

Linux 1.2

Linux 디렉토리 구조 및 기본 명령어

목차

- [리눅스 디렉토리 구조 및 역할](#)
- [리눅스 기본 명령어](#)
 - [ls 명령어와 옵션](#)
 - [디렉토리 생성](#)
 - [파일 생성](#)
 - [파일/디렉토리 삭제](#)
 - [날짜/시간 확인](#)
 - [달력 출력](#)
 - [파일/디렉토리 복사](#)
 - [파일/디렉토리 이동](#)
 - [파일 내용 확인](#)
 - [페이지 단위 출력](#)

리눅스 디렉토리 구조 및 역할

리눅스 파일 시스템은 계층적 구조로 되어 있으며, 각 디렉토리는 고유한 역할과 목적을 가지고 있다.

주요 디렉토리별 상세 설명

디렉토리	역할 및 기능	주요 내용
/boot	부팅 관련 파일 (커널)	리눅스 커널, 부트로더, initrd 파일
/etc	모든 환경 설정 파일	시스템 구성 파일, 서비스 설정
/lib	동적 라이브러리 파일 (32비트)	공유 라이브러리, 커널 모듈
/lib64	동적 라이브러리 파일 (64비트)	64비트 시스템용 라이브러리
/root	root 사용자 홈 디렉토리	관리자 전용 홈 디렉토리
/sbin	시스템 관련 명령어	시스템 관리 바이너리
/usr	일반 사용자 관련 명령어, 프로그램 설치	사용자 프로그램, 라이브러리
/bin	일반 명령어	기본 사용자 명령어
/dev	장치 파일	하드웨어 디바이스 파일
/home	일반 사용자의 홈 디렉토리	사용자별 개인 폴더

디렉토리	역할 및 기능	주요 내용
/mnt	마운트 디렉토리	임시 파일 시스템 마운트
/proc	가상 파일 시스템	프로세스 및 시스템 정보
/tmp	임시 디렉토리	임시 파일 저장소
/var	시스템이 운영 중 변경되는 파일	로그, 캐시, 메일 등

추가 중요 디렉토리

디렉토리	설명
/opt	추가 소프트웨어 패키지
/media	자동 마운트된 이동식 미디어
/run	런타임 데이터
/sys	시스템 파일 시스템
/srv	서비스 데이터

리눅스 기본 명령어

ls 명령어와 옵션

ls: 파일 및 디렉토리 목록을 출력하는 명령어, 도구, 프로그램이다.

주요 옵션

옵션	기능	설명
-l	자세히 출력	권한, 소유자, 크기, 수정시간 등 상세 정보
-a	숨김 파일까지 출력	. 으로 시작하는 모든 파일 포함
-A	숨김 파일 출력 (. , .. 제외)	현재/상위 디렉토리 제외하고 출력
-h	사람이 읽기 쉬운 크기	KB, MB, GB 단위로 표시
-t	시간 순 정렬	최근 수정된 파일 먼저
-r	역순 정렬	정렬 순서 반대로
-S	크기 순 정렬	큰 파일부터 정렬

ls 명령어 활용 예시

기본 사용법

ls -l

ls -la

상세 정보

숨김 파일 포함 상세 정보

```
ls -lah          # 읽기 쉬운 크기로 표시
ls -ltr          # 시간 순 역정렬
ls -laS          # 크기 순 정렬
```

특정 파일 타입만 보기

```
ls *.txt          # txt 파일만
ls -la | grep "^d" # 디렉토리만
ls -la | grep "^-" # 일반 파일만
```

디렉토리 생성

mkdir [옵션] [생성할 디렉토리명]: 새로운 디렉토리를 생성하는 명령어이다.

기본 사용법

```
# 단일 디렉토리 생성
mkdir dir1

# 여러 디렉토리 동시 생성
mkdir dir1 dir2 dir3

# 범위를 이용한 디렉토리 생성
mkdir dir{5..9}

# 중간 경로까지 생성 (-p 옵션)
mkdir -p dir2/dir3/dir4

# 숨김 디렉토리 생성
mkdir .dir100
```

mkdir 옵션 상세

옵션	기능	예시
-p	부모 디렉토리까지 생성	mkdir -p /home/user/docs/2024
-m	권한 설정	mkdir -m 755 mydir
-v	상세 출력	mkdir -v newdir

```
root@localhost:~  
합계 4  
-rw-----. 1 root root 991 12월 14 20:15 anaconda-ks.cfg  
drwxr-xr-x. 2 root root 6 12월 14 20:19 공개  
drwxr-xr-x. 2 root root 6 12월 14 20:19 다운로드  
drwxr-xr-x. 2 root root 6 12월 14 20:19 문서  
drwxr-xr-x. 2 root root 63 12월 14 20:51 바탕화면  
drwxr-xr-x. 2 root root 6 12월 14 20:19 비디오  
drwxr-xr-x. 2 root root 6 12월 14 20:19 사진  
drwxr-xr-x. 2 root root 6 12월 14 20:19 서식  
drwxr-xr-x. 2 root root 6 12월 14 20:19 음악  
[root@localhost ~]# mkdir dir1  
[root@localhost ~]# ll  
합계 4  
-rw-----. 1 root root 991 12월 14 20:15 anaconda-ks.cfg  
drwxr-xr-x. 2 root root 6 1월 4 20:54 dir1  
drwxr-xr-x. 2 root root 6 12월 14 20:19 공개  
drwxr-xr-x. 2 root root 6 12월 14 20:19 다운로드  
drwxr-xr-x. 2 root root 6 12월 14 20:19 문서  
drwxr-xr-x. 2 root root 63 12월 14 20:51 바탕화면  
drwxr-xr-x. 2 root root 6 12월 14 20:19 비디오  
drwxr-xr-x. 2 root root 6 12월 14 20:19 사진
```

```
root@localhost:~  
[root@localhost dir1]# touch file1  
[root@localhost dir1]# ll  
합계 0  
-rw-r--r--. 1 root root 0 1월 4 20:55 file1  
[root@localhost dir1]# cd ..  
[root@localhost ~]# mkdir dir{5..9}  
[root@localhost ~]# ll  
합계 4  
-rw-----. 1 root root 991 12월 14 20:15 anaconda-ks.cfg  
drwxr-xr-x. 2 root root 19 1월 4 20:55 dir1  
drwxr-xr-x. 2 root root 6 1월 4 20:56 dir5  
drwxr-xr-x. 2 root root 6 1월 4 20:56 dir6  
drwxr-xr-x. 2 root root 6 1월 4 20:56 dir7  
drwxr-xr-x. 2 root root 6 1월 4 20:56 dir8  
drwxr-xr-x. 2 root root 6 1월 4 20:56 dir9  
drwxr-xr-x. 2 root root 6 12월 14 20:19 공개  
drwxr-xr-x. 2 root root 6 12월 14 20:19 다운로드
```

```
[root@localhost dir1]# mkdir -p dir{2..4}
[root@localhost dir1]# ls -R
.:
dir2  dir3  dir4  file1  file5  file6  file7  file8  file9

./dir2:

./dir3:

./dir4:
[root@localhost dir1]#
```

💡 고급 활용 팁

```
# 현재 날짜로 디렉토리 생성
mkdir $(date +%Y-%m-%d)

# 권한과 함께 생성
mkdir -m 700 private_dir

# 여러 레벨 디렉토리 한번에 생성
mkdir -p project/{src,docs,tests}/{2024,2025}
```

📄 파일 생성

touch [옵션] [파일명]: 빈 파일을 생성하거나 기존 파일의 타임스탬프를 변경하는 명령어이다.

