

# Linux 2.1

## 리눅스 기본 명령어 완전 정리

### 리눅스 디렉토리 구조 및 역할

리눅스 시스템의 디렉토리 구조는 계층적으로 구성되어 있으며, 각 디렉토리는 특정한 역할을 수행한다.

### 주요 디렉토리 구조

디렉토리	역할	설명
/boot	부팅 관련 파일	커널 이미지와 부트로더 설정 파일이 저장되는 디렉토리이다
/etc	시스템 설정 파일	모든 환경 설정 파일과 시스템 구성 파일이 저장되는 디렉토리이다
/lib	동적 라이브러리 (32 비트)	시스템 프로그램이 실행될 때 필요한 공유 라이브러리 파일이 저장되는 디렉토리이다
/lib64	동적 라이브러리 (64 비트)	64비트 시스템용 공유 라이브러리 파일이 저장되는 디렉토리이다
/root	root 홈 디렉토리	시스템 관리자(root) 계정의 개인 디렉토리이다
/sbin	시스템 관련 명령어	시스템 관리에 필요한 명령어들이 저장되는 디렉토리이다
/usr	사용자 프로그램	일반 사용자 관련 명령어와 프로그램이 설치되는 디렉토리이다
/bin	기본 명령어	모든 사용자가 사용할 수 있는 기본 명령어들이 저장되는 디렉토리이다
/dev	장치 파일	하드웨어 장치에 대한 인터페이스 파일들이 저장되는 디렉토리이다
/home	사용자 홈 디렉토리	일반 사용자들의 개인 디렉토리가 생성되는 디렉토리이다
/mnt	마운트 포인트	임시로 파일시스템을 마운트할 때 사용되는 디렉토리이다
/proc	가상 파일 시스템	커널과 프로세스 정보를 제공하는 가상 파일 시스템이다
/tmp	임시 파일	시스템과 사용자가 생성하는 임시 파일들이 저장되는 디렉토리이다
/var	가변 데이터	시스템 운영 중 변경되는 파일들(로그, 스펠 등)이 저장되는 디렉토리이다

## 리눅스 기본 명령어

### ls - 파일 및 디렉토리 목록 출력

ls 명령어는 현재 디렉토리나 지정된 디렉토리의 파일과 디렉토리 목록을 출력하는 명령어이다.

### 주요 옵션

옵션	설명
-l	파일의 상세 정보를 포함하여 출력한다
-a	숨김 파일과 디렉토리까지 모두 출력한다
-A	숨김 파일과 디렉토리를 출력하되 . 과 .. 은 제외한다
-h	파일 크기를 사람이 읽기 쉬운 형태로 출력한다
-t	수정 시간 순으로 정렬하여 출력한다

---

## mkdir - 디렉토리 생성

`mkdir` 명령어는 새로운 디렉토리를 생성하는 명령어이다.

### 사용 예제

```
# 단일 디렉토리 생성
mkdir dir1

# 여러 디렉토리 동시 생성
mkdir dir1 dir2 dir3

# 중괄호 확장을 이용한 디렉토리 생성
mkdir dir{5..9}

# 상위 디렉토리도 함께 생성 (-p 옵션)
mkdir -p dir2/dir3/dir4

# 숨김 디렉토리 생성
mkdir .dir100
```

```
root@localhost:~  
합 계 4  
-rw-----. 1 root root 991 12월 14 20:15 anaconda-ks.cfg  
drwxr-xr-x. 2 root root 6 12월 14 20:19 공 개  
drwxr-xr-x. 2 root root 6 12월 14 20:19 다 운 로 드  
drwxr-xr-x. 2 root root 6 12월 14 20:19 문 서  
drwxr-xr-x. 2 root root 63 12월 14 20:51 바 탕 화 면  
drwxr-xr-x. 2 root root 6 12월 14 20:19 비 디 오  
drwxr-xr-x. 2 root root 6 12월 14 20:19 사 진  
drwxr-xr-x. 2 root root 6 12월 14 20:19 서 식  
drwxr-xr-x. 2 root root 6 12월 14 20:19 음 악  
[root@localhost ~]# mkdir dir1  
[root@localhost ~]# ll  
합 계 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 공 개  
drwxr-xr-x. 2 root root 6 12월 14 20:19 다 운 로 드  
drwxr-xr-x. 2 root root 6 12월 14 20:19 문 서  
drwxr-xr-x. 2 root root 63 12월 14 20:51 바 탕 화 면  
drwxr-xr-x. 2 root root 6 12월 14 20:19 비 디 오  
drwxr-xr-x. 2 root root 6 12월 14 20:19 사 진
```

```
root@localhost:~  
[root@localhost dir1]# touch file1  
[root@localhost dir1]# ll  
합 계 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  
합 계 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 공 개  
drwxr-xr-x. 2 root root 6 12월 14 20:19 다 운 로 드  
[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]#
```

