

UNIVERSIDAD AUTÓNOMA DE YUCATÁN

FACULTAD DE MATEMÁTICAS

ANEXO DE TESIS DE ALEX ANTONIO TURRIZA SUÁREZ

Configuración de un Sistema de Archivos en Red [NFS] entre una PC x86-64 y una BeagleBone Black

Autor:

Alex Antonio TURRIZA SUÁREZ

10 de marzo de 2017

Índice

1. Introducción	2
2. Definición de NFS	2
3. Descarga e instalación	2
3.1. Instalación en host / PC	2
3.1.1. Seguridad	5
3.2. Instalación en cliente / BeagleBone	7
4. Ejecución	8

1. Introducción

Cuando dos máquinas de diferente arquitectura deben trabajar en un sólo proyecto, suele suceder que es mucho más cómodo realizar código y documentación en una, a pesar de que los archivos estén destinados a ser usados en la otra.

Para ello, se mostrará la forma de configurar un sistema de archivos en red NFS que facilite la tarea de compartir archivos en un directorio.

En este trabajo se mostrará la instalación del sistema en una máquina host en una PC y un cliente en una BeagleBone Black, aprovechando que al conectar mediante USB, se crea una red entre ambas plataformas.

2. Definición de NFS

Sistema de archivos en red (NFS, "Network File System" por sus siglas en inglés), es un protocolo que permite acceder mediante una conexión remota a un sistema de archivos [El Manual del Administrador de Debian¹, consultado en Octubre 2016].

En su funcionamiento, permite que un equipo host comparta determinado directorio con otros equipos clientes, pudiendo determinar qué equipos tienen permisos de lectura, escritura o ambas.

3. Descarga e instalación

3.1. Instalación en host / PC

Bajo Ubuntu en sus últimas versiones en el momento de la redacción de éste documento, se abre una terminal con los comandos *Ctrl + Alt + t*.

Lo primero, es actualizar los repositorios con:

```
$ sudo apt-get update
```

Una vez actualizados, se procede a la instalación de un paquete mediante el siguiente comando:

¹<https://debian-handbook.info/browse/es-ES/stable/sect.nfs-file-server.html>

```
$ sudo apt-get install nfs-kernel-server
```

Al finalizar la descarga e instalación, se debe modificar un archivo. Copiar en la terminal el siguiente comando y colocar la contraseña:

```
$ sudo nano /etc/default/nfs-kernel-server
```

Se debe modificar la línea **NEED_SVCSSD=** y colocar "no" en el entrecorillado, como muestra la figura 1. Cuando se termine de modificar, guardar con la combinación de teclas *Ctrl* + *O* y regresar a la terminal con *Ctrl* + *X*.



```
alexrt07@HP-Omen15-GTX: ~
GNU nano 2.5.3 Archivo: /etc/default/nfs-kernel-server

# Number of servers to start up
RPCNFSDCOUNT=8

# Runtime priority of server (see nice(1))
RPCNFSDPRIORITY=0

# Options for rpc.mountd.
# If you have a port-based firewall, you might want to set up
# a fixed port here using the --port option. For more information,
# see rpc.mountd(8) or http://wiki.debian.org/SecuringNFS
# To disable NFSv4 on the server, specify '--no-nfs-version 4' here
RPCMOUNTDOPTS="--manage-gids"

# Do you want to start the svcgssd daemon? It is only required for Kerberos
# exports. Valid alternatives are "yes" and "no"; the default is "no".
NEED_SVCSSD=no

# Options for rpc.svcgssd.
RPCSVCGSSDOPTS=""

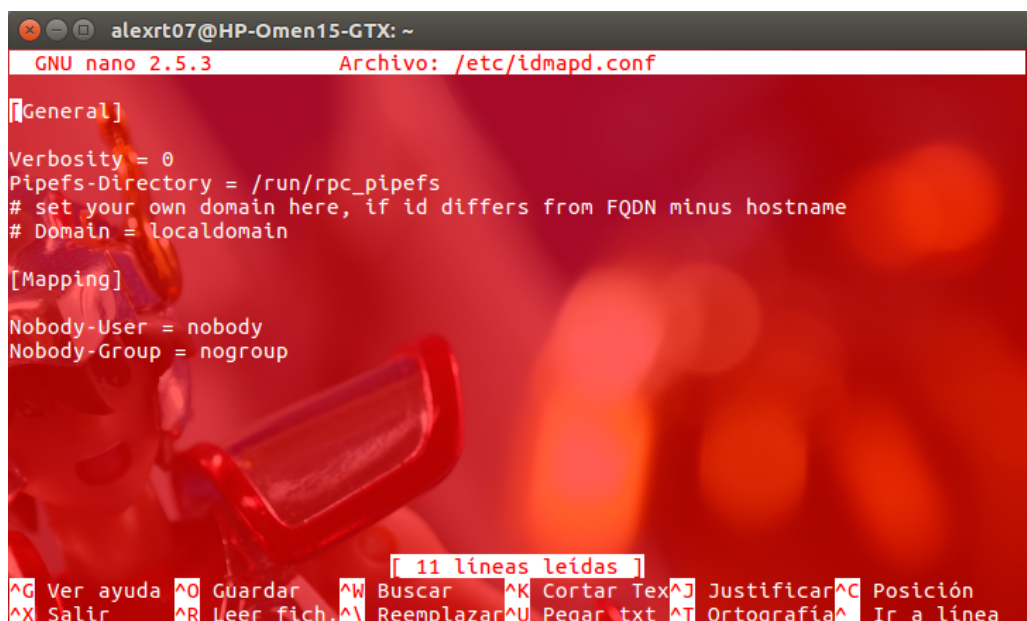
[ 22 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Figura 1: Archivo `/etc/default/nfs-kernel-server` ya modificado.

Lo siguiente es abrir el archivo ubicado en `/etc/idmapd.conf`:

```
$ sudo nano /etc/idmapd.conf
```

Verificar que existan las líneas *Nobody* – *User* = *nobody* y *Nobody* – *Group* = *nogroup* como muestra la figura 2.



```
alexrt07@HP-Omen15-GTX: ~
GNU nano 2.5.3 Archivo: /etc/idmapd.conf

[General]
Verbose = 0
Pipefs-Directory = /run/rpc_pipefs
# set your own domain here, if id differs from FQDN minus hostname
# Domain = localdomain

[Mapping]
Nobody-User = nobody
Nobody-Group = nogroup

11 líneas leídas
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Figura 2: Archivo /etc/idmapd.conf.

Cuando se realiza una conexión con la BeagleBone Black mediante un cable USB, se crea una red con las siguientes direcciones: 192,168,7,2 para la BeagleBone y 192,168,7,1 para el PC host. Entonces, tomando en cuenta lo anterior, se modifica el archivo /etc/exports de la siguiente manera:

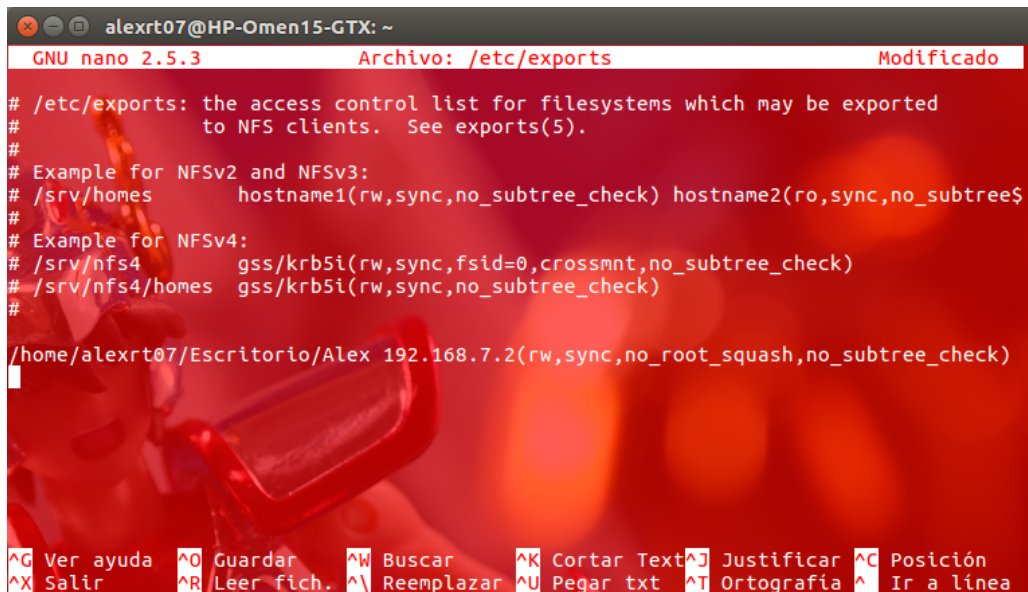
Se abre el archivo con nano, en la terminal:

```
$ sudo nano /etc/exports
```

En el archivo que se abre, añadir la siguiente línea (note que dentro del paréntesis, entre los comandos no existen espacios):

```
/home/alexrt07/Escritorio/Alex 192.168.7.2(rw, sync ,
no_root_squash , no_subtree_check)
```

Donde /home/alexrt07/Escritorio/Alex es el directorio a compartir y 192.168.7.2 es la dirección ip de la BeagleBone. Así, el archivo queda como muestra la figura 3.

A screenshot of a terminal window on a Linux system. The window title is 'alexrt07@HP-Omen15-GTX: ~'. The terminal shows the GNU nano 2.5.3 text editor editing the file /etc/exports. The file content includes comments about NFS export access control, examples for NFSv2, NFSv3, and NFSv4, and a specific export line for /home/alexrt07/Escritorio/Alex. The bottom of the screen shows a status bar with various keyboard shortcuts like '^G Ver ayuda', '^O Guardar', '^W Buscar', etc.

```
alexrt07@HP-Omen15-GTX: ~
GNU nano 2.5.3 Archivo: /etc/exports Modificado

# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree$
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home/alexrt07/Escritorio/Alex 192.168.7.2(rw,sync,no_root_squash,no_subtree_check)

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Figura 3: Archivo /etc/exports.

Finalmente, resta reiniciar el servidor con el siguiente comando:

```
$ /etc/init.d/nfs-kernel-server restart
```

Se deberá mostrar una confirmación de reinicio exitoso.

3.1.1. Seguridad

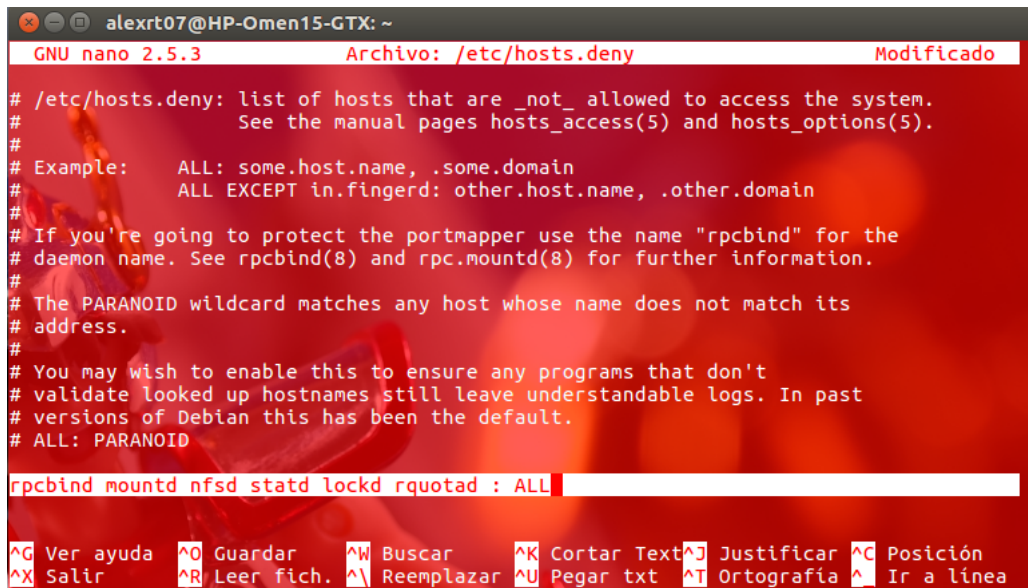
Para evitar dejar hoyos de seguridad de acceso a los archivos personales, es altamente recomendable modificar los archivos /etc/hosts.deny y /etc/hosts.allow para permitir acceso solamente a los clientes conocidos.

Abrir el archivo /etc/hosts.deny con:

```
$ sudo nano /etc/hosts.deny
```

Y añadir la siguiente línea:

```
rpcbind mountd nfsd statd lockd rquotad : ALL
```



```
alexrt07@HP-Omen15-GTX: ~
GNU nano 2.5.3 Archivo: /etc/hosts.deny Modificado

# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:  ALL: some.host.name, .some.domain
#          ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: PARANOID