🔧 기본 사용법

```
# 단일 파일 생성
touch file1

# 여러 파일 동시 생성
touch file1 file2 file3

# 범위를 이용한 파일 생성
touch file{1..3}

# 특정 시간으로 설정
touch -t [연월일시분초] [파일/디렉토리명]
```

📅 touch 옵션 상세

옵션	기능	예시
-t	특정 시간 설정	touch -t 202312251200 file.txt
-d	날짜 문자열로 설정	touch -d "2023-12-25 12:00" file.txt

옵션	기능	예시
-a	접근 시간만 변경	touch -a file.txt
-m	수정 시간만 변경	touch -m file.txt
-c	파일이 없어도 생성하지 않음	touch -c existing_file.txt

```

root@localhost:~/dir1
drwxr-xr-x. 2 root root  6 12월 14 20:19 사진
drwxr-xr-x. 2 root root  6 12월 14 20:19 서식
drwxr-xr-x. 2 root root  6 12월 14 20:19 음악
[root@localhost ~]# mkdir dir1
[root@localhost ~]# ll
합계 4
-rw----- . 1 root root 991 12월 14 20:15 anaconda-ks.cfg
drwxr-xr-x. 2 root root  6  1월  4 20:54 dir1
drwxr-xr-x. 2 root root  6 12월 14 20:19 공개
drwxr-xr-x. 2 root root  6 12월 14 20:19 다운로드
drwxr-xr-x. 2 root root  6 12월 14 20:19 문서
drwxr-xr-x. 2 root root 63 12월 14 20:51 바탕화면
drwxr-xr-x. 2 root root  6 12월 14 20:19 비디오
drwxr-xr-x. 2 root root  6 12월 14 20:19 사진
drwxr-xr-x. 2 root root  6 12월 14 20:19 서식
drwxr-xr-x. 2 root root  6 12월 14 20:19 음악
[root@localhost ~]# cd dir1
[root@localhost dir1]# ll
합계 0
[root@localhost dir1]# touch file1
[root@localhost dir1]# ll
합계 0
-rw-r--r--. 1 root root 0  1월  4 20:55 file1
[root@localhost dir1]#

```

```

[root@localhost dir1]# touch file{5..9}
[root@localhost dir1]# ll
합계 0
-rw-r--r--. 1 root root 0  1월  4 20:55 file1
-rw-r--r--. 1 root root 0  1월  4 20:57 file5
-rw-r--r--. 1 root root 0  1월  4 20:57 file6
-rw-r--r--. 1 root root 0  1월  4 20:57 file7
-rw-r--r--. 1 root root 0  1월  4 20:57 file8
-rw-r--r--. 1 root root 0  1월  4 20:57 file9
[root@localhost dir1]# S

```

```

합계 0
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
[root@localhost dir1]# ls -al
합계 4
drwxr-xr-x. 2 root root 100 1월 4 20:58 .
dr-xr-x---. 20 root root 4096 1월 4 20:56 ..
-rw-r--r--. 1 root root 0 1월 4 20:58 .file100
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
[root@localhost dir1]#

```

💡 **중요:** 숨김파일 또는 디렉토리 생성 시, 맨 앞에 `.` 을 붙인다.

🔒 숨김 파일 생성 예시

```

# 숨김 파일 생성
touch .hidden_file
touch .config
touch .bashrc

# 숨김 디렉토리 생성
mkdir .hidden_directory

```

🗑️ 파일/디렉토리 삭제

파일과 디렉토리를 삭제하는 두 가지 주요 명령어가 있다.

📁 rm 명령어 (파일 삭제)

rm [옵션] [파일/디렉토리]: 파일이나 디렉토리를 삭제하는 명령어이다.

🔧 주요 옵션

옵션	기능	주의사항
<code>-r</code>	하위 디렉토리 포함 삭제	디렉토리 삭제 시 필수
<code>-f</code>	묻지 않고 강제 삭제	매우 위험, 신중히 사용

옵션	기능	주의사항
<code>-i</code>	삭제 전 확인	안전한 삭제를 위한 옵션
<code>-v</code>	상세 출력	삭제 과정 확인

```
# 파일 삭제
rm file1
rm file1 file2 file3

# 디렉토리 삭제 (하위 포함)
rm -r directory1

# 강제 삭제 (위험!)
rm -rf file{1..5}
rm -rf f*

# 안전한 삭제 (확인 후 삭제)
rm -i important_file.txt
```

rmdir 명령어 (빈 디렉토리 삭제)

rmdir [디렉토리]: 빈 디렉토리만 삭제할 수 있는 안전한 명령어이다.

```
# 빈 디렉토리 삭제
rmdir empty_directory

# 여러 빈 디렉토리 삭제
rmdir dir1 dir2 dir3

# 부모 디렉토리까지 삭제 (모두 비어있는 경우)
rmdir -p path/to/empty/dirs
```



```
root@localhost:~/dir1
[root@localhost dir1]# ll
합계 0
drwxr-xr-x. 2 root root 6 1월 4 21:01 dir2
drwxr-xr-x. 2 root root 6 1월 4 21:01 dir3
drwxr-xr-x. 2 root root 6 1월 4 21:01 dir4
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
[root@localhost dir1]# rm -rf d*
[root@localhost dir1]# ll
합계 0
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
[root@localhost dir1]#
```

```
drwxr-xr-x. 2 root root 6 1월 4 21:07 dir10
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
[root@localhost dir1]# rmdir dir10
[root@localhost dir1]# ll
합계 0
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
```

숨김파일 삭제

중요: 숨김파일 제거 시 `rm -rf .숨김파일명` 을 사용하며, 숨김파일은 하나씩 개별 삭제가 필요하다.

```
# 숨김파일 개별 삭제
rm -rf .hidden_file1
rm -rf .hidden_file2

# 주의: 다음 명령어는 위험함
# rm -rf .* (모든 숨김파일 삭제, 절대 사용하지 말 것!)
```

⚠ 삭제 시 주의사항

```
# 안전한 삭제 습관
alias rm='rm -i'          # 항상 확인 후 삭제
rm -rf /*                 # 절대 실행하지 말 것!
rm -rf /                  # 시스템 파괴 명령어

# 삭제 전 백업 습관
cp important_file.txt important_file.txt.bak
rm important_file.txt
```

📅 날짜/시간 확인

date [옵션]: 시스템의 현재 날짜와 시간을 표시하거나 설정하는 명령어이다.

