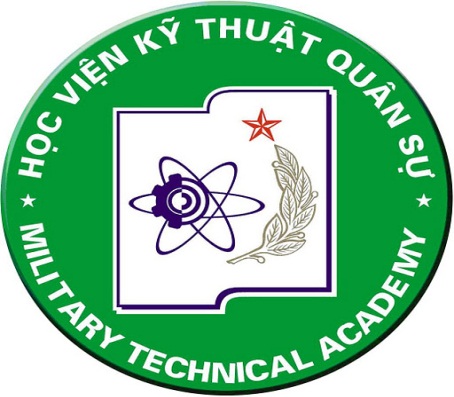
****

**BÁO CÁO**

**HỌC PHẦN**

**THIẾT KẾ MẠNG**

**Đề Tài**

**Xây dựng hệ thống Network File System**

* **Giáo viên hướng dẫn:** Tạ Minh Thanh
* **Nhóm:** 12
* **Lớp:** CNM14
* **Sinh viên thực hiện:**

Phạm Huy Tùng - 15150153

Hoàng Hùng Lâm - 15150138

MỤC LỤC

[MỤC LỤC 2](#_Toc533474278)

[BẢNG PHÂN CÔNG NHIỆM VỤ 3](#_Toc533474280)

[LỜI MỞ ĐẦU 3](#_Toc533474281)

[CHƯƠNG I: GIỚI THIỆU VỀ NFS 3](#_Toc533474282)

[I.KHÁI NIỆM 3](#_Toc533474283)

[1. Hệ thống phân tán 3](#_Toc533474284)

[2. Network File System 4](#_Toc533474285)

[II. VERSIONS 4](#_Toc533474286)

[1. Version 1 4](#_Toc533474287)

[2. Version 2 4](#_Toc533474288)

[3. Version 3 4](#_Toc533474289)

[4. Version 4 5](#_Toc533474290)

[III.KIẾN TRÚC CỦA NFS 6](#_Toc533474291)

[1. Kiến trúc tổng quan NFS Client-Server 6](#_Toc533474292)

[2. Ngăn xếp NFS của máy khách và máy chủ 7](#_Toc533474293)

[3. Kiến trúc của pNFS 8](#_Toc533474294)

[IV.ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA NFS 9](#_Toc533474295)

[1. Ưu điểm 9](#_Toc533474296)

[2. Nhượng điểm 9](#_Toc533474297)

[3. So sánh với dịch vụ chia sẻ file Samba 9](#_Toc533474298)

[V.UID và GID trong NFS 10](#_Toc533474299)

[Chương II: HƯỚNG DẪN CÀI ĐẶT NFS 10](#_Toc533474300)

[I) Yêu cầu hệ thống cơ bản: 10](#_Toc533474301)

[II) Các bước cài đặt NFS: 10](#_Toc533474302)

[Bước 1: Tải và cài đặt 10](#_Toc533474303)

[Bước 2: Tạo thư mục gắn kết trên máy Host 10](#_Toc533474304)

[Bước 3 - Cấu hình NFS Export trên máy chủ Host 11](#_Toc533474305)

[Bước 4 - Điều chỉnh Firewall trên máy Host 13](#_Toc533474306)

[Bước 5 - Tạo điểm Mount Points trên Client 14](#_Toc533474307)

[Bước 6 - Gắn kết các thư mục trên Client 14](#_Toc533474308)

[Bước 7 - Kiểm tra NFS truy cập 15](#_Toc533474309)

[Bước 8 - Gắn kết thư mục NFS từ xa lúc khởi động 16](#_Toc533474310)

[Bước 9 - Unmounting một share NFS từ xa 16](#_Toc533474311)

[CHƯƠNG III: KỊCH BẢN DEMO 18](#_Toc533474312)

[Mô Hình Triển Khai 18](#_Toc533474313)

[Ví dụ 1: Mục đích chia sẻ chung (The General Purpose Share) 19](#_Toc533474314)

[Ví dụ 2: Thư mục chia sẻ Home 21](#_Toc533474315)

[CHƯƠNG IV: Kết quả thực nghiệm 23](#_Toc533474316)

[I) Khó khăn? 23](#_Toc533474317)

[II) Thiếu gì ? Vì sao? 23](#_Toc533474318)

[III) Đề xuất? 23](#_Toc533474319)

[IV) Tài liệu tham khảo 24](#_Toc533474320)

# BẢNG PHÂN CÔNG NHIỆM VỤ

|  |  |
| --- | --- |
| **Thành viên** | **Nhiệm vụ** |
| Hoàng Hùng Lâm | Nghiên cứu tài liệu khoa học |
| Phạm Huy Tùng | Thực hành xây dựng hệ thống |

# LỜI MỞ ĐẦU

Hiện nay hệ điều hành Linux đang ngày càng được ứng dụng trong nhiều ngành khoa học cũng như phục vụ cuộc sống thường ngày, từ ứng dụng trong các hệ thống chuyên biệt phục vụ trong y tế, quân đội tới phục vụ trong hệ thống giáo dục và ứng dụng văn phòng cho người dùng cuối…

Trong những năm gần đây hệ điều hành Linux từng bước được đưa vào sử dụng tại nhiều các công ty ở Việt Nam. Nhiều tổ chức, công ty và các dự án tin học đã chọn Linux là môi trường để phát triển các ứng dụng của mình.

Chính vì thế nhu cầu tìm hiểu hệ điều hành này đang trở nên rất quan trọng và cần thiết.

Đề tài này chỉ đề cập tới một ứng dụng rất nhỏ trong việc khai thác các dịch vụ quản trị mạng trên hệ điều hành Linux đó là Network File System (NFS)

# CHƯƠNG I: GIỚI THIỆU VỀ NFS

## I.KHÁI NIỆM

### Hệ thống phân tán

* Hệ phân tán là tập hợp các máy tính được kết nối với nhau bởi một mạng máy tính và được cài đặt phần mềm hệ phân tán
* Hệ phân tán là một hệ thống có chức năng và dữ liệu phân tán trên các trạm (máy tính) được kết nối với nhau bởi một mạng máy tính
* Hệ phân tán là một tập hợp các máy tính độc lập giao tiếp với người dùng như một hệ thống nhất toàn vẹn.
* Như vậy, có thể nói hệ phân tán bao gồm mạng máy tính và phần mềm phân tán

### Network File System

NFS(**Network File System**) là một hệ thống tập tin phân phối, giao thức ban đầu được phát triển bởi Sun Microsystems vào năm 1984, NFS cho phép một người dùng trên một máy tính khách truy cập dữ liệu thông qua một mạng máy tính giống như được truy cập lưu trữ local. NFS giống như nhiều giao thức khác, được xây dựng trên hệ thống gọi thủ tục từ xa (**Open Network Computing Remote Procedure Call**). NFS là một chuẩn mở được định nghĩa trong RFC (**Request for comments**), cho phép bất kỳ ai thực hiện giao thức.

NFS thường được sử dụng với các hệ điều hành Unix, hệ điều hành MacOS của Apple và các hệ điều hành giống Unix (như Linux và FreeBSD)

## II. VERSIONS

### Version 1

* Sun chỉ sử dụng phiên bản 1 cho mục đích thử nghiệm trong nhà
* Khi nhóm phát triển thêm các thay đổi đáng kể cho phiên bản NFSv1 và phát hành nó ngoài Sun, họ quyết định phát hành phiên bản mới dưới dạng v2

### Version 2

* Phiên bản 2 của giao thức (được định nghĩa trong RFC1094, tháng 3 năm 1989) ban đầu chỉ hoạt động trên giao thức UDP. NFSv2 chỉ cho phép đọc 2 GB đầu tiên do giới hạn 32 bit

### Version 3

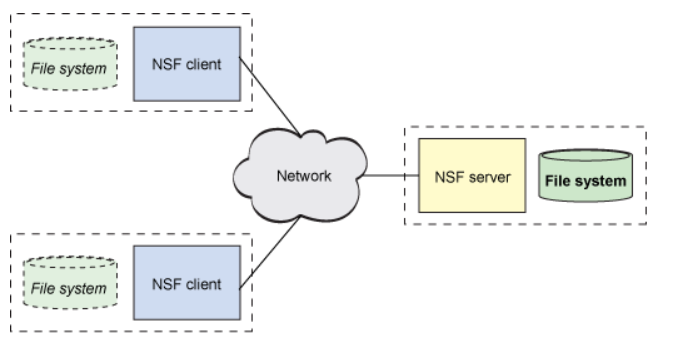
* Được định nghĩa trong RFC1813, tháng 6 năm 1995
* Khắc phục một số hạn chế của v2: Hỗ trợ kích thước và kích thước tệp 64 bit, để xử lý các tệp lớn hơn 2 GB
* Cung cấp hỗ trợ viết không đồng bộ trên máy chủ
* Thông tin trả lời trong RPC chứa các thuộc tính tệp bổ sung, tránh được nhu cầu nạp lại chúng
* Thêm thuộc tính READDIRPLUS (mở rộng đọc từ thư mục), để tối ưu hóa quét thư mục
* NFSv3 có thể hoạt động trên TCP cũng như UDP
* Cải thiện hiệu suất và cho phép nó hoạt động đáng tin cậy hơn trên Internet

### Version 4

* Được định nghĩa trong RFC3010, tháng 12 năm 2000; sửa đổi trong RFC3530, tháng 4 năm 2003 và một lần nữa trong RFC7530 , tháng 3 năm 2015
* Bao gồm cải tiến hiệu suất, yêu cầu bảo mật mạnh mẽ và giới thiệu giao thức Stateful
* Phiên bản 4 trở thành phiên bản đầu tiên được phát triển với Internet Engineering Task Force (IETF) sau khi Sun bàn giao sự phát triển của các giao thức NFS
* **Phiên bản NFS 4.1**(được định nghĩa trong RFC5661 , tháng 1 năm 2010) nhằm cung cấp hỗ trợ giao thức để tận dụng các triển khai máy chủ nhóm bao gồm khả năng cung cấp truy cập song song có thể mở rộng cho các tệp được phân phối giữa nhiều máy chủ (phần mở rộng pNFS).
* **NFS phiên bản 4.2** (được định nghĩa trong RFC7862, tháng 11 năm 2016) với các tính năng mới bao gồm: server-side tạo bản sao và bản sao, ứng dụng I/O tư vấn, tập tin thưa thớt, phòng không gian, khối dữ liệu ứng dụng (ADB), dán nhãn NFS với sec\_label có chứa bất kỳ hệ thống bảo mật MAC nào và hai hoạt động mới cho pNFS (LAYOUTERROR và LAYOUTSTATS).

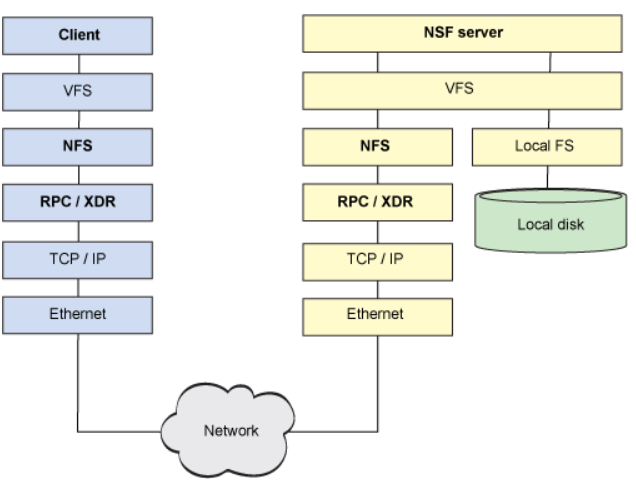
## III.KIẾN TRÚC CỦA NFS

### Kiến trúc tổng quan NFS Client-Server

****

NFS sử dụng mô hình Client-Server và chia sẻ file thông qua giao thức mạng (TCP). Máy chủ thực hiện hệ thống tệp chia sẻ và lưu trữ mà khách hàng đính kèm. Các máy khách thực hiện giao diện người dùng với hệ thống tệp được chia sẻ, được gắn trong không gian tệp cục bộ của máy khách.