rpcbind mountd nfsd statd lockd rquotad : ALL

^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar Text ^J Justificar  ^C Posición
^X Salir      ^R Leer fich.^_ Reemplazar  ^U Pegar txt   ^T Ortografía ^_ Ir a línea
```

Figura 4: Archivo /etc/hosts.deny

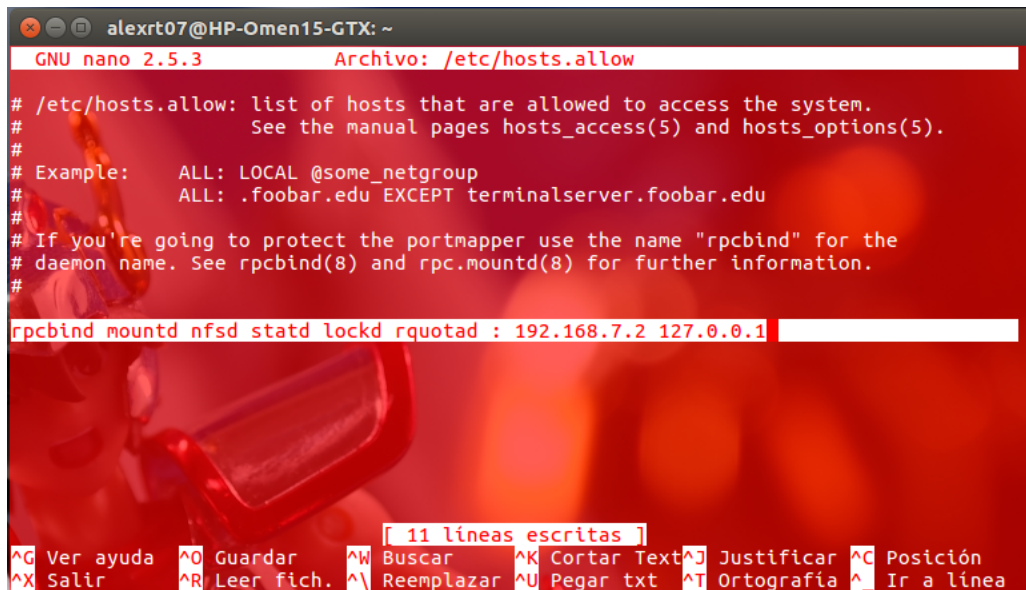
Como muestra la figura 4. Ahora, abrir el archivo /etc/hosts.allow con el comando:

```
$ sudo nano /etc/hosts.allow
```

Y añadir la siguiente línea:

```
rpcbind mountd nfsd statd lockd
rquotad : 192.168.7.2 127.0.0.1
```

Como muestra la figura 5.



```
alexrt07@HP-Omen15-GTX: ~
GNU nano 2.5.3 Archivo: /etc/hosts.allow

# /etc/hosts.allow: list of hosts that are allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:  ALL: LOCAL @some_netgroup
#          ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
rpcbind mountd nfsd statd lockd rquotad : 192.168.7.2 127.0.0.1

[ 11 líneas escritas ]
^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar Text ^J Justificar  ^C Posición
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar txt   ^T Ortografía ^_ Ir a línea
```

Figura 5: Archivo `/etc/hosts.allow`

Reiniciar el servidor con:

```
$ service nfs-kernel-server restart
```

Ahora, el PC está preparado para compartir vía red el directorio `/home/alexrt07/Escritorio/Alex/`

3.2. Instalación en cliente / BeagleBone

Asumiendo que la BeagleBone Black tiene un Debian con su archivo `/etc/apt/sources.list` correctamente configurado, ejecutamos en la terminal de nuestro host para conectarnos:

```
$ ssh -l root 192.168.7.2
```

donde `ssh` es el comando para conectarse por el protocolo secure shell, `-l` es el comando que indica que se hará un login con el usuario `root`, y `192.168.7.2` es la dirección IP de la BeagleBone en la red que se creó a través del cable USB.

Entonces, una vez hecho el login, ejecutar:


```
$ apt-get install nfs-common
```

Que instalará y preconfigurará los archivos necesarios para una correcta comunicación a través de NFS.

Es recomendable crear un directorio en donde se montarán los archivos que compartirá con la PC:

```
$ mkdir /home/debian/Alex_tesista
```

4. Ejecución

Se procede a montar el sistema de archivos con el siguiente comando:

```
$ mount -t nfs -o proto=tcp,port=2049  
192.168.7.1:/home/alexrt07/ARM-Root/home/Alex  
/home/debian/Alex_tesista/
```

En donde *mount* es el comando para montar el directorio, *-t nfs* indica que se trata de un sistema de archivos por red, *-o proto=tcp,port=2049* indica que se utilizará el protocolo de transferencia de archivos a través del puerto 2049 (mirar el manual de nfs en su página 5 con **\$ man 5 nfs** para más opciones e información), *192.168.7.1*: es la dirección IP de la PC host, */home/alexrt07/ARM-Root/home/Alex* es el directorio que contiene los archivos a compartir, y */home/debian/Alex_tesista/* es el directorio creado en donde se encontrarán los archivos.

Para cerrar esta conexión, utilice

```
$ umount /home/debian/Alex_tesista/
```

Tome en cuenta que al finalizar la conexión, no se mantendrán los archivos compartidos por nfs. Desaparecerán y contendrá los archivos originales que esa carpeta contenía antes de montar el sistema de archivos por red.