP_cliente(rw)
```

4)Salvamos el archivo /etc/exports y necesitamos que los demonios tenga esta configuración en el archivo lo cual usaremos los scripts:

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

Si no marca ningún error hemos acabado con la configuración del servidor y ahora continuaremos con el cliente, pero antes necesitaremos un floppy para montarlo en el directorio /mnt/floppy del servidor:

```
#mount /dev/fd0 /mnt/floopy
```

Configuración del cliente

Ahora estamos en el cliente y montaremos el directorio tmp que se encuentra en el servidor y crearemos un directorio que sea la IP del servidor:

```
#mkdir /direccion_IP_del_servidor
#mkdir /direccion_IP_del_servidor/tmp
#mkdir /direccion_IP_del_servidor/floppy
#mount –t nfs nombre_servidor:/tmp   /direccion_IP_del_servidor/tmp
```

si no marca error el mountaje lo podemos ver con mount:

```
#mount
```

Anota el resultado

Si queremos que este directorio se monte cada vez que el cliente inicie lo podemos poner las sig. lineas en el archivo etc/fstab. Un ejemplo de este archivo es:

/dev/hda1 / ext2 defaults 1 1

/dev/hda2	swap	swap	defaults	0 0
servidor:/tmp	/direccion_IP_servidor	nfs	defaults	1 1

Y con ello logramos cada vez que inicie la computadora cliente buscará ese directorio del servidor y cuando lo encuentre lo montará en su directorio, el comando mount tiene opciones para montar vía NFS que son las siguientes:

rsize=n y wsize=n	Especifican el tamaño del datagrama utilizado por el cliente NFS en la peticiones de lectura y escritura, respectivamente. Por defecto, cada una de ellas vale 1024 bytes, dados los limites del tamaño del datagrama UDP ya comentados.
timeo=n	Esta opción establece el tiempo máximo de espera de respuesta a una petición del cliente NFS; en centésimas de segundo. Por defecto, este valor es de 0.7 segundos.
hard	Marca el montaje del volumen como físico. Es un valor por defecto.
soft	Hace que el montaje sea solo lógico (opuesto al anterior).
intr	Esta opción habilita la posibilidad de que una señal interrumpa una espera por NFS. Es útil para poder abortarla cuando el servidor no responde.

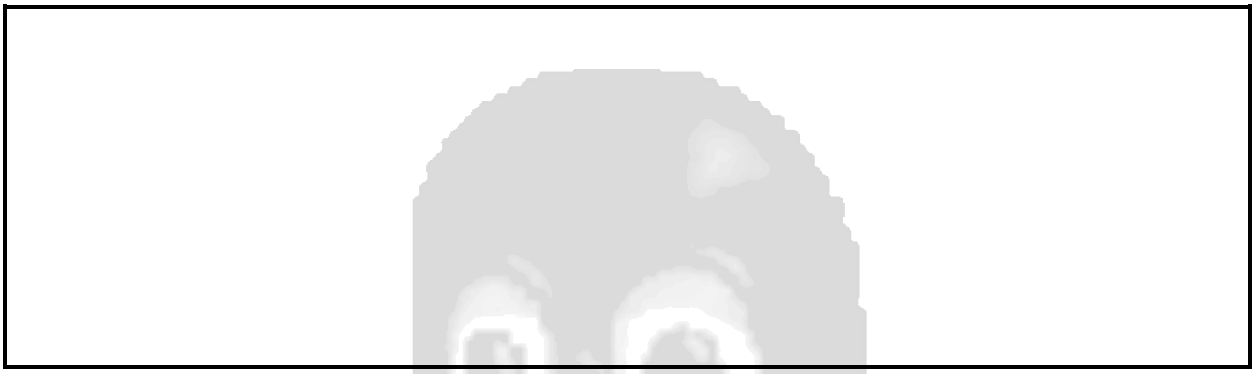
Ahora montaremos el floopy que se encuentra montado en el servidor en el directorio /mnt/floppy al cliente:

```
#mount servidor:/mnt/floppy /direccion_IP_del_servidor/floppy
```

Con ello obtenemos los datos de los directorios que queremos del servidor. Ahora para ver que maquinas han montado nuestro directorio (el caso del servidor), tenemos un comando llamado:

```
#showmount –a
```

Apunta el resultado y comentalo



Y con ello mostramos una forma de exportar y compartir archivos en una red bajo NFS.

CUESTIONARIO:

1.- ¿ Que es el NFS?
2.- ¿Que ventaja tiene el NFS?
3.- ¿Que desventaja tiene el NFS?
4.- ¿El servidor NFS también puede ser cliente? si ó no ¿Porque?
5.- En que aplicaciones y protocolos está basado NFS y describa cada una de ellas
6.- Si tenemos una maquina “X” que exporta vía NFS el directorio “direc1" a la maquina “Y” y este lo monta en el directorio “/X/direc2" y esta maquina “Y” exporta el directorio “/X/direc2" a otra maquina “Z” y este lo monta en el directorio “/X/Y/monta”. ¿Se puede hacer esta operación? Si ó no ¿Porque?
7.- ¿Cuales son los productos que tienen similitud con NFS? Hable brevemente de ellos
8.- ¿Que es el XDR (eXternal Data Representaion)?
9.- ¿En que puerto están los demonios de NFS?
10.- ¿Que comandos pertenecen a NFS y cuales usamos?

CONCLUSIONES: