Công ty An ninh mạng Viettel

Báo cáo về Backup and Restore

Sinh viên: Nguyễn Đan Trường

*MỤC LỤC*

[Chương 1. Nén và giải nén trong Linux 3](#_Toc152574330)

[Chương 2. Sao lưu và khôi phục hệ thống với dd 3](#_Toc152574331)

[Chương 3. Rsync, scp và sftp 3](#_Toc152574332)

# Nén và giải nén trong Linux

Trong hệ điều hành Linux phân chia thành 3 trường hợp:

- TH1: Nén (dữ liệu được nén lại với kích thước nhỏ hơn dữ liệu gốc)

- TH2: Đóng gói (để phục vụ cho nhiều dữ liệu nhỏ không bị phân

mảnh, chúng ta gom lại thành 1 khối)

- TH3: Đóng gói và nén (Dữ liệu vừa được gói và nén lại, vừa đảm

bảo sự đồng bộ 1 khối cũng như thu gọn lại kích thước dữ liệu)

**Trường hợp 1:**

Có 3 định dạng nén là: \*.zip, \*.gz, \*.bz2

\*.zip: Đây là định dạng thông dụng nhất đối với Linux, dễ sự dụng và nhiều phần mềm hỗ trợ.

Cấu trúc Nén:

*#zip -r <Ten\_file\_nen>.zip <file hay folder cần nén>*

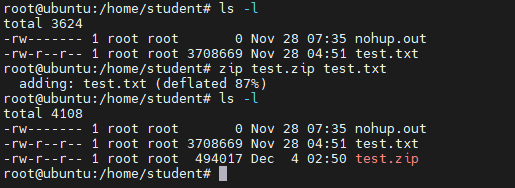
-r: thuật toán đệ quy (recurse)

Giải Nén:

*#unzip <Ten\_file\_nen>.zip*

Đặt password:

*#zip --password <pass> <Ten\_file\_nen>.zip <file hay folder cần nén>*



\*.gz: GZIP được dùng khá phổ biến trong nền tảng Unix/Linux. GZIP chỉ có thể

làm việc trên 1 tập tin hoặc 1 dòng dữ liệu. Do đó nó không thể lưu trữ được nhiều

tập tin. Vì vậy nếu muốn sử dụng cho nhiều tập tin thì chúng ta phải sử dụng TAR

đóng gói chúng lại trước.

Cấu trúc Nén:

*#gzip <Ten\_file\_nen>*

-c: compress - nén

Giải Nén:

*#gzip -dv <Ten\_file\_nen>.gz*

*Hoặc #gunzip <Ten\_file\_nen>.gz*

-d: decompress: giải nén

V: Xem quá trình extract(Verbose)



\*.bz2:

Cấu trúc Nén:

*#bzip2 -c <Ten\_file\_nen>*

-c: compress - nén

Giải Nén:

*#bzip2 -dv <Ten\_file\_nen>.bz2*

*Hoặc #bunzip <Ten\_file\_nen>.bz2*

-d: decompress: giải nén

V: Xem quá trình extract

**Trường hợp 2: đóng gói**

Câu lệnh và công cụ sử dụng là: tar

Cấu trúc đóng gói:

*#tar -cvf <Ten\_file\_đóng gói>.tar <file hay folder cần đóng gói>*

Trong đó: C là create, v là verbose, f là file

Cấu trúc mở đóng gói:

*#tar -xvf <Ten\_file\_đóng gói>.tar*

Trong đó: x là extract

**Trường hợp 3: Đóng gói và nén**

* Nén và giải nén với định dạng \*tar.gz:

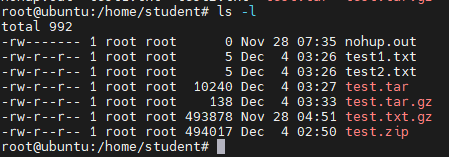
File \*tar.gz là file nén mặc định của linux, điểm mạnh là tiết kiệm dung lượng sau khi nén.

