

# Hướng Dẫn Chi Tiết: Hardlinks Trong Linux

---

## 1. Giới Thiệu về Hardlinks

### 1.1 Hardlink là gì?

- Một tên file khác trỏ đến cùng một inode
- Không thể phân biệt file gốc và hardlink
- Chia sẻ cùng nội dung và quyền truy cập
- Chỉ xóa file khi xóa tất cả hardlinks

### 1.2 Đặc điểm của Hardlink:

- Phải cùng filesystem
- Không thể link đến thư mục
- Cùng inode number
- Cùng quyền và nội dung

## 2. Bài Tập Thực Hành Cơ Bản

### Bài 1: Tạo Hardlink Đơn Giản

#### 1. Tạo file gốc:

```
# Tạo file gốc
echo "Đây là file gốc" > original.txt

# Kiểm tra inode
ls -li original.txt
```

*Giải thích:*

- `-li` hiển thị inode number
- File gốc có một link count = 1

#### 2. Tạo hardlink:

```
# Tạo hardlink
ln original.txt hardlink1.txt

# Kiểm tra
ls -li original.txt hardlink1.txt
```

*Giải thích:*

- Cùng inode number

- Link count tăng lên 2
- Cả hai file có cùng nội dung

## Bài 2: Thử Nghiệm với Nội Dung

### 1. Sửa đổi qua hardlink:

```
# Thêm nội dung qua hardlink
echo "Thêm qua hardlink" >> hardlink1.txt

# Kiểm tra cả hai file
cat original.txt
cat hardlink1.txt
```

*Giải thích:*

- Thay đổi một file ảnh hưởng file còn lại
- Vì cùng trỏ đến một inode
- Cùng chia sẻ dữ liệu

## Bài 3: Kiểm Tra Quyền và Chủ Sở Hữu

### 1. Thay đổi quyền:

```
# Thay đổi quyền trên file gốc
chmod 600 original.txt

# Kiểm tra cả hai file
ls -l original.txt hardlink1.txt
```

*Giải thích:*

- Quyền thay đổi trên cả hai file
- Vì cùng một inode
- Không thể có quyền khác nhau

## 3. Bài Tập Nâng Cao

### Bài 1: Quản Lý Nhiều Hardlinks

#### 1. Tạo nhiều hardlinks:

```
# Tạo file dữ liệu
echo "Dữ liệu quan trọng" > data.txt

# Tạo nhiều hardlinks
ln data.txt link1.txt
```

```
ln data.txt link2.txt
ln data.txt link3.txt

# Kiểm tra link count
ls -li data.txt link*.txt
```

*Giải thích:*

- Link count = 4
- Tất cả file đều ngang hàng
- Không có khái niệm file "gốc"

2. Xóa thử một link:

```
# Xóa một hardlink
rm link1.txt

# Kiểm tra lại
ls -li data.txt link*.txt
```

*Giải thích:*

- Link count giảm xuống 3
- Dữ liệu vẫn tồn tại
- Các link khác không bị ảnh hưởng

## Bài 2: Backup với Hardlink

1. Tạo hệ thống backup:

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

# Tạo file quan trọng
echo "Dữ liệu cần backup" > important.txt

# Tạo backup bằng hardlink
ln important.txt ~/backups/important_backup.txt
```

*Giải thích:*

- Backup không tốn thêm dung lượng
- Cập nhật tự động khi file gốc thay đổi
- Phù hợp cho backup nội bộ

## 4. Dự Án Thực Tế: Quản Lý Phiên Bản

## Bài 1: Quản Lý Config Files

### 1. Thiết lập hệ thống:

```
# Tạo thư mục config
mkdir -p ~/configs/{current,versions}

# Tạo file config
cat > ~/configs/current/app.conf << 'EOF'
# Configuration v1
debug=true
port=8080
EOF

# Tạo phiên bản với hardlink
ln ~/configs/current/app.conf ~/configs/versions/app_v1.conf
```

### 2. Cập nhật config:

```
# Sửa file config
cat > ~/configs/current/app.conf << 'EOF'
# Configuration v2
debug=false
port=8080
EOF

# Tạo phiên bản mới
ln ~/configs/current/app.conf ~/configs/versions/app_v2.conf
```

#### Giải thích:

- Mỗi phiên bản là một hardlink
- Dễ dàng theo dõi thay đổi
- Tiết kiệm dung lượng

## 5. Lưu Ý Quan Trọng

### 5.1 Giới Hạn của Hardlink:

- Không thể link thư mục
- Phải cùng filesystem
- Cần quyền write vào thư mục chứa

### 5.2 So sánh với Symbolic Link:

```
# Tạo symbolic link để so sánh
ln -s original.txt symlink.txt
```

```
# So sánh các loại link  
ls -li original.txt hardlink1.txt symlink.txt
```

*Giải thích:*

- Hardlink: cùng inode, cùng nội dung
- Symlink: inode khác, chỉ trỏ đến file gốc
- Symlink có thể link thư mục và cross-filesystem

## 6. Best Practices

### 6.1 Khi nào dùng Hardlink:

- Backup file cục bộ
- Quản lý phiên bản
- Tiết kiệm dung lượng
- Chia sẻ dữ liệu không đổi

### 6.2 Khi nào không dùng:

- Link thư mục
- Cross-filesystem
- Dữ liệu thường xuyên thay đổi
- Cần tính độc lập

### 6.3 Quản lý Hardlink:

```
# Tìm tất cả hardlinks của một file  
find / -samefile original.txt  
  
# Kiểm tra số lượng hardlinks  
stat original.txt
```