🔧 기본 사용법

```
# 현재 시간 출력
date

# UTC 시간 출력
date -u

# 특정 형식으로 출력
date +%Y-%m-%d %H:%M:%S

# 특정 날짜 계산
date -d "1 day ago"
date -d "next monday"
date -d "2024-12-25"
```

📊 date 형식 지정자

지정자	의미	예시
%Y	4자리 연도	2024
%m	월 (01-12)	03
%d	일 (01-31)	15

지정자	의미	예시
%H	시간 (00-23)	14
%M	분 (00-59)	30
%S	초 (00-59)	45
%A	요일 전체	Monday
%a	요일 축약	Mon

```

root@localhost dir1]# date
2025. 01. 04. (토) 21:09:09 KST
root@localhost dir1]# date -u
2025. 01. 04. (토) 12:09:12 UTC
root@localhost dir1]#

```

💡 date 명령어 활용 예시

```

# 로그 파일명에 날짜 포함
echo "Log entry" >> log_$(date +%Y%m%d).txt

# 백업 파일명에 타임스탬프
cp important.txt important_$(date +%Y%m%d_%H%M%S).txt

# 시간 계산
date -d "$(date) + 30 days"      # 30일 후
date -d "$(date) - 1 week"     # 1주 전

```

📅 달력 출력

cal [옵션] [설정 값]: 달력을 출력하는 명령어이다.

🔧 기본 사용법

```

# 현재 월 달력
cal

# 3개월 달력 (이전, 현재, 다음)
cal -3

# 연간 달력
cal -y

# 연간 달력 (줄리안 날짜 포함)
cal -yj

# 특정 연도 달력
cal -y 2020

```

특정 월 달력

cal 05 2023

cal 옵션 상세

옵션	기능	예시
-3	3개월 달력	이전/현재/다음 월
-y	연간 달력	전체 년도 달력
-j	줄리안 날짜	1월 1일부터 일수
-m	월요일부터 시작	유럽식 달력

```
[root@localhost dir1]# cal
```

```
1월 2025
일 월 화 수 목 금 토
      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

```
[root@localhost dir1]# cal -3
```

```
12월 2024      1월 2025      2월 2025
일 월 화 수 목 금 토 일 월 화 수 목 금 토 일 월 화 수 목 금 토
 1  2  3  4  5  6  7      1  2  3  4      1
 8  9 10 11 12 13 14    5  6  7  8  9 10 11    2  3  4  5  6  7  8
15 16 17 18 19 20 21   12 13 14 15 16 17 18    9 10 11 12 13 14 15
22 23 24 25 26 27 28   19 20 21 22 23 24 25   16 17 18 19 20 21 22
29 30 31             26 27 28 29 30 31       23 24 25 26 27 28
```

```
[root@localhost dir1]# cal -y
```

```
2025

1월      2월      3월
일 월 화 수 목 금 토 일 월 화 수 목 금 토 일 월 화 수 목 금 토
      1  2  3  4      1      1
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    2  3  4  5  6  7  8
12 13 14 15 16 17 18    9 10 11 12 13 14 15    9 10 11 12 13 14 15
19 20 21 22 23 24 25   16 17 18 19 20 21 22   16 17 18 19 20 21 22
26 27 28 29 30 31     23 24 25 26 27 28     23 24 25 26 27 28 29
                                     30 31

4월      5월      6월
일 월 화 수 목 금 토 일 월 화 수 목 금 토 일 월 화 수 목 금 토
      1  2  3  4  5      1  2  3      1  2  3  4  5  6  7
 6  7  8  9 10 11 12    4  5  6  7  8  9 10    8  9 10 11 12 13 14
13 14 15 16 17 18 19   11 12 13 14 15 16 17   15 16 17 18 19 20 21
20 21 22 23 24 25 26   18 19 20 21 22 23 24   22 23 24 25 26 27 28
27 28 29 30           25 26 27 28 29 30 31   29 30

7월      8월      9월
일 월 화 수 목 금 토 일 월 화 수 목 금 토 일 월 화 수 목 금 토
```

 cal 명령어 활용 팁

```
# 특정 날짜 강조 (ncal 사용)
ncal -h                                # 오늘 날짜 강조 없음
ncal -B1 -A1                           # 이전/다음 월 포함

# 달력을 파일로 저장
cal 2024 > calendar_2024.txt

# 특정 월의 마지막 날 확인
cal $(date +%m) $(date +%Y) | awk 'NF {DAYS = $NF}; END {print DAYS}'
```

파일/디렉토리 복사

cp [옵션] [원본] [대상]: 파일이나 디렉토리를 복사하는 명령어이다.

기본 사용법

```
# 파일을 디렉토리로 복사
cp file1 dir1

# 파일 이름 변경하여 복사
cp file1 file2

# 시스템 파일 복사
cp /etc/passwd .
cp /etc/passwd /home/dir1

# 디렉토리 복사 (하위 포함)
cp -r source_dir dest_dir
```

cp 주요 옵션

옵션	기능	사용 예시
-r, -R	디렉토리 재귀 복사	cp -r dir1 dir2
-p	권한/소유자 보존	cp -p file1 file2
-a	아카이브 모드	cp -a dir1 dir2
-u	최신 파일만 복사	cp -u file1 file2
-i	덮어쓰기 확인	cp -i file1 file2
-v	상세 출력	cp -v file1 file2

```
root@localhost:~/dir1/copy
[root@localhost copy]# ll
합계 0
[root@localhost copy]# cd ..
[root@localhost dir1]# ll
합계 0
drwxr-xr-x. 2 root root 6  1월  4 21:15 copy
-rw-r--r--. 1 root root 0  1월  4 20:55 file1
-rw-r--r--. 1 root root 0  1월  4 20:57 file5
-rw-r--r--. 1 root root 0  1월  4 20:57 file6
-rw-r--r--. 1 root root 0  1월  4 20:57 file7
-rw-r--r--. 1 root root 0  1월  4 20:57 file8
-rw-r--r--. 1 root root 0  1월  4 20:57 file9
[root@localhost dir1]# cp file1 copy
[root@localhost dir1]# cd copy
[root@localhost copy]# ll
합계 0
-rw-r--r--. 1 root root 0  1월  4 21:16 file1
[root@localhost copy]#
```

```
[root@localhost copy]# cp /etc/passwd .
[root@localhost copy]# ll
합계 4
-rw-r--r--. 1 root root  0  1월  4 21:16 file1
-rw-r--r--. 1 root root 1956 1월  4 21:18 passwd
[root@localhost copy]#
```

```
[root@localhost copy]# cp /etc/passwd /root/dir1
[root@localhost copy]# cd ..
[root@localhost dir1]# ll
합계 4
drwxr-xr-x. 2 root root  33 1월  4 21:18 copy
-rw-r--r--. 1 root root   0 1월  4 20:55 file1
-rw-r--r--. 1 root root   0 1월  4 20:57 file5
-rw-r--r--. 1 root root   0 1월  4 20:57 file6
-rw-r--r--. 1 root root   0 1월  4 20:57 file7
-rw-r--r--. 1 root root   0 1월  4 20:57 file8
-rw-r--r--. 1 root root   0 1월  4 20:57 file9
-rw-r--r--. 1 root root 1956 1월  4 21:18 passwd
[root@localhost dir1]#
```

💡 cp 고급 활용

```
# 백업 생성
cp file.txt file.txt.$(date +%Y%m%d)

# 여러 파일을 한 디렉토리로 복사
cp file1 file2 file3 destination_dir/

# 와일드카드 사용
cp *.txt backup_dir/
cp src/*.{c,h} dest_dir/

# 심볼릭 링크 보존
```

```
cp -d link_file new_location/
```

```
# 안전한 복사 (덮어쓰기 방지)
```

```
cp -n source dest
```

파일/디렉토리 이동

mv [원본] [대상]: 파일이나 디렉토리를 이동하거나 이름을 변경하는 명령어이다.

mv의 두 가지 기능

1. 파일/디렉토리 이동
2. 이름 변경

```
# 파일 이동
```

```
mv file1.txt documents/
```

```
mv file1.txt /home/user/backup/
```

```
# 디렉토리 이동
```

```
mv old_directory new_location/
```

```
# 이름 변경 (같은 디렉토리 내)
```

```
mv old_name.txt new_name.txt
```

```
mv old_dir new_dir
```

```
# 이동과 이름 변경 동시에
```

```
mv file.txt /home/user/new_file.txt
```

mv 주요 옵션

옵션	기능	사용 예시
-i	덮어쓰기 확인	mv -i file1 file2
-u	최신 파일만 이동	mv -u file1 file2
-v	상세 출력	mv -v file1 file2
-n	덮어쓰기 금지	mv -n file1 file2
-b	백업 생성	mv -b file1 file2

```

drwxr-xr-x. 2 root root 33 1월 4 21:18 copy
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
-rw-r--r--. 1 root root 1956 1월 4 21:18 passwd
[root@localhost dir1]# vi passwd
[root@localhost dir1]# mv passwd file1
mv: overwrite 'file1'? y
[root@localhost dir1]# vi passwd
[root@localhost dir1]# cat -n passwd
cat: passwd: 그런 파일이나 디렉터리가 없습니다
[root@localhost dir1]# cat -n file1
 1 root:x:0:0:root:/root:/bin/bash
 2 bin:x:1:1:bin:/bin:/sbin/nologin
 3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
 4 adm:x:3:4:adm:/var/adm:/sbin/nologin
 5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

```

💡 mv 활용 팁

```

# 대량 파일 이름 변경
for file in *.txt; do
    mv "$file" "${file%.txt}.backup"
done

# 확장자 변경
mv document.txt document.md

# 안전한 이동 (백업과 함께)
mv -b important_file.txt new_location/

# 여러 파일을 한 번에 이동
mv file1 file2 file3 destination_dir/

```

📖 파일 내용 확인

cat [옵션] [파일명]: 파일의 내용을 출력하거나 파일을 생성하는 다목적 명령어이다.

🔧 기본 사용법

```

# 파일 내용 출력
cat filename.txt

# 여러 파일 내용 출력
cat file1.txt file2.txt

# 행 번호와 함께 출력
cat -n filename.txt

```



```
# 빈 행 제외하고 행 번호
cat -b filename.txt
```

cat 주요 옵션

옵션	기능	사용 예시
-n	행 번호 표시	cat -n file.txt
-b	빈 행 제외 행 번호	cat -b file.txt
-s	연속 빈 행 압축	cat -s file.txt
-T	탭 문자 표시	cat -T file.txt
-E	행 끝 표시	cat -E file.txt
-A	모든 제어 문자 표시	cat -A file.txt

리다이렉션 활용

```
# 파일 생성 (덮어쓰기)
cat > file100.txt
# 내용 입력 후 Ctrl + D로 저장

# 파일 추가 (이어쓰기)
cat >> file100.txt
# 내용 입력 후 Ctrl + D로 저장

# 여러 파일 병합
cat file1.txt file2.txt > merged_file.txt

# 파일 끝에 다른 파일 추가
cat additional.txt >> main_file.txt
```

```
drwxr-xr-x. 2 root root 33 1월 4 21:18 copy
-rw-r--r--. 1 root root 0 1월 4 20:55 file1
-rw-r--r--. 1 root root 0 1월 4 20:57 file5
-rw-r--r--. 1 root root 0 1월 4 20:57 file6
-rw-r--r--. 1 root root 0 1월 4 20:57 file7
-rw-r--r--. 1 root root 0 1월 4 20:57 file8
-rw-r--r--. 1 root root 0 1월 4 20:57 file9
-rw-r--r--. 1 root root 1956 1월 4 21:18 passwd
[root@localhost dir1]# vi passwd
[root@localhost dir1]# mv passwd file1
mv: overwrite 'file1'? y
[root@localhost dir1]# vi passwd
[root@localhost dir1]# cat -n passwd
cat: passwd: 그런 파일이나 디렉터리가 없습니다
[root@localhost dir1]# cat -n file1
 1 root:x:0:0:root:/root:/bin/bash
 2 bin:x:1:1:bin:/bin:/sbin/nologin
 3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
 4 adm:x:3:4:adm:/var/adm:/sbin/nologin
 5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
root@localhost:~/dir1
[root@localhost dir1]# cat file1 >> file2
[root@localhost dir1]# cat -n file2
 1 root:x:0:0:root:/root:/bin/bash
 2 bin:x:1:1:bin:/bin:/sbin/nologin
 3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
 4 adm:x:3:4:adm:/var/adm:/sbin/nologin
 5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
 6 sync:x:5:0:sync:/sbin:/bin/sync
 7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
 8 halt:x:7:0:halt:/sbin:/sbin/halt
 9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10 operator:x:11:0:operator:/root:/sbin/nologin
11 games:x:12:100:games:/usr/games:/sbin/nologin
12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
13 nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
14 tss:x:59:59:Account used for TPM access:/:/usr/sbin/nologin
15 systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
16 dbus:x:81:81:System message bus:/:/sbin/nologin
17 polkitd:x:998:996:User for polkitd:/:/sbin/nologin
18 sssd:x:997:995:User for sssd:/:/sbin/nologin
19 avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
20 geoclue:x:996:994:User for geoclue:/var/lib/geoclue:/sbin/nologin
21 rtkit:x:172:172:RealtimeKit:/:/sbin/nologin
```

