

B4: Network filesystems

Motivation

Network filesystems create a way to access files on another computer as if they are located on your computer. A basic approach to accessing remote files would be to download them, edit them and then upload the edited versions to the server. Mounting the files as a directory on your computer makes it easier to manage and use the files and synchronize changes between your computer and the remote server. Data integrity loss due to device failure can be very problematic. To prevent such data loss, redundancy and integrity mechanisms can be integrated into file systems.

Description of the exercise

In this exercise you will setup various network filesystems and raid 5. You have to compare network filesystems shortly using internet (or book) sources and your own experiments. Please use NFSv3 for this exercise. Doing the demo with NFSv4 is also acceptable, as long as you can answer the questions.

Additional reading

You should have a basic idea of filesystems in general and filesystems in Linux in particular. Also, you should have an idea of how Linux treats devices in general and storage and virtual devices in particular. Here are some resources:

1. [Linux Filesystem Introduction](#)
2. [Device File](#)
3. [File System Architectures](#) (an informal guide)
4. [File system](#)

In addition, you can read about:

- [Samba](#) - Windows interoperability suite of programs for Linux and Unix
- NFS - Network File System - [Sourceforge](#), [Wikipedia](#)
- [sshfs](#) - SSH Filesystem (and [FUSE](#))
- [WebDAV](#) - Web-based Distributed Authoring and Versioning [1]

1. Preparation

You'll need two virtual machines for this exercise. Add aliases (*lab1* and *lab2*) for the addresses to */etc/hosts*.

Create two new users (e.g. "testuser1" and "testuser2") with `adduser` to both the computers. Ensure that users have the same UID on both computers (eg. testuser1 UID is 1001 on lab1 and lab2, testuser2 is 1002). The easiest way is to create both users in

the same order on both computers. Make sure you have a home directory for testuser1 on lab1.

2. Configuring and testing NFS

NFS is an acronym for "network filesystem". NFS is implemented for nearly all unix variations and even for windows.

Make sure you have nfs-kernel-server installed on lab1. Export /home directory via [/etc/exports](#). Restart the NFS server daemon. Mount lab1:/home to lab2:/mnt. You can change user with su, e.g. "su testuser1". Test that NFS works by writing a file in lab1:/home/testuser1/test.txt and open the same file at lab2:/mnt/testuser1/test.txt.

2 . 1	Demonstrate a working configuration.	1 p
2 . 2	Is it possible to encrypt all NFS traffic? How?	1 p

3. Configuring and testing samba

Samba is unix/linux implementation for normal Windows network shares(netbios and CIFS (common internet filesystem)). You can configure samba via [/etc/samba/smb.conf](#). You can access files shared with samba with command smbclient or by mounting the filesystem via mount, like with NFS. Mounting will require cifs-utils to be installed on lab2.

Start by unmounting with *umount(8)* the NFS directory in lab2 from the previous assignment. If unmounting complains "resource busy", you have a shell with your current directory in the /mnt directory and you need to switch to another directory.

Make sure you have samba installed on lab1. Share /home with read and write permissions (/home shares are already at smb.conf but it needs a little bit of tweaking) and reload samba. Run smbpasswd on lab1 and add testuser1 as a user. Try to mount [//lab1/home/testuser1](#) to lab2:/mnt with username testuser1 and testuser1's password. If it doesn't work, check that necessary services and ports are open.

3 . 1	Demonstrate a working configuration.	2 p
-------------	--------------------------------------	--------

3 . 2	Only root can use mount. What problem does this pose for users trying to access their remote home directories? Is there a workaround for the problem?	1 p
-------------	---	--------

4. Configuring and testing sshfs

sshfs is filesystem for FUSE (filesystem in userspace).

Start by unmounting the samba share on lab2.

Next mount lab1:/home/testuser1 to lab2:/home/testuser1/mnt using *sshfs*. Demonstrate this to the assistant.

4 . 1	Provide the commands that you used.	1 p
4 . 2	When is <i>sshfs</i> a good solution?	1 p
4 . 3	What are the advantages of FUSE?	1 p
4 . 4	Why doesn't everyone use encrypted channels for all network filesystems?	1 p

5. Configuring and testing WebDAV

WebDAV (Web-based Distributed Authoring and Versioning) is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.

In this exercise we'll use the built-in WebDAV module of Apache2 server platform. Check that *apache2* is installed and enable the *dav_fs* module. Restart *apache2* every time after enabling a module.

Create a directory */var/www/WebDAV* for storing WebDAV related files and add subdirectory files to be shared using WebDAV. Change the owner of the directories

created to *www-data* (Apache's user ID) and the group to your user ID. Change the permissions if needed.

Create an alias to the virtual host file (*/etc/apache2/sites-available/000-default.conf*) so that */var/www/WebDAV/files* can be reached through *http://localhost/webdav*. Enable the virtual host by creating a symbolic link between */etc/apache2/sites-available/000-default.conf* and */etc/apache2/sites-enabled/*.

Restart *apache2* and check that you can reach the server with, for example, *elinks(1)*.

Set up Authorization

Enable the *auth_digest* module for apache. Create a password file for a testuser with *htdigest(1)* to */var/www/WebDAV*. Edit permissions of the file so that only *www-data* and root can access it. Use the following template to add the location to the virtual host file:

```
<Location /webdav>
  DAV On
  AuthType Digest
  AuthName "your_auth_name"
  AuthUserFile path_to_your_password_file
  Require valid-user
</Location>
```

Restart Apache2 and test the server from another machine using *cadaver(1)*. You should reach the server *http://lab1/webdav*.

5 . 1	Demonstrate a working setup. (View for example a web page on one machine and edit it from another using <i>cadaver</i>).	3 p
5 . 2	Demonstrate mounting a WebDAV resource into the local filesystem.	1 p
5 . 3	Does your implementation support versioning? If not, what should be added?	1 p

6. Raid 5

In this task, you are going to establish a Network Attached Storage (NAS) system with lab1 as a server. The server should use Raid for data integrity. Set up Raid 5 on the NAT server and create EXT4 filesystem on the array.

You need at least three partitions to do this, you can either partition current storage or add more virtual storage to your virtual machine. Then use [mdadm](#) tool to create the raid 5. Share the NAS device you setup with NFS.

6 . 1	What is raid? What is parity? Explain raid5?	1p
6 . 2	Show that your raid5 solution is working.	3p
6 . 3	Access the NAS device from lab2 over NFS	1p

7. Final question

7 . 1	Describe briefly a few use cases for <i>samba</i> , <i>nfs</i> , <i>sshfs</i> and <i>WebDAV</i> . Where, why, weaknesses?	4 p
-------------	---	--------

8. Finishing your work

When finishing your work, please remember to backup files related to the assignment and after your demo possibly reset the configuration changes that you did to the lab environment to start the next assignment with a clean slate.