### Ngăn xếp NFS của máy khách và máy chủ

****

Trong Linux, công tắc hệ thống tệp ảo (VFS) cung cấp các phương tiện để hỗ trợ đồng thời nhiều hệ thống tệp trên một máy chủ. VFS xác định lưu trữ yêu cầu nào được yêu cầu, sau đó hệ thống tệp nào phải được sử dụng để đáp ứng yêu cầu. Vì lý do này, NFS là một hệ thống tập tin có thể cắm được giống như bất kỳ hệ thống nào khác. Sự khác biệt duy nhất với NFS là các yêu cầu đầu vào/đầu ra (I/O) có thể không được đáp ứng cục bộ, thay vì phải đi qua mạng để hoàn thành.

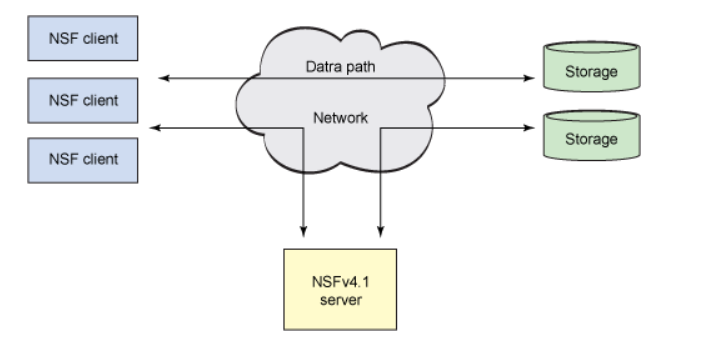
Khi một yêu cầu được tìm thấy được định sẵn cho NFS, VFS chuyển nó tới thể hiện NFS trong nhân. NFS diễn giải I/O yêu cầu và chuyển nó thành một thủ tục NFS (OPEN, ACCESS, CREATE, READ, CLOSE, REMOVE, v.v.) Các thủ tục này, được ghi lại trong NFS RFC cụ thể, chỉ định các hành vi trong giao thức NFS. Khi một thủ tục được chọn từ yêu cầu I/O, nó được thực hiện trong lớp gọi thủ tục từ xa (RPC). Như tên của nó, RPC cung cấp các phương tiện để thực hiện các cuộc gọi thủ tục giữa các hệ thống. Nó so sánh yêu cầu NFS và các đối số đi kèm với nhau, quản lý việc gửi chúng đến một đối tác từ xa thích hợp, sau đó quản lý và theo dõi phản hồi, cung cấp cho người yêu cầu thích hợp.

Hơn nữa, RPC bao gồm một lớp tương tác quan trọng được gọi là ***biểu diễn dữ liệu ngoài*** (XDR), đảm bảo rằng tất cả những người tham gia NFS đều nói cùng một ngôn ngữ khi nói đến các kiểu dữ liệu. Khi một kiến ​​trúc cụ thể thực hiện một yêu cầu, biểu diễn kiểu dữ liệu có thể khác với máy chủ đích thỏa mãn yêu cầu. XDR quản lý các kiểu chuyển đổi thành biểu diễn chung (XDR) để tất cả các kiến ​​trúc có thể tương tác và chia sẻ các hệ thống tệp. XDR chỉ định định dạng bit cho các loại như float và thứ tự byte cho các kiểu như mảng cố định và độ dài biến đổi. Mặc dù XDR được biết đến nhiều nhất với việc sử dụng nó trong NFS, nhưng nó là một đặc tả hữu dụng bất cứ khi nào bạn đang xử lý nhiều kiến ​​trúc trong một cài đặt ứng dụng chung.

Khi XDR đã dịch dữ liệu sang biểu diễn chung, yêu cầu được chuyển qua mạng được đưa ra một giao thức tầng vận tải. NFS ban đầu đã sử dụng giao thức Universal Datagram Protocol (UDP), nhưng ngày nay TCP thường được sử dụng để có độ tin cậy cao hơn.

Tại máy chủ, NFS hoạt động theo cách tương tự. Yêu cầu này sẽ tăng lưu lượng mạng, thông qua RPC / XDR (để dịch các kiểu dữ liệu sang kiến ​​trúc của máy chủ) và đến máy chủ NFS. Máy chủ NFS chịu trách nhiệm đáp ứng yêu cầu. Yêu cầu được chuyển tới trình nền NFS, xác định cây hệ thống tệp mục tiêu cần cho yêu cầu và VFS lại được sử dụng để truy cập vào hệ thống tệp đó trong bộ nhớ cục bộ. Lưu ý ở đây rằng hệ thống tệp cục bộ tại máy chủ là một hệ thống tệp Linux điển hình. Như vậy, NFS không phải là một hệ thống tệp theo nghĩa truyền thống mà thay vào đó là một giao thức để truy cập các hệ thống tệp từ xa.