```
[root@localhost dir1]# cat -n file3
 1 root:x:0:0:root:/root:/bin/bash
 2 bin:x:1:1:bin:/bin:/sbin/nologin
 3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
 4 adm:x:3:4:adm:/var/adm:/sbin/nologin
 5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
 6 sync:x:5:0:sync:/sbin:/bin/sync
 7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
 8 halt:x:7:0:halt:/sbin:/sbin/halt
 9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10 operator:x:11:0:operator:/root:/sbin/nologin
11 games:x:12:100:games:/usr/games:/sbin/nologin
12 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
13 nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
14 tss:x:59:59:Account used for TPM access:/:/usr/sbin/nologin
15 systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
16 dbus:x:81:81:System message bus:/:/sbin/nologin
17 polkitd:x:998:996:User for polkitd:/:/sbin/nologin
```

```
[root@localhost dir1]# touch file9
[root@localhost dir1]# cat > file9
Hello World
Linux!
Window11![root@localhost dir1]#
[root@localhost dir1]#
[root@localhost dir1]# cat -n file9
 1 Hello World
 2 Linux!
 3 Window11![root@localhost dir1]#
```

💡 cat 고급 활용

```
# Here Document 사용
cat << EOF > config.txt
```

```
server_name=localhost
port=8080
debug=true
EOF
```

파일 내용에 번호 추가하여 새 파일 생성

```
cat -n original.txt > numbered.txt
```

여러 파일의 특정 부분만 결합

```
head -5 file1.txt > temp.txt
```

```
tail -3 file2.txt >> temp.txt
```

```
cat temp.txt
```

페이지 단위 출력

대용량 파일을 효율적으로 확인하기 위한 명령어들이다.

head 명령어

head [옵션] [파일명]: 파일의 처음 부분을 출력하는 명령어이다.

기본 (처음 10줄)

```
head filename.txt
```

처음 5줄 출력

```
head -5 passwd
```

```
head -n 5 passwd
```

여러 파일의 처음 부분

```
head file1.txt file2.txt
```

바이트 단위로 출력

```
head -c 100 filename.txt
```

tail 명령어

tail [옵션] [파일명]: 파일의 마지막 부분을 출력하거나 실시간으로 추가되는 내용을 확인하는 명령어이다.

기본 (마지막 10줄)

```
tail filename.txt
```

마지막 5줄 출력

```
tail -5 passwd
```

```
tail -n 5 passwd
```

실시간 모니터링 (매우 유용!)

```
tail -f file2.txt          # 종료: Ctrl + C
```

```
tail -f /var/log/messages # 로그 실시간 확인
```

```
# 여러 파일 동시 모니터링
```

```
tail -f file1.txt file2.txt
```

```
# 바이트 단위로 출력
```

```
tail -c 100 filename.txt
```

head/tail 옵션 비교

명령어	옵션	기능	사용 예시
head	-n 숫자	처음 n줄 출력	head -n 20 file.txt
head	-c 숫자	처음 n바이트 출력	head -c 500 file.txt
tail	-n 숫자	마지막 n줄 출력	tail -n 20 file.txt
tail	-c 숫자	마지막 n바이트 출력	tail -c 500 file.txt
tail	-f	실시간 추가 내용 출력	tail -f /var/log/syslog
tail	-F	파일 교체 시에도 추적	tail -F rotating.log

head/tail 고급 활용

```
# 파일의 특정 범위 출력 (11-20줄)
```

```
head -20 file.txt | tail -10
```

```
# 여러 파일에서 처음과 마지막 확인
```

```
head -5 *.txt
```

```
tail -5 *.txt
```

```
# 로그 파일 실시간 분석
```

```
tail -f /var/log/apache2/access.log | grep "ERROR"
```

```
# 대용량 파일의 중간 부분 확인
```

```
head -1000 large_file.txt | tail -100
```

```
# 파일 크기가 변하는 것 감지
```

```
tail -f growing_file.txt --pid=$(pgrep process_name)
```

기타 유용한 페이지 출력 명령어

more 명령어

```
# 페이지 단위로 출력 (기본)
```

```
more filename.txt
```

```
# 스페이스바: 다음 페이지
```

```
# Enter: 한 줄씩  
# q: 종료
```

less 명령어 (more의 개선 버전)

```
# 양방향 스크롤 가능  
less filename.txt  
  
# 유용한 less 단축키  
# 스페이스바/f: 다음 페이지  
# b: 이전 페이지  
# /검색어: 앞으로 검색  
# ?검색어: 뒤로 검색  
# g: 파일 처음으로  
# G: 파일 끝으로  
# q: 종료
```

명령어 조합 및 실무 활용

파이프라인 활용

```
# 파일 목록을 페이지별로 확인  
ls -la | less  
  
# 큰 디렉토리에서 최근 파일 찾기  
ls -lt | head -10  
  
# 프로세스 목록에서 특정 프로세스 찾기  
ps aux | grep apache  
  
# 디스크 사용량 상위 10개 디렉토리  
du -sh */ | sort -hr | head -10
```

검색 및 필터링

```
# 특정 확장자 파일만 보기  
ls *.txt | head -5  
  
# 숨김 파일 제외하고 보기  
ls -A | grep -v "^\."
```

```
# 큰 파일들 찾기  
ls -lah | grep "^-" | sort -k5 -hr  
  
# 오늘 수정된 파일 찾기  
find . -type f -newermt $(date +%Y-%m-%d) | head -10
```

시스템 모니터링

```
# 로그 실시간 모니터링
tail -f /var/log/syslog | grep -i error

# 시스템 리소스 모니터링
watch -n 1 'df -h | head -5'

# 메모리 사용량 실시간 확인
watch -n 2 'free -h'

# 네트워크 연결 상태 모니터링
watch -n 3 'netstat -tuln | head -10'
```

학습 정리 및 체크리스트

필수 명령어 마스터 체크리스트

디렉토리 구조 이해

- ☐ 주요 디렉토리 역할 숙지 (/etc , /home , /var 등)
- ☐ 절대경로와 상대경로 구분
- ☐ FHS(Filesystem Hierarchy Standard) 이해

파일/디렉토리 관리

- ☐ ls 옵션 활용 (-la , -ltr , -lah)
- ☐ mkdir 중첩 디렉토리 생성 (-p)
- ☐ touch 파일 생성 및 타임스탬프 조작
- ☐ rm / rmdir 안전한 삭제 방법
- ☐ cp 복사 옵션 이해 (-r , -p , -a)
- ☐ mv 이동 및 이름 변경

파일 내용 관리

- ☐ cat 기본 사용법 및 리다이렉션
- ☐ head / tail 페이지 단위 출력
- ☐ tail -f 실시간 모니터링

시간 및 시스템

- ☐ date 날짜/시간 출력 및 형식화
- ☐ cal 달력 출력 옵션

실무 활용 시나리오

로그 분석 워크플로우