Để nén một thư mục ta sử dụng lệnh sau:

*tar -czvf < Ten\_file\_nen >.tar.gz < file hay folder cần nén >*

Giải nén:

*tar -zxvf < Ten\_file\_nen >.tar.gz*

**

* Nén và giải nén với định dạng \*tar.bz2

Loại file nén này cũng tương tự như tar.gz

Để nén một thư mục ta sử dụng lệnh sau:

*tar -cjvf < Ten\_file\_nen >.tar.bz2 < file hay folder cần nén >*

Giải nén:

*tar -jxvf < Ten\_file\_nen >.tar.bz2*

# Sao lưu và khôi phục hệ thống với dd

## Sao lưu và phục hồi hệ thống với dd

Lệnh dd dùng để copy dữ liệu thô ở mức thấp >> copy theo block, dd có thể copy 1 phân vùng ổ đĩa cứng lưu ra file và ngược lại, dd đọc dữ liệu từ một file và ghi ra thành một file theo một định dạng nào đó.

File ở đây theo nghĩa rộng, thường là device file, không nhất thiết phải là file thông thường.

Cú pháp:

***dd if={input file} of={output file} bs={block size} count={how many blocks}***

Các tham số của lệnh dd theo kiểu option=value

- if: đường dẫn file nguồn để đọc dữ liệu vào

- of: đường dẫn file đích mà dữ liệu ghi ra

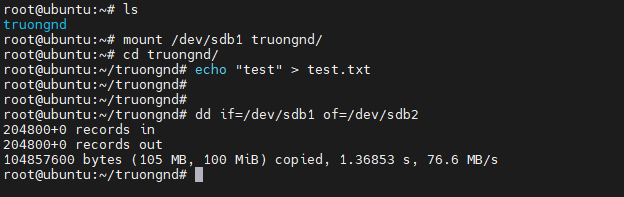
Nếu không chỉ rõ if và of thì mặc định dd sẽ lấy các kí tự từ /dev/stdin (bàn phím) và ghi dữ liệu ra /dev/stdout (màn hình)

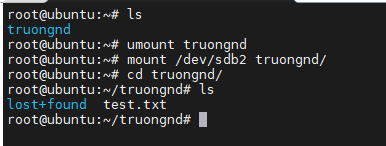
- bs: dữ liệu được đọc vào/ghi ra thành từng khối block, bs (block size) là kích thước của khối, block size mặc định là 512 byte. Ví dụ: bs=5 (5 byte), bs=5MB...

- count: số block được đọc vào và ghi ra. Nếu thiếu count lệnh dd sẽ đọc hết toàn bộ if và ghi toàn bộ dữ liệu ra of

Backup toàn bộ dữ liệu ổ cứng /dev/sda sang ô cứng /dev/sdb

*# dd if=/dev/sda of=/dev/sdb*





Ghi đĩa CD ra file .iso

*# dd if=/dev/cdrom of=cd.iso*

Tạo một file giả làm partition

*# dd if=/dev/zero of=fs1.img bs=1M count=100*

Format file-system

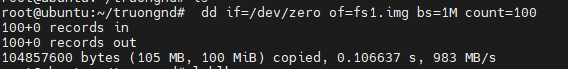
*# mkfs.ext3 fs1.img*

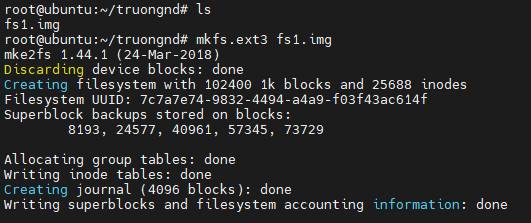
Mount file-system

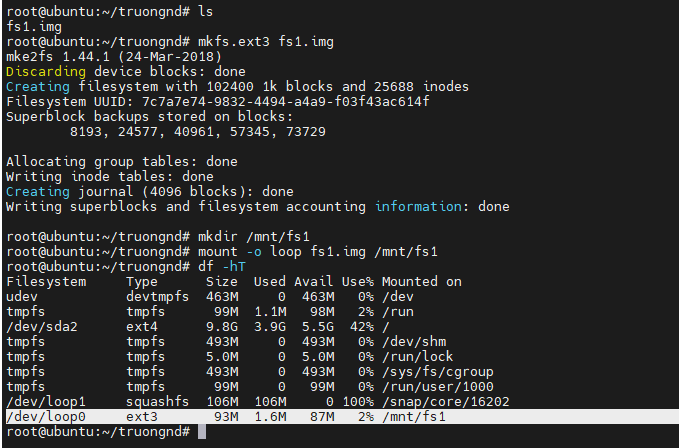
*# mkdir /mnt/fs1*

*# mount -o loop fs1.img /mnt/fs1*

*# df –hT*

**

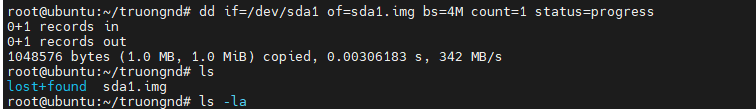




Kiểm tra phân vùng boot trong hệ thống:



Phân vùng /dev/sda1 bắt đầu từ sector 2048 và có kích thước 2048 sectors tương được với 1M.



Trong trường hợp hệ thống bị trục trặc, không thể khởi động Linux từ ổ cứng, có thể phục hồi được một số lỗi thông qua cơ chế khởi động Phục hồi Hệ thống (Linux rescue mode)

Khi khởi động ở chế độ này, một phiên bản thu gọn của Linux và một hệ thống file ảo được nạp và chạy ngay trên RAM hệ thống. Hệ thống file thật trên ổ cứng sẽ được tìm kiếm và ánh xạ vào một thư mục của hệ thống file ảo này. Người dùng có thể dùng lệnh chroot để chuyển qua hệ thống file thật và xử lý sự cố. Thông thường nếu tìm thấy, nó sẽ được ánh xạ vào thư mục /mnt/sysimage của hệ thống ảo. Do đó cần chuyển root làm việc sang /mnt/sysimage, lúc đó đường dẫn đến các file thi hành và thư viện của hệ thống mới được thiết lập lại cho đúng và các chương trình trong hệ thống lúc này mới có thể thực thi được.

Chuyển root sang từ /mnt/sysimage về /

*# chroot /mnt/sysimage*

## Xóa 446 byte đầu tiên của MBR và phục hồi

Xóa phần GRUB đặt trong 446 byte đầu tiên của MBR

*# dd if=/dev/zero of=/dev/sda bs=446 count=1*

Khởi động lại và kiểm tra hệ thống không boot được từ GRUB

*# reboot*

**Phục hồi dữ liệu**

Khởi động bằng đĩa DVD cài đặt ở chế độ Linux rescue

*# chroot /mnt/sysimage*

Cài đặt GRUB lên MBR bằng lệnh grub-install

*# grub-install /dev/sda*

Hoặc copy lại từ file backup MBR

*# dd if=mbr.bin of=/dev/sda bs=446 count=1*

## Xóa 512 byte của MBR và phục hồi

Không ghi đè 512 byte của MBR vì từ byte 447 -> 512 lưu bảng partition. Nếu xóa mất thì việc lập lại bảng này rất khó. Nên backup MBR sang một máy tính khác.

Xóa 512 byte của MBR

*# dd if=/dev/zero of=/dev/sda bs=512 count=1*

Khởi động lại và kiểm tra hệ thống không nhận dạng được các phân vùng ổ cứng

*# reboot*

Phục hồi dữ liệu

Khởi động hệ thống bằng đĩa LiveCD và copy file MBR đã backup trước đó lên.

Trên LiveCD thực hiện các lệnh:

*$ su - root*

*# passwd*

*# service sshd start*

*# service iptables stop*

Kết nối đến máy Linux bằng SFTP và copy file mbr.bin, khôi phục MBR bằng lệnh dd

*# dd if=mbr.bin of=/dev/sda1 bs=512 count=1*

Kiểm tra các phân vùng đã được khôi phục và khởi động lại hệ thống

*# fdisk -l*

*# reboot*

# Rsync, scp và sftp

## Rsync

Rsync là công cụ mạnh mẽ được sử dụng để sao chép và đồng bộ dữ liệu giữa các máy tính. Được thiết kế dể hoạt động hiệu quả với các tập tin lớn và thường được sử dụng trong các kịch bản sao lưu, đồng bộ hóa tập tin giữa các máy chủ và quản lý dữ liệu từ xa.