### Kiến trúc của pNFS



NFSv4.1 giới thiệu khái niệm NFS song song (**pNFS**) để nhân rộng và hiệu suất cao hơn. Để hỗ trợ mở rộng quy mô lớn hơn, NFSv4.1 thực hiện kiến ​​trúc dữ liệu/siêu dữ liệu phân tách với phân chia theo cách tương tự như các hệ thống tệp được phân cụm. Như được hiển thị trên hình, pNFS chia hệ sinh thái thành ba phần: máy khách, máy chủ và lưu trữ. Bạn có thể thấy rằng hai đường dẫn tồn tại: một cho dữ liệu và một cho kiểm soát (**Storage**). pNFS phân chia bố cục dữ liệu từ chính dữ liệu, cho phép kiến ​​trúc đường dẫn kép. Khi khách hàng muốn truy cập một tệp, máy chủ sẽ phản hồi với bố cục. Bố cục mô tả ánh xạ của tệp tới các thiết bị lưu trữ. Khi máy khách có bố cục, nó có thể truy cập trực tiếp vào bộ lưu trữ mà không phải làm việc thông qua máy chủ (cho phép mở rộng quy mô và hiệu suất cao hơn)

Cả dữ liệu và siêu dữ liệu được lưu trữ trong vùng lưu trữ. Các máy khách có thể thực hiện I/O trực tiếp khi nhận bố cục và máy chủ NFSv4.1 xử lý việc lưu trữ và quản lý siêu dữ liệu. Mặc dù hành vi này không nhất thiết phải mới, pNFS bổ sung khả năng hỗ trợ nhiều phương thức truy cập cho bộ lưu trữ. Ngày nay, pNFS hỗ trợ sử dụng các giao thức dựa trên khối, giao thức dựa trên đối tượng và chính NFS (ngay cả ở dạng không pNFS).

## IV.ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA NFS

### Ưu điểm

* Máy chủ và máy khách không trạng thái
* Máy chủ có thể được khởi động lại và người dùng trên máy khách có thể không biết
* Phân biệt máy client/server xảy ra ở cấp application/user không phải là cấp hệ thống => rất linh hoạt, vì vậy cần phải xử lý trong cấu hình/quản trị của mình

### Nhượng điểm

* Sử dụng xác thực RPC => dễ bị giả mạo
* Dữ liệu hệ thống tập tin được truyền đi trong văn bản rõ ràng => dữ liệu có thể được sao chép

### So sánh với dịch vụ chia sẻ file Samba

|  |  |
| --- | --- |
| **Samba** | **NFS** |
| Là một chương trình cung cấp SMB, một giao thức có nguồn gốc từ các hệ thống Windows | Là một giao thức có nguồn gốc từ các hệ thống UNIX |
| Chia sẻ hệ thống tập tin giữa nền tảng UNIX và WINDOWNS | Chia sẻ hệ thống tập tin giữa nền tảng UNIX |
| Khi kết nối với một FS qua SMB, bạn xác thực bằng tên người dùng/mật khẩu và trong toàn bộ phiên | Hoàn toàn dựa vào bảo mật "tin cậy" (bạn tin tưởng bảo mật của máy đó tương đương với máy chủ) |
| Hỗ trợ các máy chủ trạng thái | Là một máy chủ phi trạng thái |

## V.UID và GID trong NFS

Khi máy chủ NFS được cấu hình để từ chối quyền truy cập root từ xa, UID và GID được chọn sẽ được sử dụng để xử lý các yêu cầu từ người dùng root từ xa. Nên chọn UID và GID từ hệ thống để cung cấp mức độ truy cập không đặc quyền thích hợp.

# Chương II: HƯỚNG DẪN CÀI ĐẶT NFS

## I) Yêu cầu hệ thống cơ bản:

**Host**: 192.168.43.49

**Client**: 192.168.43.81

**NFS sử dụng thủ tục RPC (Remote Procedure Calls) để gửi, nhận yêu cầu giữa máy trạm và máy chủ nên dịch vụ portmap (dịch vụ quản lý yêu cầu RPC) cần phải được khởi động trước. Trên máy chủ NFS (máy dự định sẽ chia sẻ dữ liệu) khởi động hai dịch vụ nfs và portmap.**

**Nó nên được đặt tại /sbin nhưng đôi khi trong /usr/sbin. Hầu hết các bản phân phối linux gần đây đều khởi động dịch vụ này trong “kịch bản khởi động” (boot scripts –tự khởi động khi server khởi động) nhưng vẫn phải đảm bảo nó được khởi động đầu tiên trước khi bạn làm việc với NFS (chỉ cần gõ lệnh netstat -anp |grep portmap để kiểm tra).**

## II) Các bước cài đặt NFS:

$sudo apt-get update

$sudo apt-get upgrade

### Bước 1: Tải và cài đặt

#### a) Trên máy Host:

$sudo apt install nfs-kernel-server

#### b) Trên máy Client:

$sudo apt install nfs-common

### Bước 2: Tạo thư mục gắn kết trên máy Host

#### Ví dụ 1: Tạo một thư mục cho mục đích chia sẻ chung (a General Purpose Mount)

-Đầu tiên, tạo một thư mục chia sẻ được gọi là NFS

$sudo mkdir /var/nfs/general –p

-Vì tạo ra nó với lệnh sudo, thư mục này được sở hữu bởi root trên máy chủ

$ls -la /var/nfs/general

4 drwxr-xr-x 2 root root 4096 Jul 25 15:26 .

**NFS sẽ chuyển bất kỳ hoạt động *root* nào trên máy client thành các *nobody:nogroup* ủy nhiệm như một biện pháp bảo mật. Do đó, cần thay đổi quyền sở hữu thư mục để khớp với các thông tin đăng nhập đó.**

$sudo chown nobody:nogroup /var/nfs/general

-Thư mục này đã sẵn sàng để xuất (export).