```
# 1. 로그 파일 위치 확인
ls -la /var/log/

# 2. 로그 파일 크기 확인
ls -lah /var/log/*.log

# 3. 최근 에러 로그 확인
tail -100 /var/log/syslog | grep -i error

# 4. 실시간 로그 모니터링
tail -f /var/log/syslog
```

프로젝트 디렉토리 설정

```
# 1. 프로젝트 구조 생성
mkdir -p myproject/{src,docs,tests,config}

# 2. 기본 파일 생성
touch myproject/README.md
touch myproject/src/{main.py,utils.py}
touch myproject/tests/test_main.py

# 3. 구조 확인
tree myproject/ # tree 명령어가 있는 경우
# 또는
ls -la myproject/*/
```

백업 및 정리 작업

```
# 1. 백업 디렉토리 생성
mkdir -p backup/$(date +%Y%m%d)

# 2. 중요 파일 백업
cp -a important_files/* backup/$(date +%Y%m%d)/

# 3. 오래된 파일 정리 (30일 이상)
find . -type f -mtime +30 -exec ls -la {} \;

# 4. 임시 파일 정리
rm -f *.tmp *.bak *~
```

 다음 단계 학습 가이드

 중급 명령어 학습 추천

1. 텍스트 처리: `grep`, `sed`, `awk`, `sort`, `uniq`
2. 파일 권한: `chmod`, `chown`, `chgrp`, `umask`
3. 압축/해제: `tar`, `gzip`, `zip`, `unzip`
4. 프로세스 관리: `ps`, `top`, `htop`, `kill`, `jobs`
5. 네트워크: `ping`, `wget`, `curl`, `ssh`, `scp`

고급 기능 탐구

1. 정규표현식: 패턴 매칭 및 검색
2. 스크립팅: Bash 스크립트 작성
3. 시스템 관리: `systemctl`, `crontab`, `mount`
4. 성능 모니터링: `iostat`, `vmstat`, `sar`

유용한 팁과 트릭

키보드 단축키

```
# 명령어 편집
Ctrl + A    # 줄 처음으로
Ctrl + E    # 줄 끝으로
Ctrl + K    # 커서부터 끝까지 삭제
Ctrl + U    # 커서부터 처음까지 삭제
Ctrl + W    # 단어 삭제

# 히스토리 활용
!!          # 이전 명령어 실행
!n          # n번째 명령어 실행
!str        # str로 시작하는 최근 명령어 실행
Ctrl + R    # 명령어 검색
```

별칭(Alias) 설정

```
# 자주 사용하는 명령어 단축
alias ll='ls -la'
alias la='ls -A'
alias l='ls -CF'
alias ..='cd ..'
alias ...='cd ../..'
alias grep='grep --color=auto'

# 안전한 삭제 설정
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```



```
# 별칭 영구 저장 (~/.bashrc에 추가)
echo "alias ll='ls -la'" >> ~/.bashrc
```

💡 **학습 팁:** 각 명령어를 실제로 실행해보면서 옵션들을 조합해보는 것이 가장 효과적인 학습 방법이다.

⚠️ **주의사항:** `rm -rf` 명령어는 복구가 불가능하므로 항상 신중하게 사용해야 한다. 중요한 파일은 미리 백업하는 습관을 기르자.

🔍 **추가 정보:** 각 명령어의 상세한 옵션은 `man 명령어` 또는 `명령어 --help`로 확인할 수 있다. `ls -laS` # 숨김 파일 + 상세 정보 + 크기 순

📁 ls -l 출력 정보 해석

```
```bash
-rw-r--r-- 1 root root 1234 Jan 15 10:30 example.txt
drwxr-xr-x 2 user user 4096 Jan 14 15:20 documents
```

위치	의미	예시	설명
1번째	파일 타입	<code>-</code> , <code>d</code> , <code>l</code>	일반파일, 디렉토리, 링크
2-4번째	소유자 권한	<code>rw</code> <code>x</code>	읽기, 쓰기, 실행
5-7번째	그룹 권한	<code>r</code> <code>-</code> <code>x</code>	읽기, 실행
8-10번째	기타 권한	<code>r</code> <code>-</code> <code>-</code>	읽기만
다음	링크 수	<code>1</code> , <code>2</code>	하드링크 개수
다음	소유자	<code>root</code> , <code>user</code>	파일 소유자
다음	그룹	<code>root</code> , <code>user</code>	파일 그룹
다음	크기	<code>1234</code>	바이트 단위
다음	수정 시간	<code>Jan 15 10:30</code>	마지막 수정 시간
마지막	파일명	<code>example.txt</code>	파일/디렉토리명

## 📁 디렉토리 이동 명령어

디렉토리 간 이동을 위한 `cd` 명령어 활용법이다.

```
절대 경로 이동
cd /
cd /usr/local
cd /home/user

루트 디렉토리 (최상위)
절대 경로로 이동
특정 사용자 홈으로 이동

상대 경로 이동
```

```
cd # 홈 디렉토리
cd ~ # 홈 디렉토리 (명시적)
cd ~/Documents # 홈의 Documents 폴더
cd .. # 상위 디렉토리
cd ../.. # 두 단계 상위 디렉토리
cd - # 이전 디렉토리 (직전 위치)

특수 디렉토리
cd /tmp # 임시 디렉토리
cd /var/log # 로그 디렉토리
cd /etc # 설정 파일 디렉토리
```

## 디렉토리 이동 활용 팁

```
현재 위치 확인
pwd # 현재 경로 출력

디렉토리 스택 활용
pushd /var/log # 디렉토리 스택에 저장하며 이동
popd # 스택에서 이전 디렉토리로 복귀
dirs # 스택 내용 확인

자주 사용하는 디렉토리 바로가기
alias ll='ls -la' # 별칭 설정
alias ..='cd ..' # 상위 디렉토리 바로가기
alias ...='cd ../..' # 두 단계 상위 디렉토리
```

## 추가 파일 관리 명령어

```
디렉토리 생성
mkdir dirname # 디렉토리 생성
mkdir -p path/to/dir # 중간 경로까지 생성

파일 생성 및 수정
touch filename # 빈 파일 생성 또는 시간 업데이트
nano filename # 텍스트 에디터로 파일 편집
vim filename # Vim 에디터로 파일 편집