Ví dụ, một tập tin lớn có dung lượng 5 GB. Sử dụng chức năng checksum mã MD5 cho các file ở hai đầu của đường dẫn đồng bộ để xem liệu chúng có khác nhau hay không. Nếu mã MD5 không khớp, một tiện ích đồng bộ hóa sẽ gửi toàn bộ file 5GB từ nguồn đến đích. Để giải quyết vấn đề phải gửi toàn bộ file, Rsync chia nhỏ toàn bộ file đó ra, so sánh mã băm của các đoạn nhỏ, và sau đó chỉ chuyển những khối dữ liệu không khớp. Nếu chỉ có 1MB dữ liệu thực sự thay đổi trong một file dung lượng 5 GB, thì chỉ 1MB dữ liệu được truyền từ nguồn tới đích → Tiết kiệm thời gian và băng thông.

Cú pháp: ***rsync -options source destination***

Đồng bộ dữ liệu: SERVER1: /var/www/html ›› SERVER2: /var/www

**Thực hiện rsync trên SERVER2**

***# rsync -avzP --delete -e ssh root@SERVER1:/var/www/html /var/www***

-a: (archive) các file backup có thể giữ nguyên các thuộc tính: sự phân quyền, chủ sở hữu, nhóm sở hữu, thời gian, recursive mode và các symbolic links

-v: để trình bày chi tiết

-z: nén dữ liệu để tiết kiệm băng thông

-P: hiển thị chi tiết quá trình rsync

--delete: nếu các file ở nguồn (source) bị xóa thì các file đó củng được xóa ở đích (destination)

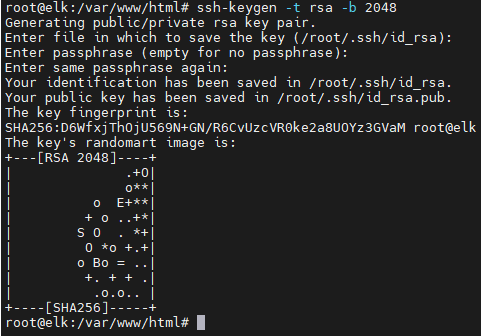
-e ssh: chỉ định sử dụng ssh với rsync

--exclude=FOLDER\_hay\_FILE: loại trừ không đồng bộ các file

**Chứng thực RSA**

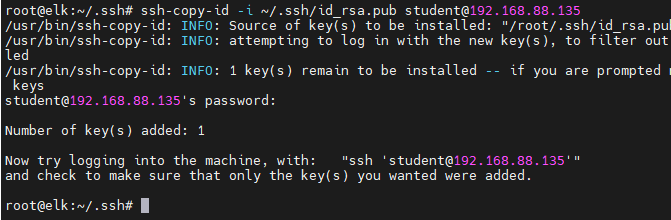
Gen cặp khóa rsa độ dài 2048 bit trên SERVER2

*# ssh-keygen -t rsa -b 2048*

**

Copy public key id\_rsa.pub lên SERVER1

*# ssh-copy-id -i ~/.ssh/id\_rsa.pub root@SERVER1*

**

Lệnh ssh-copy-id copy public key và đổi tên key thành authorized\_keys

Đặt lịch đồng bộ dữ liệu

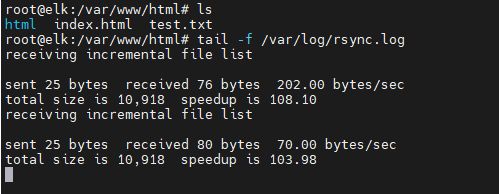
# vim /etc/cron.d/rsync

*\*/1 \* \* \* \* root rsync -avz --delete -e ssh root@SERVER1:/var/www/html /var/www >>/var/log/rsync.log 2>&1*

**

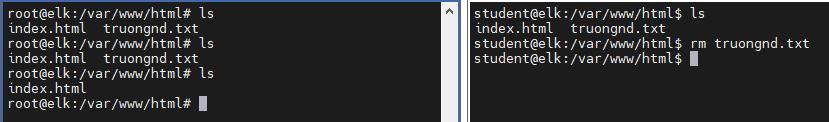
# tail -f /var/log/cron

# tail -f /var/log/rsync.log



Sau mỗi phút thư mục SERVER2: /var/www/html sẽ đồng bộ với SERVER1: /var/www/html



**

Nếu dịch vụ ssh không dùng port 22 sử dụng lệnh rsync thêm option "ssh -p portnumber":