## 주요 옵션

옵션	설명
-p	상위 디렉토리가 없으면 함께 생성한다
-m	디렉토리 생성 시 권한을 설정한다

---

## touch - 파일 생성 및 시간 변경

`touch` 명령어는 빈 파일을 생성하거나 기존 파일의 수정 시간을 변경하는 명령어이다.

## 사용 예제

```
# 빈 파일 생성
touch file1

# 여러 파일 동시 생성
touch file1 file2 file3

# 중괄호 확장을 이용한 파일 생성
touch file{1..3}

# 특정 시간으로 파일 생성/수정
touch -t 202412251530 file1

# 현재 시간으로 파일 시간 변경
touch existing_file
```

```
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]#

```

## 숨김 파일 생성 방법

숨김 파일이나 디렉토리를 생성할 때는 이름 앞에 마침표( . )를 붙인다.

```

touch .hidden_file
mkdir .hidden_directory

```

## rm/rmdir - 파일 및 디렉토리 삭제

파일과 디렉토리를 삭제하는 명령어들이다.

### rm 명령어

```

# 파일 삭제
rm file1

# 여러 파일 동시 삭제
rm file1 file2 file3

# 패턴을 이용한 삭제
rm file{1..5}
rm f*

# 디렉토리 와 내용 모두 삭제
rm -rf directory_name

```

# 확인 없이 강제 삭제

```
rm -f file_name
```

## rmdir 명령어

# 빈 디렉토리만 삭제 가능

```
rmdir empty_directory
```

```
root@localhost:~/dir1
[root@localhost dir1]# ll
합 계 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
합 계 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
합 계 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
```

## 숨김 파일 삭제

숨김 파일은 개별적으로 삭제해야 한다.



```
rm -rf .hidden_file
```

## ⚠ 주요 옵션

옵션	설명
-r	디렉토리와 하위 내용을 재귀적으로 삭제한다
-f	확인 메시지 없이 강제로 삭제한다
-i	삭제 전에 확인 메시지를 출력한다

## 🕒 date - 날짜 및 시간 출력

date 명령어는 현재 시스템의 날짜와 시간을 출력하거나 설정하는 명령어이다.

## 💡 사용 예제

```
# 현재 날짜와 시간 출력
date

# UTC 시간 출력
date -u

# 특정 형식으로 출력
date "+%Y-%m-%d %H:%M:%S"

# 특정 날짜 출력
date -d "2024-12-25"
```

```
[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]#
```

## 📌 주요 옵션

옵션	설명
-u	UTC 시간을 출력한다
-d	특정 날짜를 지정하여 출력한다
+format	사용자 정의 형식으로 출력한다



## cal - 달력 출력

cal 명령어는 달력을 출력하는 명령어이다.



### 사용 예제

```
# 현재 월 달력 출력
```

```
cal
```

```
# 3개월 달력 출력 (이전, 현재, 다음 월)
```

```
cal -3
```

```
# 전체 년도 달력 출력
```

```
cal -y
```

```
# 특정 년도 달력 출력
```

```
cal -y 2024
```

```
# 특정 월 달력 출력
```

```
cal 12 2024
```

```
# 줄리안 형식으로 출력
```

```
cal -yj
```

```
[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
```

## 📌 주요 옵션

옵션	설명
-3	3개월 달력을 출력한다
-y	전체 년도 달력을 출력한다
-j	줄리안 형식으로 출력한다

## 📄 cp - 파일 및 디렉토리 복사

cp 명령어는 파일이나 디렉토리를 복사하는 명령어이다.