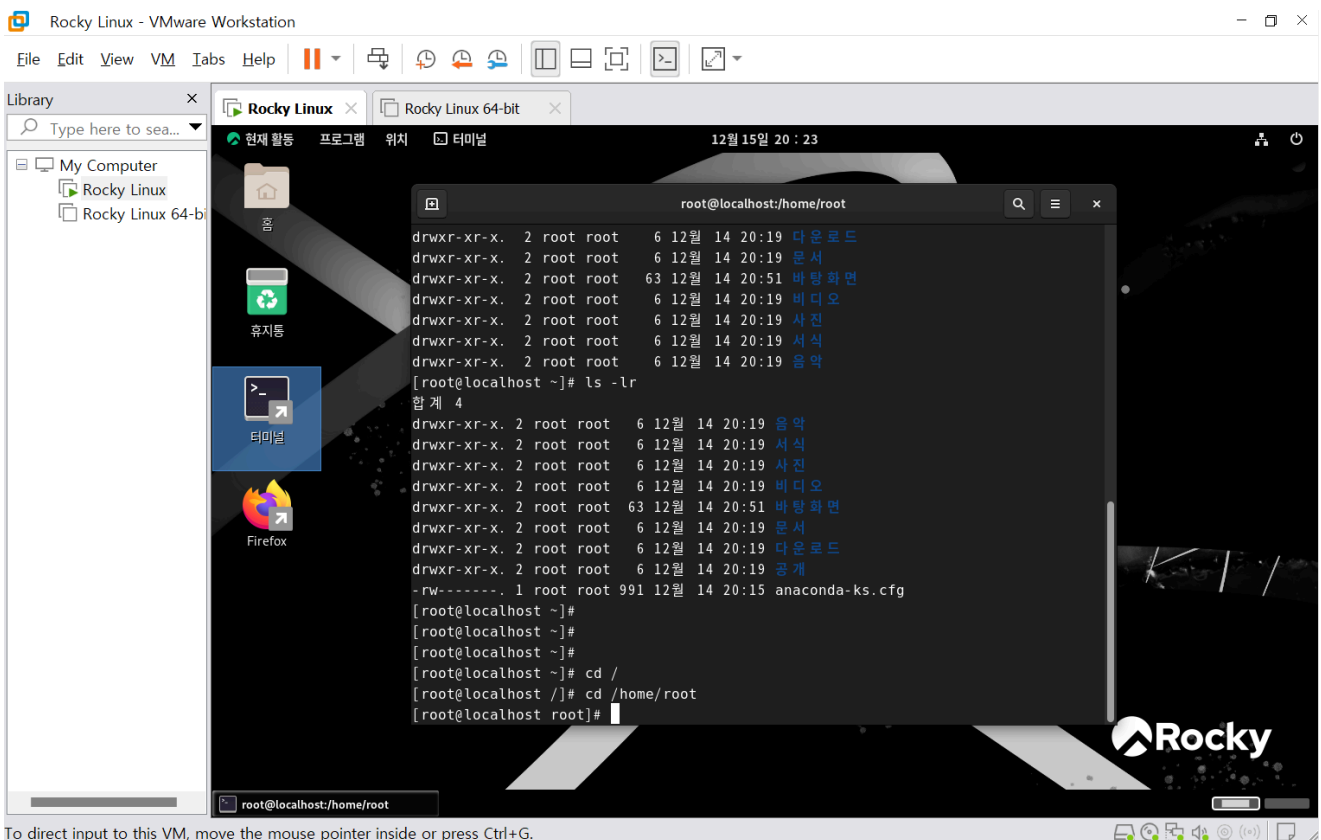
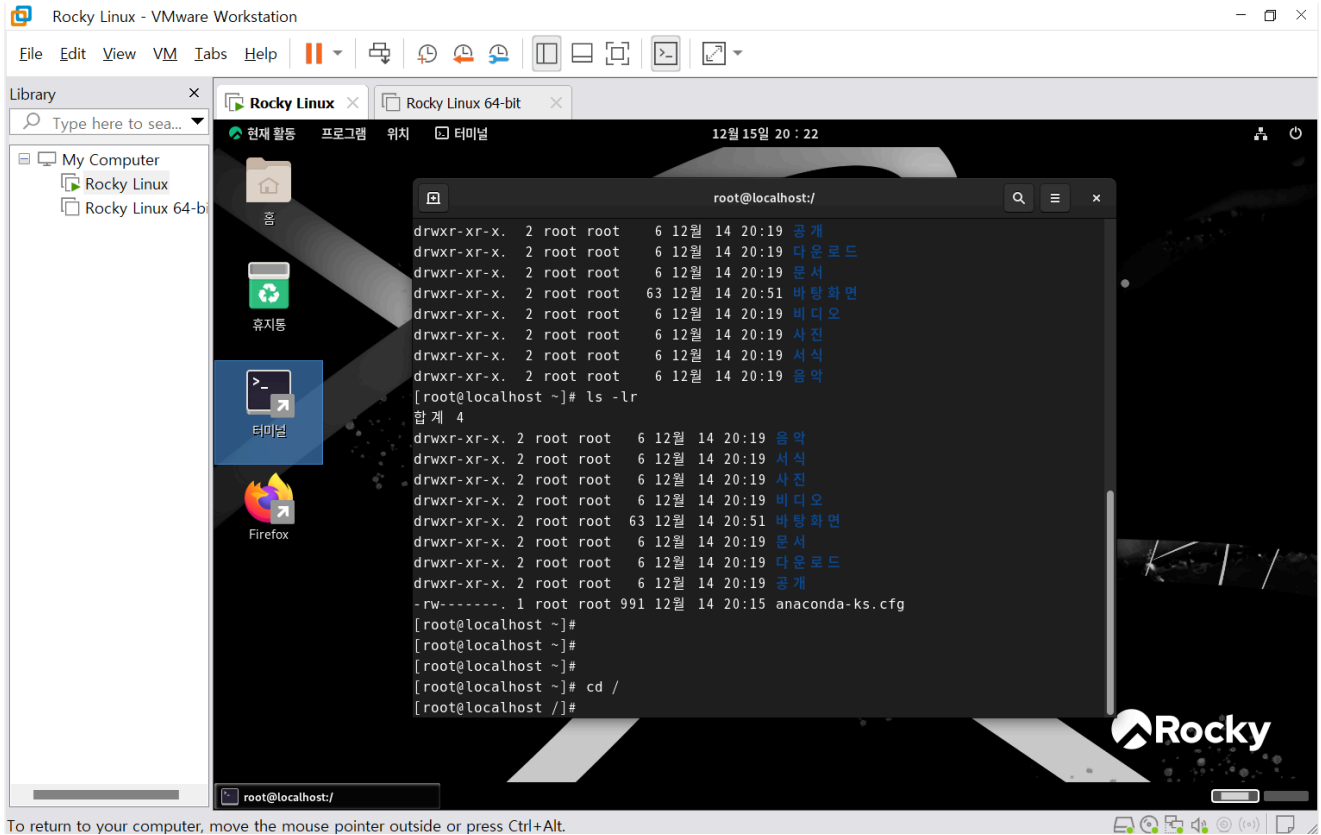
파일 복사 및 이동
cp source dest # 파일 복사
cp -r source dest # 디렉토리 복사
mv source dest # 파일 이동/이름 변경