*rsync -avz --delete -e "ssh -p portnumber" user@remoteip:/path/to/files/ /local/path*

Tạo scirpt đồng bộ chạy trong background

# vim rsync.sh

#!/bin/bash

WAIT=60

RSYNC=$(which rsync)

while true

do

sleep $WAIT

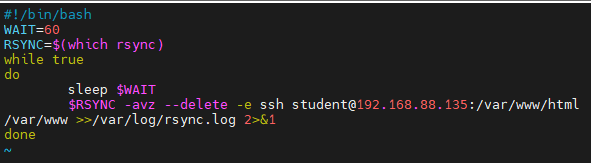
$RSYNC -avz --delete -e ssh root@SERVER1:/var/www/html /var/www

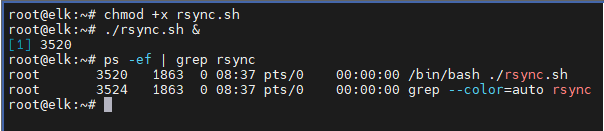
>>/var/log/rsync.log 2>&1

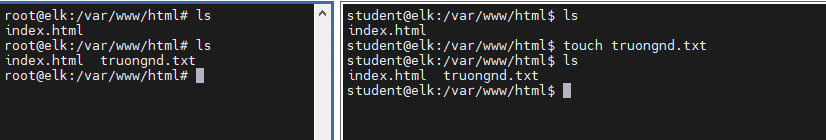
done

# sh rsync.sh &

# ps -ef | grep rsync







## SCP

SCP là viết tắt của “Secure Copy” được sử dụng để sao chép thư mục và file giữa các server Linux một cách an toàn. Lệnh SCP cho phép sao chép giữa các server nội bộ và các server public. Nó sử dụng cùng một xác thực và bảo mật như được sử dụng trong giao thức Secure Shell (SSH). SCP được biết đến với tính đơn giản, bảo mật và tính sẵn sàng.

Khi truyền dữ liệu bằng scp, cả tệp và mật khẩu đều được mã hóa để bất kỳ ai theo dõi lưu lượng truy cập đều không nhận được bất kỳ thông tin nhạy cảm nào.

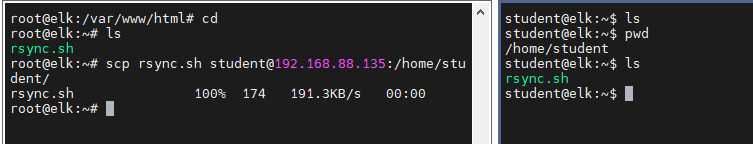
Cú pháp lệnh scp có dạng sau:

***scp [OPTION] [user@]SRC\_HOST:]file1 [user@]DEST\_HOST:]file2***

* OPTION: các tùy chọn scp như mật mã, cấu hình ssh, cổng ssh,…
* [user@]SRC\_HOST:]file1 - Source file
* [user@]DEST\_HOST:]file2 - Destination file Local files phải được chỉ định bằng đường dẫn tuyệt đối hoặc tương đối, trong khi remote file phải bao gồm thông số kỹ thuật của người dùng và máy chủ.
* SCP cung cấp một số tùy chọn kiểm soát mọi khía cạnh của hành vi của nó. Các tùy chọn được sử dụng rộng rãi nhất là:
* -P : Chỉ định cổng ssh máy chủ từ xa.
* -p : Duy trì thời gian sửa đổi và truy cập file
* -q : Sử dụng tùy chọn này nếu bạn muốn loại bỏ đo lường tiến trình và thông báo không lỗi.
* -C : Tùy chọn này buộc scp nén dữ liệu khi nó được gửi đến máy đích
* -r : Tùy chọn này yêu cầu scp sao chép các thư mục một cách đệ quy.

Để có thể sao chép tệp, ít nhất bạn phải có quyền đọc trên tệp nguồn và quyền ghi trên hệ thống đích. Hãy cẩn thận khi sao chép các tệp có cùng tên và vị trí trên cả hai hệ thống, scp sẽ ghi đè tệp mà không có cảnh báo Khi chuyển các tệp lớn, bạn nên chạy lệnh scp bên phiên tmux.