#### Ví dụ 2: Xuất thư mục home

**Mục đích là để làm cho các thư mục người dùng trong *home* được lưu trữ trên các máy chủ có sẵn trên các máy chủ client, trong khi cho phép các quản trị viên đáng tin cậy của các máy chủ client truy cập để thuận tiện quản lý người dùng.**

**Để làm điều này, chúng tôi sẽ xuất thư mục */home*. Nếu nó đã tồn tại, không cần phải tạo nữa. Sẽ không thay đổi quyền. Nếu làm việc đó, nó sẽ gây ra vấn đề.**

### Bước 3 - Cấu hình NFS Export trên máy chủ Host

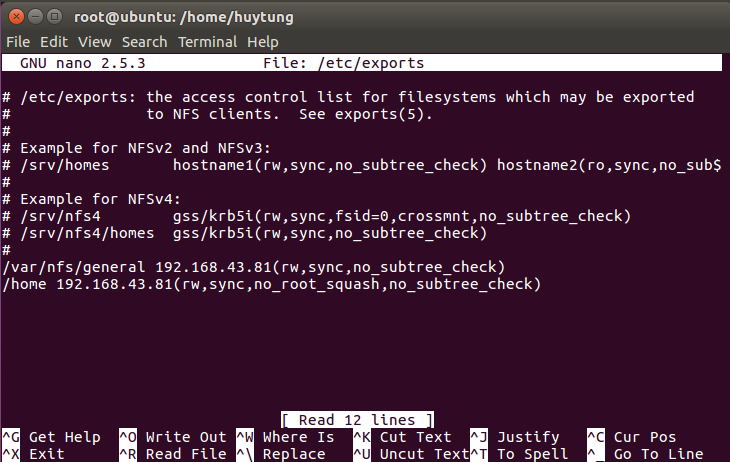
-Mở tập tin /etc/exports

$sudo nano /etc/exports

-Các tập tin có comment cho thấy cấu trúc chung của mỗi dòng cấu hình. Cú pháp cơ bản là:

directory-to-share client(share\_option1,...,share\_optionN)

-Ta sẽ thấy như hình dưới:



**Ý nghĩa của một số kí tự:**

**rw**: Tùy chọn này cho phép máy tính client truy cập cả đọc và viết vào bộ đĩa (volume).

**ro**: Quyền chỉ đọc.  
**noaccess**: Cấm truy cập vào các thư mục cấp con của thư mục được chia sẻ.

**sync**: Tùy chọn này bắt buộc NFS phải ghi các thay đổi vào đĩa trước khi trả lời. Điều này dẫn đến một môi trường ổn định và phù hợp hơn kể từ khi trả lời phản ánh tình trạng thực tế của bộ đĩa (volume) từ xa. Tuy nhiên, nó cũng làm giảm tốc độ của hoạt động tập tin.

**no\_subtree\_check**: tùy chọn này ngăn cản việc kiểm tra cây con, đó là một quá trình mà host phải kiểm tra xem các tập tin thực sự vẫn có sẵn trong cây xuất cho mỗi yêu cầu.

**no\_root\_squash**: Theo mặc định, NFS chuyển yêu cầu từ người dùng root từ xa vào một người dùng không có đặc quyền trên máy chủ. Điều này đã được dự định như là tính năng bảo mật để ngăn chặn một tài khoản root trên máy khách (client) sử dụng hệ thống tập tin của máy chủ như là root.

**Ví dụ**: Bạn muốn chia sẻ thư mục */var/www/html* cho các máy tính có địa chỉ IP là 192.168.3.2, 192.168.3.5, 192.168.3.7 có quyền đọc, ghi là:

/var/www/html 192.168.3.2(rw) 192.168.3.5(rw) 192.168.3.7(rw)

-Khởi động lại máy chủ NFS:

$sudo systemctl restart nfs-kernel-server

### Bước 4 - Điều chỉnh Firewall trên máy Host

-Kiểm tra tình trạng tường lửa:

$sudo ufw status

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

-Mở cổng 2049 trên máy chủ:

$sudo ufw allow from 192.168.43.81 to any port nfs

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

2049 ALLOW 192.168.43.81

OpenSSH (v6) ALLOW Anywhere (v6)

### Bước 5 - Tạo điểm Mount Points trên Client

**Lưu ý: Nếu có những tập tin và thư mục trong điểm gắn kết (mount), ngay sau khi gắn kết (mount) chia sẻ *NFS*, chúng sẽ được ẩn. Hãy chắc chắn rằng bạn *share* trong một thư mục đã tồn tại, thư mục này phải rỗng.**

-Tạo hai thư mục để gắn kết:

$sudo mkdir -p /nfs/general

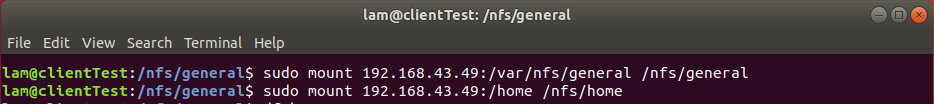
$sudo mkdir -p /nfs/home

### Bước 6 - Gắn kết các thư mục trên Client

-Gắn kết các *shares* từ các máy chủ trên máy client:

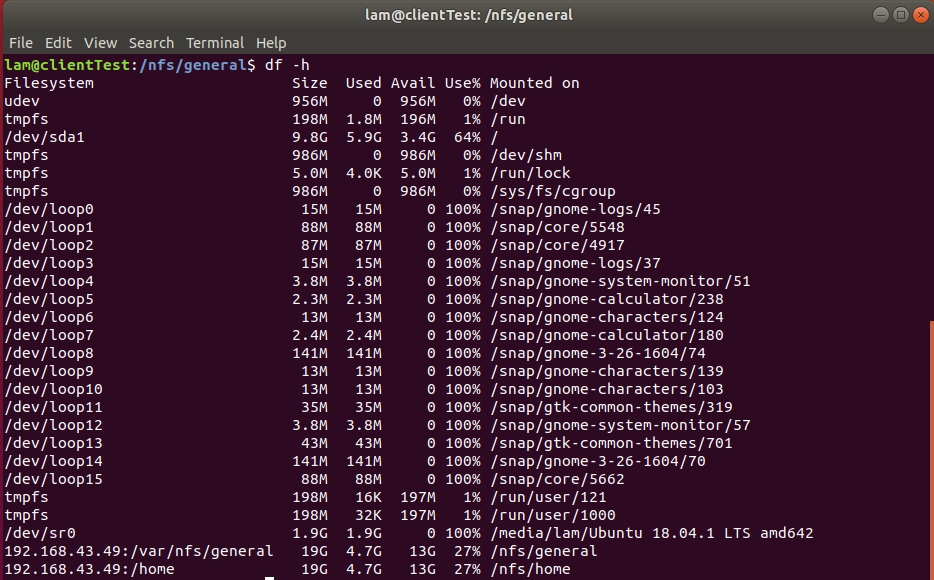
$sudo mount 192.168.43.49:/var/nfs/general /nfs/general