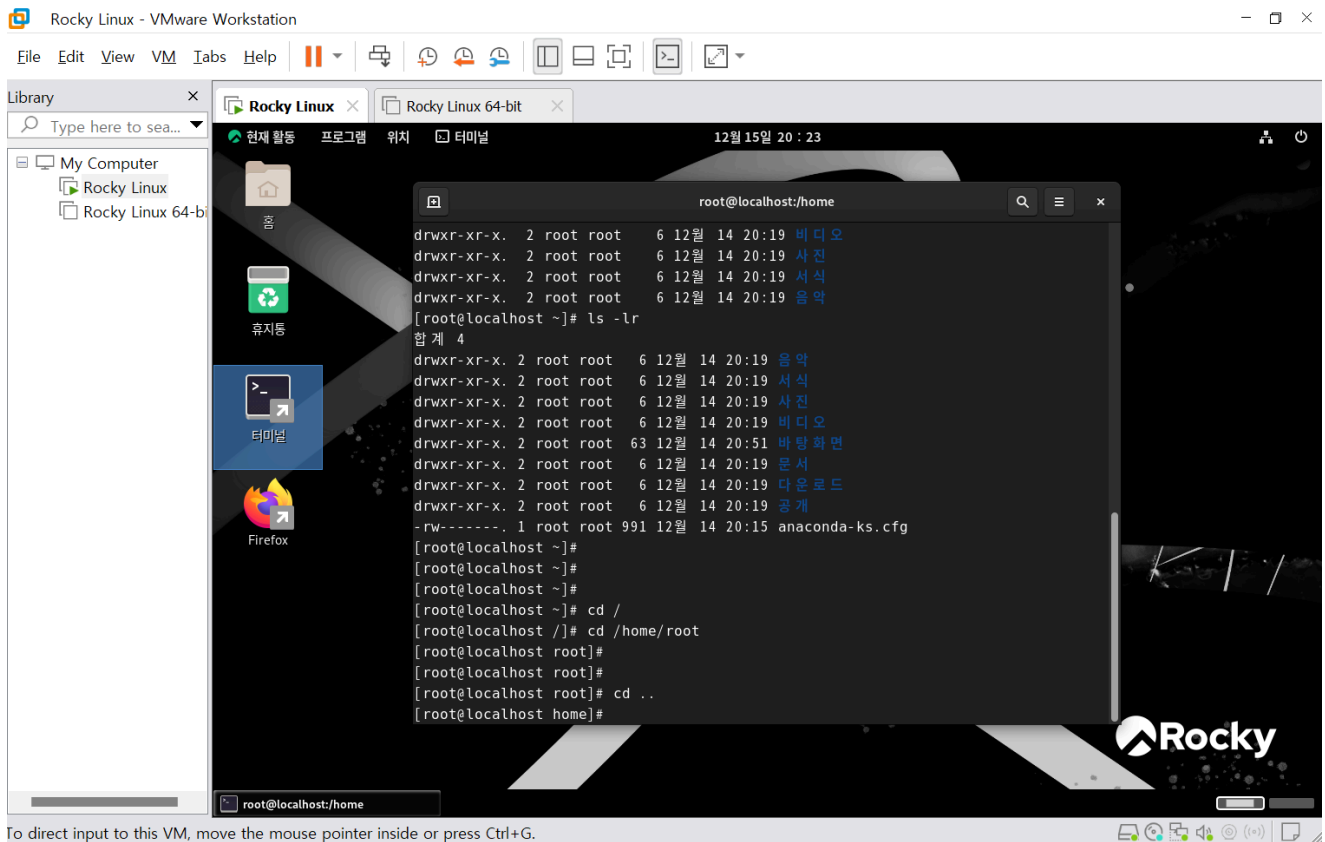
파일 삭제
rm filename # 파일 삭제
rm -r dirname # 디렉토리 삭제
rm -rf dirname # 강제 삭제 (주의!)
```

## GUI 환경 설정

## GNOME 환경 설정 이미지

다음 이미지들은 GNOME 환경에서의 설정 과정을 보여준다:





## 🔧 GUI 환경 최적화 팁

### 터미널 설정

```
터미널 투명도 및 테마 설정
gsettings set
org.gnome.Terminal.Legacy.Profile:/org/gnome/terminal/legacy/profiles:/:$(gsetting
gs get org.gnome.Terminal.ProfilesList default | tr -d '\')/ use-transparent-
background true
gsettings set
org.gnome.Terminal.Legacy.Profile:/org/gnome/terminal/legacy/profiles:/:$(gsetting
gs get org.gnome.Terminal.ProfilesList default | tr -d '\')/ background-
transparency-percent 10
```

### 바탕화면 설정

```
바탕화면 이미지 변경
gsettings set org.gnome.desktop.background picture-uri
file:///path/to/your/wallpaper.jpg

다크 모드 활성화
gsettings set org.gnome.desktop.interface gtk-theme 'Adwaita-dark'
```

### 시스템 모니터링

```
시스템 리소스 확인
htop

대화형 프로세스 뷰어
```

<code>free -h</code>	# 메모리 사용량
<code>df -h</code>	# 디스크 사용량
<code>lscpu</code>	# CPU 정보

## 학습 요약


이 가이드에서 다룬 주요 내용들이다:


### 핵심 명령어 체크리스트

- ☐ 시스템 제어: `shutdown`, `reboot`, `systemctl`
- ☐ 파일 관리: `ls`, `cd`, `pwd`, `mkdir`, `cp`, `mv`, `rm`
- ☐ 권한 관리: `su`, `sudo`, `chmod`, `chown`
- ☐ 시스템 정보: `whoami`, `id`, `free`, `df`
- ☐ 프로세스 관리: `ps`, `top`, `htop`, `kill`

### 다음 단계 학습 추천

- 텍스트 에디터: `nano`, `vim`, `emacs` 사용법
- 파일 권한: `chmod`, `chown` 상세 학습
- 프로세스 관리: `systemctl`, `service` 명령어
- 네트워크 설정: `ip`, `netstat`, `ss` 명령어
- 패키지 관리: `dnf`, `rpm` 고급 사용법

 **팁:** 명령어 사용 시 `man` 명령어 또는 `명령어 --help` 를 사용하여 상세한 도움말을 확인할 수 있다.

 **주의:** `rm -rf` 명령어는 복구가 불가능한 삭제를 수행하므로 신중하게 사용해야 한다.

[Linux 1.1 - 로키](#)