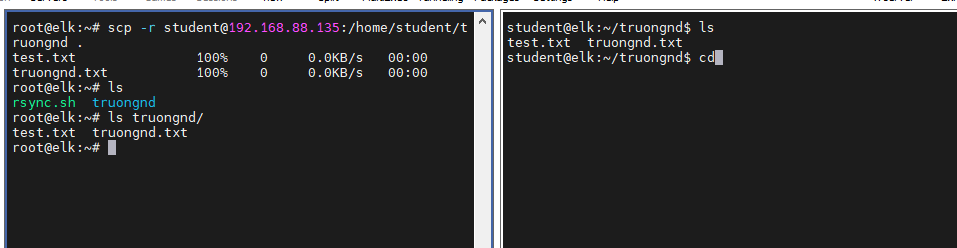
**Sao chép Local file đến Remote system:**

****

Lệnh sao chép một thư mục cũng giống như khi sao chép tệp. Sự khác biệt duy nhất là bạn cần sử dụng cờ -r cho đệ quy. Để sao chép thư mục từ hệ thống cục bộ sang hệ thống từ xa, hãy sử dụng tùy chọn -r:

*scp -r /local/directory* [*remote\_username@10.10.0.2:/remote/directory*](mailto:remote_username@10.10.0.2:/remote/directory)

**Sao chép file/thư mục từ Remote system về local:**

****

**Sao chép file giữa 2 Remote system:**

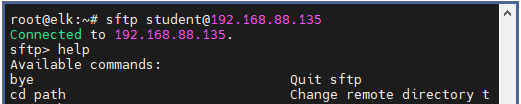
*scp user1@host1.com:/files/file.txt* [*user2@host2.com:/files*](mailto:user2@host2.com:/files)

## SFTP

File Transfer Protocol (FTP) là giao thức được sử dụng rộng rãi để truyền tệp hoặc dữ liệu từ xa ở định dạng không được mã hóa, đây là giao thức không an toàn vì dữ liệu không được mã hóa trong quá trình truyền tải.

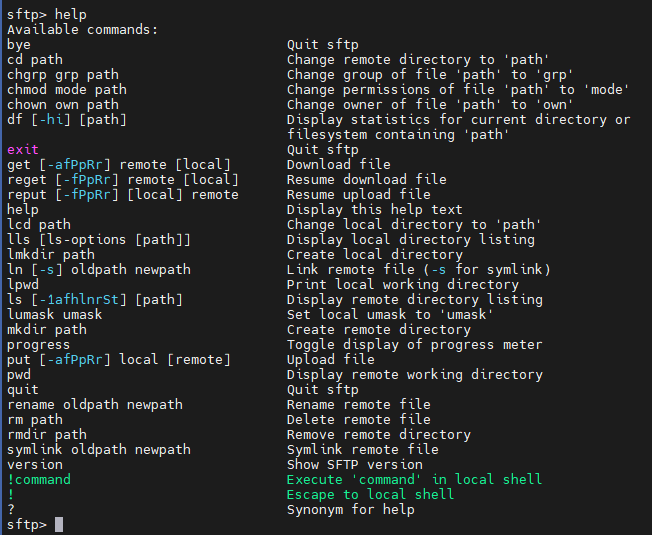
SFTP ( Secure File Transfer Protocol ) chạy qua giao thức SSH trên cổng tiêu chuẩn 22 theo mặc định để thiết lập kết nối an toàn. SFTP đã được tích hợp vào nhiều công cụ GUI ( FileZilla, WinSCP, FireFTP,…)

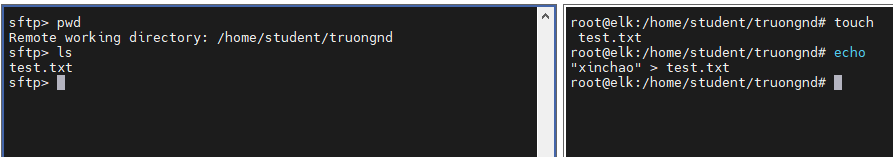
Để sử dụng được sftp thì chúng ta cần thiết lập ssh trước cho hệ thống. Sau đó ta có thể thiết lập một phiên sftp



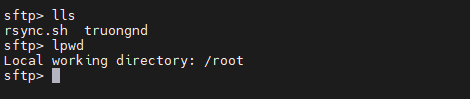
Nếu bạn đang làm việc trên một cổng SSH tùy chỉnh (không phải cổng mặc định 22), bạn có thể mở một phiên SFTP như sau:

*sftp -oPort=custom\_port user@your\_server\_ip\_or\_remote\_hostname*



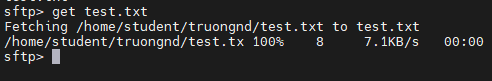


Nếu cần truy cập vào hệ thống file cục bộ, bạn có thể hướng các command tới hệ thống file cục bộ bằng cách đặt “l” trước commad



**Truyền file bằng SFPT**

Nếu bạn muốn download file từ server từ xa, dùng lệnh sau: *get remoteFile*



Bạn có thể sao chép file từ xa sang một tên khác bằng cách: *get remoteFile localFile*

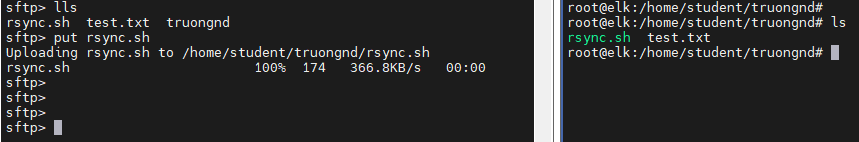
Command *get* cũng có một số flag tùy chọn. Ví dụ: chúng ta có thể sao chép một thư mục và tất cả nội dung của nó bằng cách chỉ định tùy chọn đệ quy:

*get -r someDirectory*

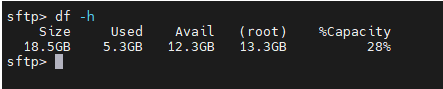
**Truyền file cục bộ sang Remote system**

Việc chuyển các file đến hệ thống từ xa được thực hiện bằng cách sử dụng lệnh:

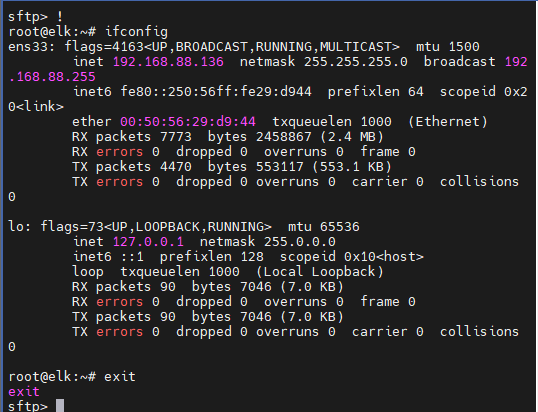
*Put localfile*

**

Bạn cũng có thể kiểm tra dung lượng của remote server trước khi thực hiện upload file:



Command “!” có thể đưa trở lại shell local mà vẫn giữ phiên sftp:



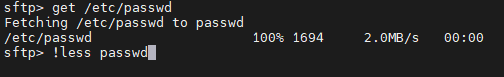
**Thao tác với file trong SFTP**

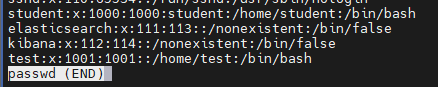
SFTP cho phép bạn thực hiện kiểu bảo trì file cơ bản khi làm việc với cấu trúc phân cấp file.

Ví dụ: bạn có thể thay đổi chủ sở hữu của file trên hệ thống từ xa bằng:

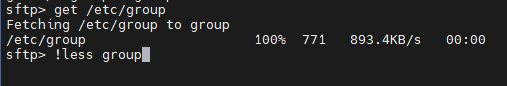
*Chown userID file*

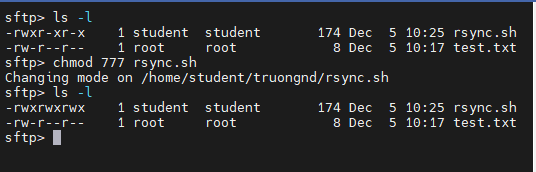
Trong SFTP sử dụng UID thay vì username, để biết được UID từ bên trong SFPT interface, ta sử dụng lệnh:





Tương tự, chúng ta có thể thay đổi chủ sở hữu nhóm của file bằng: *chgrp groupID file*





Đóng kết nối bằng lệnh: bye.