

Hướng Dẫn Chi Tiết: Di Chuyển và Đổi Tên File/Thư Mục Trong Linux

1. Giới Thiệu về Lệnh mv

Lệnh mv là gì?

- mv = move (di chuyển)
- Dùng để di chuyển hoặc đổi tên file/thư mục
- Syntax cơ bản: `mv nguồn đích`

Các tùy chọn quan trọng:

- `-i`: Hỏi trước khi ghi đè
- `-n`: Không ghi đè file đã tồn tại
- `-f`: Ghi đè mà không hỏi
- `-v`: Hiển thị quá trình thực hiện

2. Bài Tập Thực Hành Với File

Bài 1: Đổi Tên File Cơ Bản

1. Tạo file test:

```
# Tạo file gốc
echo "Nội dung test" > oldname.txt

# Đổi tên file
mv oldname.txt newname.txt

# Kiểm tra kết quả
ls -l newname.txt
```

Giải thích:

- File giữ nguyên nội dung
- Chỉ thay đổi tên
- Quyền và timestamp được giữ nguyên

Bài 2: Di Chuyển File

1. Tạo cấu trúc thư mục:

```
# Tạo thư mục
mkdir -p ~/documents/backup
```

```
# Tạo file test
echo "Data cần backup" > data.txt

# Di chuyển file
mv data.txt ~/documents/backup/

# Kiểm tra
ls -l ~/documents/backup/
```

Giải thích:

- File được chuyển sang thư mục mới
- Đường dẫn có thể tuyệt đối hoặc tương đối
- Tên file không thay đổi

Bài 3: Di Chuyển và Đổi Tên Cùng Lúc

1. Thực hiện:

```
# Tạo file và thư mục
echo "Dữ liệu quan trọng" > old_data.txt
mkdir -p ~/archive

# Di chuyển và đổi tên
mv old_data.txt ~/archive/new_data_2024.txt

# Kiểm tra
ls -l ~/archive/
```

Giải thích:

- Kết hợp di chuyển và đổi tên
- File có vị trí và tên mới
- Nội dung vẫn giữ nguyên

3. Bài Tập Thực Hành Với Thư Mục

Bài 1: Đổi Tên Thư Mục

1. Tạo và đổi tên thư mục:

```
# Tạo thư mục
mkdir old_project

# Tạo một số file trong thư mục
touch old_project/{file1.txt,file2.txt}

# Đổi tên thư mục
mv old_project new_project
```

```
# Kiểm tra
ls -R new_project
```

Giải thích:

- Toàn bộ cấu trúc được giữ nguyên
- Chỉ thay đổi tên thư mục
- Các file con không bị ảnh hưởng

Bài 2: Di Chuyển Thư Mục

1. Di chuyển toàn bộ thư mục:

```
# Tạo cấu trúc
mkdir -p ~/projects/backup
mkdir testdir
touch testdir/{test1.txt,test2.txt}

# Di chuyển
mv testdir ~/projects/backup/

# Kiểm tra
ls -R ~/projects/backup/
```

Giải thích:

- Di chuyển cả thư mục và nội dung
- Cấu trúc bên trong được giữ nguyên
- Quyền truy cập không thay đổi

4. Bài Tập Nâng Cao

Bài 1: Sử Dụng Tùy Chọn -i

1. Thực hành với tùy chọn an toàn:

```
# Tạo các file test
echo "File cũ" > test.txt
echo "File mới" > new.txt

# Di chuyển với xác nhận
mv -i new.txt test.txt

# Kiểm tra nội dung
cat test.txt
```

Giải thích:

- `-i` yêu cầu xác nhận trước khi ghi đè
- Tránh mất dữ liệu do sơ suất
- Nên dùng khi làm việc với file quan trọng

Bài 2: Di Chuyển Nhiều File

1. Di chuyển nhiều file cùng lúc:

```
# Tạo môi trường test
mkdir -p ~/backup
touch file1.txt file2.txt file3.txt

# Di chuyển nhiều file
mv file{1,2,3}.txt ~/backup/

# Kiểm tra
ls -l ~/backup/
```

Giải thích:

- Có thể di chuyển nhiều file một lúc
- Sử dụng pattern matching
- Hiệu quả với số lượng file lớn

5. Dự Án Thực Hành: Quản Lý Dự Án

1. Thiết lập môi trường:

```
# Tạo cấu trúc dự án
mkdir -p ~/project_old/{src,docs,tests}
touch ~/project_old/src/{main.py,utils.py}
touch ~/project_old/docs/README.md
```

2. Tổ chức lại dự án:

```
# Đổi tên dự án
mv ~/project_old ~/project_v2

# Tạo thư mục mới và di chuyển code
mkdir -p ~/project_v2/backend
mv ~/project_v2/src/* ~/project_v2/backend/

# Di chuyển và đổi tên tài liệu
mv ~/project_v2/docs/README.md ~/project_v2/README.md
```

6. Lưu Ý Quan Trọng

6.1 An Toàn:

- Luôn dùng `-i` với file quan trọng
- Backup trước khi di chuyển lớn
- Kiểm tra đường dẫn đích

6.2 Hiệu quả:

- Sử dụng tab completion
- Dùng pattern matching khi có thể
- Kiểm tra không gian đĩa đích

6.3 Xử lý lỗi:

- Kiểm tra quyền truy cập
- Xác nhận đủ dung lượng
- Kiểm tra kết quả sau khi di chuyển

7. Best Practices

1. Quy tắc đặt tên:

- Tránh dùng khoảng trắng
- Sử dụng dấu gạch dưới
- Đặt tên có ý nghĩa

2. Tổ chức file:

- Phân loại theo chức năng
- Duy trì cấu trúc rõ ràng
- Backup định kỳ

3. Bảo mật:

- Kiểm tra quyền trước khi di chuyển
- Bảo vệ dữ liệu nhạy cảm
- Ghi log các thay đổi lớn