NFS Guide

by Brian Fraser

Last update: Jan 11, 2024

This document guides the user through

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

Guide has been tested on

BeagleBone (Target):	**Debian 11.8**
PC OS (host):	**Debian 11.8** (or higher)

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 11, 2024: Initial version for class

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

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

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

3. Boot your BBG and if using a VM, map the BBG to the VM.
4. Find your host's IP address:

```
(host)$ ip -4 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85470sec preferred_lft 85470sec
4: enxb0d5cc4700d8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    inet 192.168.7.1/24 brd 192.168.7.255 scope global noprefixroute enxb0d5cc4700d8
       valid_lft forever preferred_lft forever
3: enxb0d5cc4700da: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 192.168.6.1/24 brd 192.168.6.255 scope global noprefixroute enxb0d5cc4700da
       valid_lft forever preferred_lft forever
```

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

     enp0s3: my VM's connection to the host OS to access the internet
     enxb0d5cc4700d8: my BeagleBone via Ethernet over USB (may appear out of order)

   - There are other connections as well;

     lo: Loop back (always 127.0.0.1) for talking to the local machine (talking to itself)
     enxb0d5cc4700da: 2nd Ethernet over USB connection to the target; we don't use this one

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

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

6. For Ethernet over USB, add the following line to the end of the `/etc/exports` file:

   ```
   /home/brian/cmpt433/public 192.168.7.0/255.255.255.0(rw,sync,no_subtree_check)
   ```

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

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

   - If you are using ***wired*** Ethernet and your host IP address is 111.222.333.444 with a network mask ("Mask") of 255.255.255.0, then add:

   ```
   /home/brian/cmpt433/public 111.222.333.0/255.255.255.0(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.

7. 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 with the following two commands (output shown below each):

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

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

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

   ```
   (host)$ /usr/sbin/showmount -e
   Export list for my-VM:
   /home/brian/cmpt433/public 192.168.7.0/255.255.255.252
   ```

9. 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 and you can ping the host:

   ```
   (bbg)$ ip -4 addr
   (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. On the target, install the NFS client.
    - You must have internet access working from the target. See the Networking guide for “Connecting target to internet” if using Ethernet over USB.
    - Ensure the target has internet access:
      ```
      (bbg)$ ping google.ca
      ```
    - Install NFS client:
      ```
      (bbg)$ sudo apt update
      (bbg)$ sudo apt upgrade             # Optional, but a reasonable idea
      (bbg)$ sudo apt install nfs-common
      ```
      Each of these commands may take a few minutes to complete, or seem to pause at 95% or more for a minute. Just be patient!
4. Mount the NFS server on the target using the following command on the target.
    ```
    (bbg)$ sudo mount -t nfs 192.168.7.1:/home/brian/cmpt433/public /mnt/remote
    ```
    - Change the IP address and username in the path to match your setup.
    - If using an older version of BBG image, try using busybox with the following command all on one line
      ```
      (bbg)$ sudo busybox mount -o tcp -t nfs -o nolock
            192.168.7.1:/home/userName/cmpt433/public /mnt/remote
      ```
5. 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
    ```
6. 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
    ```
7. 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 mount -t nfs 192.168.7.1:/home/brian/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
      ```
8. If needed, you can unmount with:
    ```
    (bbg)$ sudo umount /mnt/remote
    ```
9. 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 /
    - If you are running on a VM, ensure that the BeagleBone is correctly mapped to the VM. It can happen that the BBG is still mapped to the host OS (Windows), and the host OS has a network connection to the target so your Linux command to SSH to the target may work (through the host OS); however, since there is no direct network connection to the Linux VM trying to mount from the target will fail. Run `` `ip addr` `` on the host to see if it has the expected network connections, and if needed map the hardware to the VM.
- Check network settings on host and target, and host is reachable
  ```
  (bbg)$ ping 192.168.7.1
  ```
- 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)`"
- If you get the error: "mount: /mnt/remote: bad option; for several filesystems (e.g. nfs, cifs) you may need a /sbin/mount.<type> helper program." then you need to run
  ```
  (bbg)$ sudo apt install nfs-common
  ```

10. 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)$ ip addr
      ```
    - 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)$ ip addr
    ```
  - 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.