

NFS Guide

by Brian Fraser

Last update: Jan 19, 2021

This document guides the user through

1. Setting up an NFS server.
2. Connecting to the NFS server from the target.

Table of Contents

1. NFS Server Setup.....	2
2. Mounting an NFS Drive on the Target.....	3

Note: This guide has not yet been tested in the SFU Surrey Linux Lab. Some changes may be needed. Remember that you cannot execute any commands as root (using `sudo`) in the host OS in the Linux lab!

Formatting

1. Commands for the host Linux's console are show as:
`(host)$ echo "Hello PC world!"`
2. Commands for the target (BeagleBone) Linux's console are shown as:
`(bbg)$ echo "Hello embedded world!"`
3. Almost all commands are case sensitive.

Revision History

- Jan 4, 2021: Initial version for class
- Jan 19: Changed prompts to `(bbg)$` and `(host)$`

1. NFS Server Setup

1. If you have not done so already, create a `cmpt433/public/` folder in your home directory and make it read/write/executable by everyone.

```
(host)$ cd ~
(host)$ mkdir -p cmpt433/public
(host)$ chmod a+rwX cmpt433/public
```

2. Install the NFS server program on your host computer:

```
(host)$ sudo apt-get install portmap nfs-kernel-server
```

3. Find your host's IP address:

```
(host)$ ifconfig
ens33      Link encap:Ethernet  HWaddr 00:0c:29:53:e3:5a
            inet addr:192.168.0.102  Bcast:192.168.0.255  Mask:255.255.255.0
            inet6 addr: fe80::fe39:ab90:3579:543d/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:215512 errors:0 dropped:0 overruns:0 frame:0
            TX packets:102313 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:300079150 (300.0 MB)  TX bytes:7129865 (7.1 MB)

enx8030dc9359be Link encap:Ethernet  HWaddr 80:30:dc:93:59:be
            inet addr:192.168.7.1  Bcast:192.168.7.3  Mask:255.255.255.252
            inet6 addr: fe80::adbc:5be0:f486:c64b/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:46 errors:0 dropped:0 overruns:0 frame:0
            TX packets:107 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:6934 (6.9 KB)  TX bytes:18801 (18.8 KB)

...
```

- In my case, my VM has two Ethernet connections:

ens33: my hard-wired Ethernet network

enx8030dc9359be: my BeagleBone via Ethernet over USB

4. Configure the server by editing the `/etc/exports` file:

```
(host)$ sudo gedit /etc/exports
```

5. Add the following line to the end of the `/etc/exports` file:

```
/home/brian/cmpt433/public 192.168.0.0/255.255.255.0(rw,sync,no_subtree_check)
```

- You must enter *your* user name on the host computer. You can find this by:

```
(host)$ echo $USER
```

brian

- Change the IP address (`192.168.0.0`) to be the base of the network you are actually on.

- For example, if you are using wired Ethernet and your host's `eth0` IP address is 111.222.333.444 with a network mask ("Mask") of 255.255.255.0, then set the IP address in this line to: 111.222.333.0. The full line you need to add is:

```
/home/brian/cmpt433/public 192.168.0.0/255.255.255.0(rw,sync,no_subtree_check)
```

- If you are connecting to the board via Ethernet over USB, your host IP address is 192.168.7.1, so use just the following line:

```
/home/brian/cmpt433/public 192.168.7.0/255.255.255.252(rw,sync,no_subtree_check)
```

- If you are connecting via *both* Ethernet over USB, *and* a wired Ethernet connection, add both of the above two lines to the file (one for each network). This then shares the directory via either network.

6. Any time you change the `/etc/exports` file, you must restart the NFS server on the host. You should only need to do this once, as the next time your computer starts up it will be loading the correct configuration data from the `/etc/exports` file.

Restart the server (output shown below command in smaller font):

```
(host)$ sudo exportfs -rav
exporting 192.168.0.0/255.255.255.0:/home/brian/cmpt433/public
exporting 192.168.7.0/255.255.255.252:/home/brian/cmpt433/public

(host)$ sudo /etc/init.d/nfs-kernel-server restart
[ ok ] Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
```

7. Check that the correct directory is exported using the command:

```
(host)$ showmount -e
Export list for ubuntu:
/home/brian/cmpt433/public 192.168.7.0/255.255.255.252,192.168.0.0/255.255.255.0
```

8. Troubleshooting:

- Ensure the `public` directory is set up as expected.
- Check the path you are using for the NFS directory, especially that you put your user name instead of “user”.
- Double check that you set the `/etc/exports` file correctly. Ensure you have the correct “base” address of your network.
- When you run the `exportfs` command (using `sudo`), ensure it prints out the networks/directories you expect to be sharing. If not, double check the `/etc/exports` file.
- If you are unable to write to the NFS mounted folder (permission denied error), ensure there is no space before or during the “(rw, sync, no_subtree_check)”
- Ensure your home directory is not encrypted. Look for a hidden folder named like ‘.encrypted’

2. Mounting an NFS Drive on the Target

These steps assume that the NFS server is setup, the target has booted Linux, and a console/terminal is open on the board using the serial port (via `Screen` or `Minicom`), or `SSH`, or `telnet`.