$sudo mount 192.168.43.49:/home /nfs/home



-Kiểm tra gắn kết thành công:

$df –h



-Để xem có bao nhiêu không gian thật sự đang được sử dụng theo từng điểm gắn kết, sử dụng lệnh sau:

$du -sh /nfs/home

Output

36K /nfs/home

**Điều này cho chúng ta thấy rằng nội dung của toàn bộ thư mục *home* chỉ sử dụng *36K* của không gian có sẵn.**

### Bước 7 - Kiểm tra NFS truy cập

$service nfs start

$service portmap start

-Kiểm tra hoạt động của các dịch vụ liên quan đến NFS:

$rpcinfo –p

program vers proto port  
100000 2 tcp 111 portmapper  
100000 2 udp 111 portmapper

100003 2 udp 2049 nfs  
100003 3 udp 2049 nfs  
100005 2 udp 1027 mountd  
100005 2 tcp 1067 mountd  
100005 3 udp 1027 mountd  
100005 3 tcp 1067 mountd

$service nfs stop

$service nfs status

### Bước 8 - Gắn kết thư mục NFS từ xa lúc khởi động

-Có thể gắn kết các shares NFS từ xa tự động lúc khởi động bằng cách thêm chúng vào tập tin */etc/fstab* trên máy khách*.*

$sudo nano /etc/fstab

-Thêm các dòng sau:

192.168.43.49:/var/nfs/general /nfs/general nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0

192.168.43.49:/home /nfs/home nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0

**Các máy chủ client sẽ tự động gắn kết (mount) các phân vùng từ xa lúc khởi động, mặc dù nó có thể mất một vài phút để kết nối được thực hiện và các *shares* có sẵn.**

### Bước 9 - Unmounting một *share* NFS từ xa

$cd ~

$sudo umount /nfs/home

$sudo umount /nfs/general

-Kiểm tra lại dùng lênh:

$df –h

Output

Filesystem Size Used Avail Use% Mounted on

udev 238M 0 238M 0% /dev

none 49M 628K 49M 2% /run

/dev/vda1 20G 1.2G 18G 7% /

none 245M 0 245M 0% /dev/shm

none 5.0M 0 5.0M 0% /run/lock

none 245M 0 245M 0% /sys/fs/cgroup

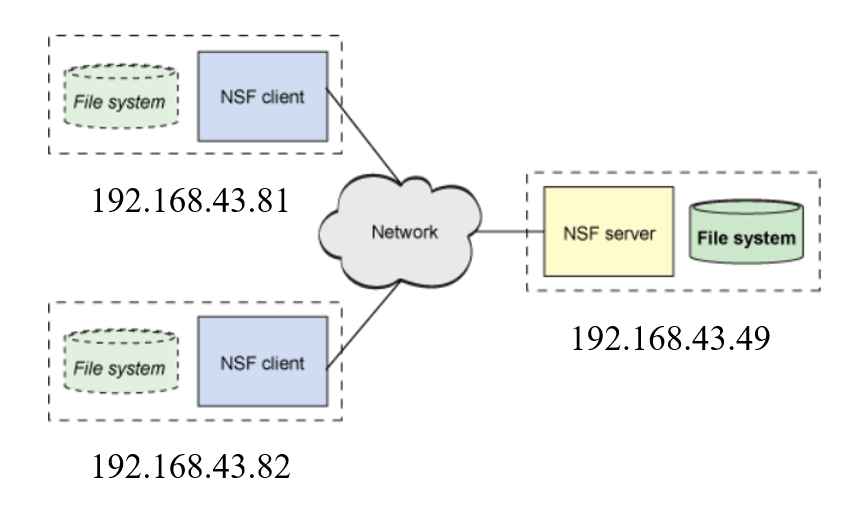
none 49M 0 49M 0% /run/user/0

**Nếu bạn cũng muốn ngăn cản chúng được *remounted* trong lần khởi động tiếp theo, chỉnh sửa */etc/fstab* và *xóa* các dòng hoặc *comment* nó bằng cách đặt một biểu tượng # ở đầu dòng.**

**XONG!!!**

# CHƯƠNG III: KỊCH BẢN DEMO

## Mô Hình Triển Khai

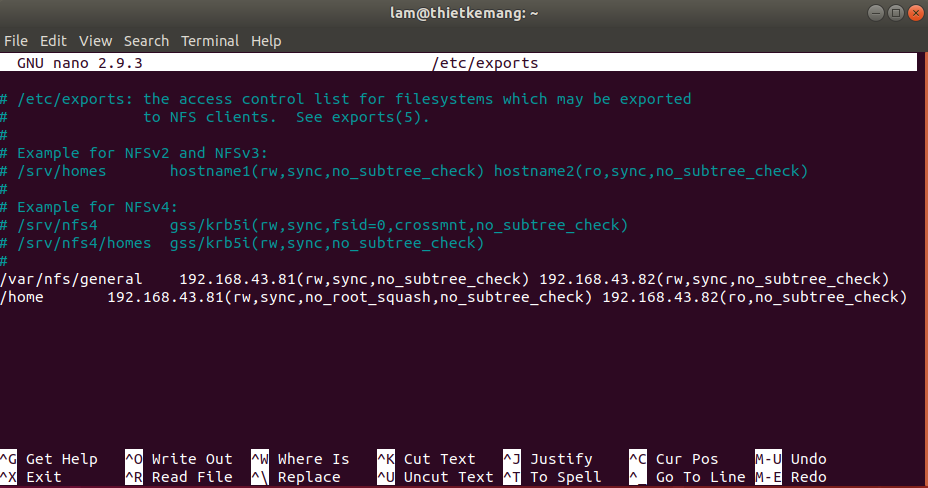


**Server: 192.168.43.49**

**Client 1: 192.168.43.81**

**Client 2: 192.168.43.82**

-**Server** sẽ thêm một số nội dung vào file /etc/exports như sau:

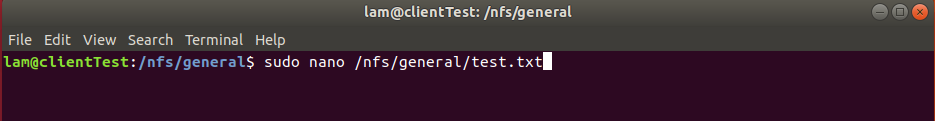


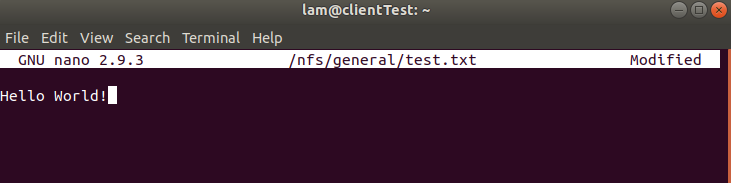
## Ví dụ 1: Mục đích chia sẻ chung (The General Purpose Share)

-Đầu tiên, hãy viết một tập tin thử nghiệm tới /var/nfs/general share với nội dung Hello World!.

Client 1:

$sudo nano /nfs/general/test.txt





$ls -l /nfs/general/test.txt

Output

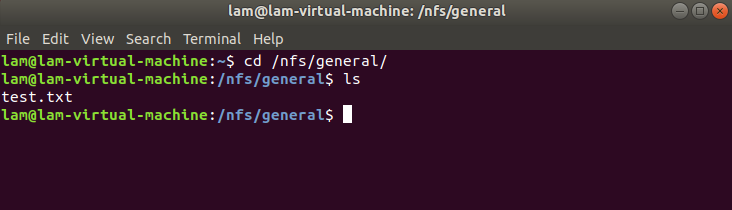
-rw-r--r-- 1 nobody nogroup 0 Nov 4 13:31 /nfs/general/test.txt

**Bởi vì chúng tôi gắn kết bộ đĩa (*volume*) này không làm thay đổi hoạt động mặc định NFS và tạo ra các tập tin với người dùng *root* máy client thông qua lệnh *sudo*, quyền sở hữu của các tập tin mặc định đến *nobody:nogroup*.**

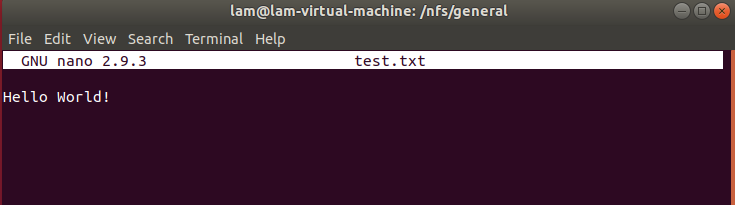
***Superusers* *client* sẽ không thể thực hiện hành động quản trị điển hình, như thay đổi chủ sở hữu của một tập tin hoặc tạo ra một thư mục mới cho một nhóm người dùng, trên chia sẻ *NFS-mounted* này.**

-Bên Client 2 sẽ kiểm tra xem file đã được tạo chưa nếu, được tạo rồi sẽ thay đổi nội dung file thành Hello Boss! và Client 1 sẽ kiểm tra lại nội dung file

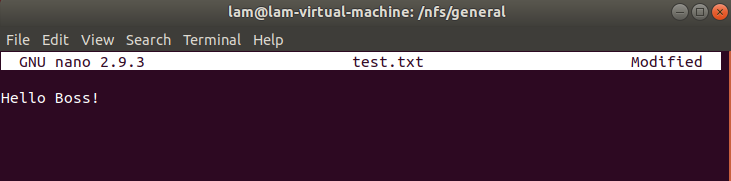
Client 2:

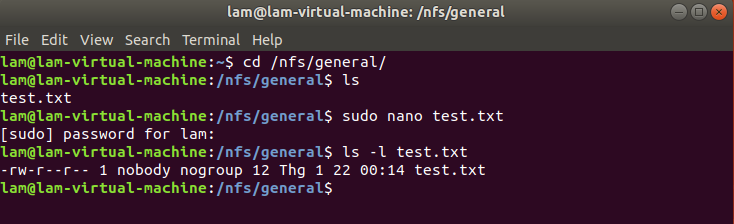


-Kiểm tra nội dung file

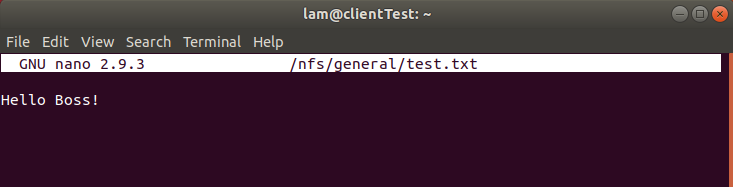


-Thay đổi nội dung file





Client 1: Kiểm tra lại xem nội dung đã thay đổi

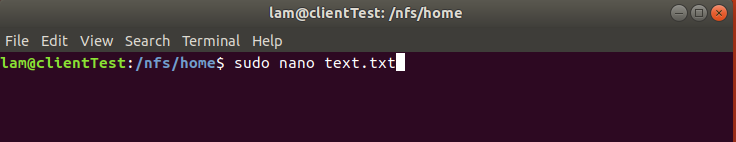


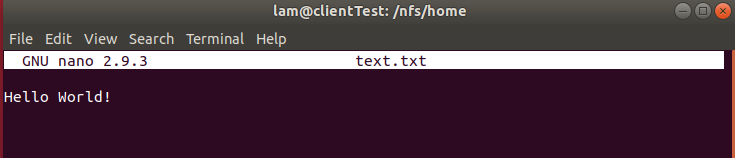
## Ví dụ 2: Thư mục chia sẻ Home

-Để so sánh các điều khoản của mục đích chia sẻ chung với các phần Home Directory, tạo một tập tin Home Directory cùng một cách:

Client 1:

$sudo nano /nfs/home/text.txt





-Sau đó nhìn vào quyền sở hữu của tập tin:

$ls -l /nfs/home/text.txt

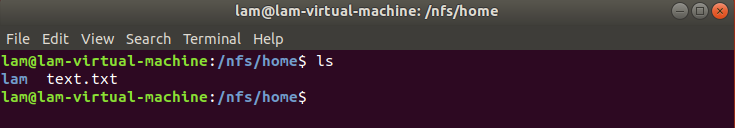
Output

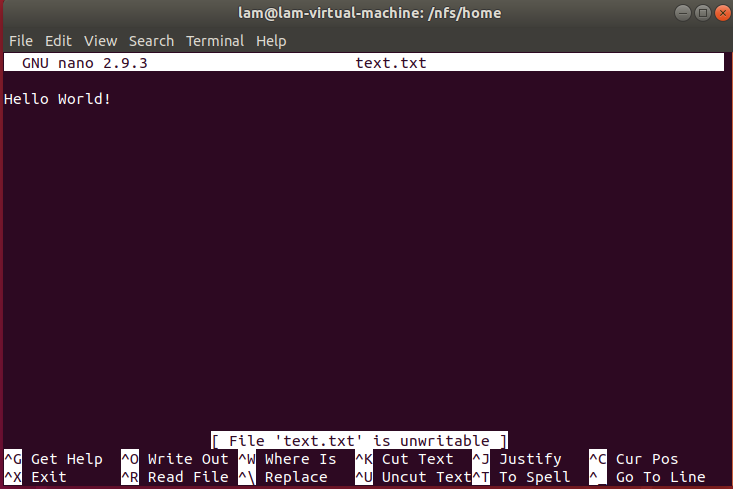
-rw-r--r-- 1 root root 0 Nov 4 13:32 /nfs/home/text.txt

**Ta tạo ra *home.test* như root thông qua lệnh *sudo*, chính xác cách mà ta tạo ra các tập tin *test.txt*. Tuy nhiên, trong trường hợp này nó được sở hữu bởi *root* bởi vì chúng ta huỷ bỏ hoạt động mặc định khi ta xác định các tùy chọn *no\_root\_squash* trên gắn kết (mount) này.**

**Điều này cho phép người sử dụng *root* trên máy *client* hoạt động như *root* và làm cho việc quản lý các tài khoản người dùng thuận tiện hơn nhiều.**

-Bên Client 2 sẽ kiểm tra xem file đã được tạo và vì Client 2 chỉ được phép đọc nên sẽ không thể chỉnh sửa nội dung file





# CHƯƠNG IV: Kết quả thực nghiệm

## I) Khó khăn?

Tài liệu tham khảo chủ yếu trên mạng nên chưa có sự kiểm chứng.

Linux hiện nay có rất nhiều bản phân phối, chính vì thế cũng có sự sai khác giữa các bước thiết lập trên từng bản.

Hiệu năng của máy tính cá nhân chưa được cao nên không demo được tất cả các ví dụ trường hợp

Từ thực tế đó chắc chắn đề tài này không tránh khỏi những thiếu sót và chưa chính xác. Rất mong nhận được sự đóng góp ý kiển của thầy

## II) Thiếu gì ? Vì sao?

NFS còn thiếu tính bảo mật. Vì nội dung truyền đi không được mã hóa

## III) Đề xuất?

Mặc dù NFS là hệ thống tệp mạng phổ biến nhất trên các hệ thống UNIX và Linux, nhưng chắc chắn đây không phải là lựa chọn duy nhất. Trên các hệ thống Windows, Server Message Block (SMB) (còn được gọi là CIFS), là tùy chọn được sử dụng rộng rãi nhất (mặc dù Windows cũng hỗ trợ NFS, vì Linux hỗ trợ SMB).

Một trong những hệ thống tệp phân tán mới nhất, cũng được hỗ trợ trong Linux, là Ceph. Ceph được thiết kế từ đầu như một hệ thống tệp phân tán chịu lỗi với khả năng tương thích với Hệ điều hành di động UNIX (POSIX)

Các ví dụ khác bao gồm OpenAFS, phiên bản nguồn mở của hệ thống tệp phân tán Andrew (từ Carnegie Mellon và IBM), GlusterFS, tập trung vào hệ thống tệp phân tán đa năng để lưu trữ có thể mở rộng và Luster (là hệ thống tệp phân tán song song tập trung vào điện toán cụm). Tất cả đều là giải pháp phần mềm nguồn mở cho lưu trữ phân tán.

## IV) Tài liệu tham khảo

* Lý thuyết:

1) <https://123doc.org//document/4802038-nfs-network-file-system-va-nis-network-information-service.htm>

2) <https://www.ibm.com/developerworks/library/l-network-filesystems/>

3) <https://github.com/hocchudong/ghichepnfs/blob/master/NDChien_Baocao_NFS.md>

* Cài đặt: <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-18-04#step-5-%E2%80%94-creating-mount-points-and-mounting-directories-on-the-client>