## 💡 사용 예제

```
# 파일 복사
cp file1 file2

# 파일을 디렉토리에 복사
cp file1 dir1/

# 여러 파일을 디렉토리에 복사
cp file1 file2 file3 dir1/

# 시스템 파일 복사
cp /etc/passwd .

# 디렉토리 복사 (재귀적)
cp -r dir1 dir2

# 권한과 소유자 정보 유지하며 복사
cp -p 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]#
```

## 📌 주요 옵션

옵션	설명
-r	디렉토리를 재귀적으로 복사한다
-p	원본 파일의 권한과 소유자 정보를 유지한다
-i	덮어쓰기 전에 확인 메시지를 출력한다
-u	대상 파일보다 새로운 파일만 복사한다

## 🔄 mv - 파일 및 디렉토리 이동/이름 변경

**mv** 명령어는 파일이나 디렉토리를 이동하거나 이름을 변경하는 명령어이다.

## 💡 사용 예제

```
# 파일 이름 변경
mv old_name new_name

# 파일을 디렉토리로 이동
mv file1 dir1/

# 여러 파일을 디렉토리로 이동
mv file1 file2 file3 dir1/

# 디렉토리 이름 변경
mv old_dir new_dir

# 디렉토리를 다른 디렉토리로 이동
mv dir1 dir2/
```

```

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

```

## 📌 주요 옵션

옵션	설명
-i	덮어쓰기 전에 확인 메시지를 출력한다
-f	확인 없이 강제로 이동한다
-u	대상 파일보다 새로운 파일만 이동한다

## 📖 cat - 파일 내용 출력 및 편집

`cat` 명령어는 파일 내용을 출력하거나 파일을 생성하는 명령어이다.

## 💡 사용 예제

```

# 파일 내용 출력
cat file1.txt

# 행 번호와 함께 출력
cat -n file1.txt

# 빈 행을 제외하고 행 번호 출력
cat -b file1.txt

# 여러 파일 내용 출력
cat file1.txt file2.txt

# 파일 생성 (입력 모드)
cat > new_file.txt

# 내용 입력 후 Ctrl+D로 저장

```

```
# 파일 내용 덮어쓰기
cat > existing_file.txt

# 파일 내용 이어쓰기
cat >> existing_file.txt

# 여러 파일 내용을 하나의 파일로 병합
cat file1.txt file2.txt > merged_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]#
```

## 리다이렉션 연산자

연산자	설명
>	출력을 파일로 덮어쓴다
>>	출력을 파일에 이어쓴다
<	파일에서 입력을 받는다

## 주요 옵션

옵션	설명
-n	모든 줄에 행 번호를 출력한다
-b	빈 줄을 제외하고 행 번호를 출력한다
-s	연속된 빈 줄을 하나로 압축한다

## head/tail - 파일 일부 내용 출력

파일의 앞부분이나 뒷부분을 출력하는 명령어들이다.



## 📌 head 명령어

파일의 앞부분을 출력한다.

```
# 파일의 처음 10줄 출력 (기본값)
head filename

# 파일의 처음 5줄 출력
head -5 filename
head -n 5 filename

# 파일의 처음 100바이트 출력
head -c 100 filename

# 여러 파일의 처음 부분 출력
head file1 file2 file3
```

## 📌 tail 명령어

파일의 뒷부분을 출력한다.

```
# 파일의 마지막 10줄 출력 (기본값)
tail filename

# 파일의 마지막 5줄 출력
tail -5 filename
tail -n 5 filename

# 파일의 마지막 100바이트 출력
tail -c 100 filename

# 파일의 변화를 실시간으로 모니터링
tail -f logfile.txt
# 종료: Ctrl+C

# 파일의 n번째 줄부터 끝까지 출력
tail -n +10 filename
```

## 📌 주요 옵션

명령어	옵션	설명
head	-n	출력할 줄 수를 지정한다
head	-c	출력할 바이트 수를 지정한다
tail	-n	출력할 줄 수를 지정한다
tail	-f	파일의 변화를 실시간으로 모니터링한다
tail	-c	출력할 바이트 수를 지정한다

---

## 마무리

이 가이드는 리눅스 시스템에서 가장 자주 사용되는 기본 명령어들을 다루었다. 각 명령어는 다양한 옵션을 제공하므로, 필요에 따라 `man` 명령어를 사용하여 더 자세한 정보를 확인할 수 있다.

```
# 명령어의 상세한 매뉴얼 보기
```

```
man ls
```

```
man cp
```


```
man mv
```

리눅스 명령어를 익히는 가장 좋은 방법은 직접 실습해보는 것이다. 안전한 테스트 환경에서 다양한 옵션들을 시도해보며 각 명령어의 동작을 이해하는 것이 중요하다.

## 추가 학습 자료

- `man` 명령어로 각 명령어의 상세 매뉴얼 확인
- `--help` 옵션으로 간단한 도움말 확인
- `info` 명령어로 GNU 프로그램의 상세 정보 확인

---

 **팁:** 명령어 사용 시 실수를 방지하기 위해 중요한 파일이나 디렉토리는 항상 백업을 해두는 것이 좋다.