1. On the **target**, ensure networking is correctly setup. Generally, this will be done using DHCP.

Check using:

```
(bbg)$ ifconfig
```

- If the target's IP address is incorrect, try running the DHCP client on the target:

```
(bbg)$ sudo dhclient
```
- Ensure you can ping the host PC from the target. For example, if your host PC address is 192.168.7.1 then:

```
(bbg)$ ping 192.168.7.1
```

2. Create the mount point on your target system. We'll mount the NFS server to `/mnt/remote`:

```
(bbg)$ sudo mkdir /mnt/remote
(bbg)$ sudo chown debian /mnt/remote
```

3. Mount the NFS server on the target using the following command on the target. The command should be entered all on one line:

```
(bbg)$ sudo busybox mount -o tcp -t nfs -o nolock
Ser.ver.IpH.ere:/Abs/Path/Of/Share/On/Server /mnt/remote
```

- This will likely be similar to the following (userName is your home computer's user name).

The command should all be entered on one line:

```
(bbg) $ sudo busybox mount -o tcp -t nfs -o nolock  
192.168.7.1:/home/userName/cmpt433/public /mnt/remote
```

4. On the host PC, make some change to the shared folder, such as adding a file. For example:

```
(host) $ cd ~/cmpt433/public  
(host) $ echo Hello via NFS > nfs_message.txt
```

5. On the target, change to the /mnt/remote directory and list the files; you should be able to access these files (display them with cat, run them like ./helloWorld, copy them with cp).

```
(bbg) $ cd /mnt/remote  
(bbg) $ ls -l  
(bbg) $ cat nfs_message.txt
```

6. Create a mount script on the target

- Create a script to easily mount the NFS folder from your BeagleBone using echo.

```
(bbg) $ echo YourCommandHere > mountNFS.sh
```

For example (all on one line):

```
(bbg) $ echo sudo busybox mount -o tcp -t nfs -o nolock  
192.168.7.1:/home/userName/cmpt433/public /mnt/remote > mountNFS.sh
```

- Change the permissions on the file to be executable:

```
(bbg) $ chmod +x mountNFS.sh
```

- In the future, you can run this command via:

```
(bbg) $ ./mountNFS.sh
```

7. If needed, you can unmount with:

```
(bbg) $ sudo umount /mnt/remote
```

8. Troubleshooting:

- When you try and mount, if you get the error:

```
mount: 192.168.7.1:/home/user/cmpt433/public failed, reason given by server: Permission denied  
mount: mounting 192.168.7.1:/home/user/cmpt433/public on /mnt/remote failed: Bad file descriptor
```

- Ensure you have run the command with super user credentials (use sudo)
- Ensure you have completed all the steps for setting up the server correctly. Pay special attention that you configured the IP address of your network correctly based on how your BeagleBone is currently connecting to your host PC.
- Ensure you have the correct full path of the shared folder on the host; ensure it starts with a /
- Check network settings on server and target, and server is reachable (use ping).
- Check the user ID in path are correct, and that /mnt/remote exists.
- Check the directory permissions are correct on the server (read/write).
- If you are in the /mnt/remote directory when it is mounted, you will first need to leave the directory (cd ..) and then re-enter the directory.
- If you get the error while trying to edit/remove a file from the BeagleBone such as:
rm: cannot remove `/mnt/remote/test': Permission denied
Then the permissions are wrong on the server's public folder. Use chmod to change permissions so all users have full read/write/execute permissions.

- If you get an error about “Stale file handle”, reboot the target and try mounting again
 - If you are unable to write to the NFS mounted folder (permission denied error), ensure there is no space before or during the “(rw, sync, no_subtree_check)”
9. Still not working? Double check the above steps and trouble shooting steps, reboot the host and target, then post on the class forum for help. The following is useful information to include in your help request. Run the commands and show the output (copy and paste session).
- Run on the host:
 - Network info of host:
(host) \$ **ifconfig**
 - Ping the target from the host
 - Display the permissions on the public folder:
(host) \$ **ls -la ~/cmpt433/public/**
 - Display current user:
(host) \$ **echo \$USER**
 - Contents of `exports` file:
(host) \$ **cat /etc/exports**
 - Restart the server:
(host) \$ **sudo exportfs -ra**
(host) \$ **sudo /etc/init.d/nfs-kernel-server restart**
 - List of mounts:
(host) \$ **showmount -e**
 - Run on the target:
 - Network info of target:
(bbg) \$ **ifconfig**
 - Ping the host from the target
 - Display folder on target:
(bbg) \$ **ls -la /mnt/**
 - List of mounted directories:
(bbg) \$ **mount**
 - Full command used on the target to mount the host, along with its output.