

# Linux 3.1

## Linux 명령어 완벽 가이드

---

### 1. 파일 및 디렉토리 검색

#### find 명령어

기본 구문: `find [옵션] [설정값] [찾을파일/디렉토리명]`

#### 명령어 예제

```
# 기본 검색
find / -name passwd
find /etc -name passwd

# 타입별 검색
find /home -name passwd -type f    # passwd라는 파일만 검색
find /home -name passwd -type d    # passwd라는 디렉토리만 검색

# 패턴 검색 및 상세 정보
find /root/dir1 -name 'f*' -type f -ls    # f로 시작하는 모든 파일을 찾아 상세 정보를 출력

# 검색 후 실행
find /root/dir1 -name f3 -exec rm {} \;    # f3라는 파일을 찾아서 삭제
find /root/dir1 -name 'f*' -exec rm {} \;  # f로 시작되는 모든 파일 찾아서 삭제
```

```
root@localhost:~  
[root@localhost dir1]# ll  
합계 8  
-rw-r--r--. 2 root root 64 1월 7 19:54 copyf1.txt  
-rw-r--r--. 2 root root 64 1월 7 19:54 f1.txt  
lrwxrwxrwx. 1 root root 6 1월 7 19:51 lnfl.txt -> f1.txt  
[root@localhost dir1]# find -name f1.txt  
./f1.txt  
[root@localhost dir1]# cd ..  
[root@localhost ~]# find /etc -name f1.txt  
[root@localhost ~]# find /etc -name passwd
```

```
root@localhost:~  
[root@localhost dir1]# cd ..  
[root@localhost ~]# find /root/dir1 -name test1.txt -type f  
/root/dir1/test1.txt  
[root@localhost ~]# find /root/dir1 -name dir6 -type d  
/root/dir1/dir6  
[root@localhost ~]# find /root/dir1 -name 't*' -type f -ls  
68498764    0 -rw-r--r--  1 root    root          0  2월 10 21:31 /root/dir1/test1.txt  
68401672    0 -rw-r--r--  1 root    root          0  2월 10 21:31 /root/dir1/test2.txt  
68401698    0 -rw-r--r--  1 root    root          0  2월 10 21:31 /root/dir1/test3.txt  
68397115    4 -rw-r--r--  1 root    root        30  2월 10 21:32 /root/dir1/test.txt
```

```
root@localhost:~/dir1
[root@localhost dir1]# ll
합계 0
-rw-r--r-- 1 root root 0 2월 10 21:42 file1.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file2.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file3.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file4.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file5.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file6.txt
[root@localhost dir1]# touch test{1..5}.txt
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# find /root/dir1 -name file1.txt -exec rm {} \;
[root@localhost dir1]# ll
합계 0
-rw-r--r-- 1 root root 0 2월 10 21:42 file2.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file3.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file4.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file5.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file6.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test1.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test2.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test3.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test4.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test5.txt
[root@localhost dir1]#
```

```
root@localhost:~/dir1
[root@localhost dir1]# ll
합계 0
-rw-r--r-- 1 root root 0 2월 10 21:42 file2.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file3.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file4.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file5.txt
-rw-r--r-- 1 root root 0 2월 10 21:42 file6.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test1.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test2.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test3.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test4.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test5.txt
[root@localhost dir1]# ll
합계 0
-rw-r--r-- 1 root root 0 2월 10 21:43 test1.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test2.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test3.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test4.txt
-rw-r--r-- 1 root root 0 2월 10 21:43 test5.txt
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# find /root/dir1 -name 'f*' -exec rm {} \;
[root@localhost dir1]#
```

```
root@localhost:~/dir1
[root@localhost dir1]# ll
합계 0
drwxr-xr-x 2 root root 6 2월 10 21:47 dir1
drwxr-xr-x 2 root root 6 2월 10 21:47 dir2
drwxr-xr-x 2 root root 6 2월 10 21:47 dir3
drwxr-xr-x 2 root root 6 2월 10 21:47 dir4
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# find /root/dir1/ -name 't*' -type d -exec rm -r {} \;
find: '/root/dir1/test7': 그런 파일이나 디렉터리가 없습니다
find: '/root/dir1/test8': 그런 파일이나 디렉터리가 없습니다
find: '/root/dir1/test9': 그런 파일이나 디렉터리가 없습니다
[root@localhost dir1]# ll
합계 0
[root@localhost dir1]#
```

## 2. 파일 내용 비교

### 비교 명령어 종류

- **cmp** : 간단하게 비교
- **comm** : 비교하면서 보여주기

- **diff** : 두 파일의 차이점을 줄 단위로 비교하여 출력

## 2-1. ♦ **cmp** 명령어

기본 구문: **cmp** 파일명1 파일명2

```

root@localhost:~/dir1
[root@localhost dir1]# cat -n file1.txt
1 Linux
2 Window
3 Unix
4 Mac OS
5 Android
6 ios
7
[root@localhost dir1]# cat -n file2.txt
1 Java
2 Kotlin
3 Python
4 HTML
5 Java Script
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# cmp file1.txt file2.txt
file1.txt file2.txt differ: byte 1, line 1
[root@localhost dir1]# comm file1.txt file2.txt
      Java
      Kotlin
Linux
      Python
comm: file 2 is not in sorted order
      HTML
      Java Script
Window
comm: file 1 is not in sorted order
Unix
Mac OS
Android
ios

comm: input is not in sorted order
[root@localhost dir1]# diff file1.txt file2.txt
1,7c1,5
< Linux
< Window
< Unix
< Mac OS

```

💡 **특징**: 첫 번째로 다른 것만 찾아낸다.

## 2-2. ♦ **comm** 명령어

기본 구문: **comm** 파일명1 파일명2

```

root@localhost:~/dir1
[root@localhost dir1]# cat -n file1.txt
1 Linux
2 Window
3 Unix
4 Mac OS
5 Android
6 ios
7
[root@localhost dir1]# cat -n file2.txt
1 Java
2 Kotlin
3 Python
4 HTML
5 Java Script
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# cmp file1.txt file2.txt
file1.txt file2.txt differ: byte 1, line 1
[root@localhost dir1]# comm file1.txt file2.txt
      Java
      Kotlin
Linux
      Python
comm: file 2 is not in sorted order
      HTML
      Java Script
Window
comm: file 1 is not in sorted order
Unix
Mac OS
Android
ios

comm: input is not in sorted order
[root@localhost dir1]# diff file1.txt file2.txt
1,7c1,5
< Linux
< Window
< Unix
< Mac OS

```

⚠ **주의**: **comm** 명령어는 두 파일을 비교하여 공통점과 차이점을 출력한다. 따라서 반드시 정렬된 상태여야 한다.

**정렬 방법**: **sort** 기존파일명 -o 정렬후저장할파일명

```
root@localhost:~/dir1
[root@localhost dir1]# cat -n file1.txt
1  Android
2  Linux
3  Mac OS
4  Unix
5  Window
6  ios
[root@localhost dir1]# cat -n file2.txt
1  HTML
2  Java
3  Java Script
4  Kotlin
5  Python
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# comm file1.txt file2.txt
Android
      HTML
      Java
      Java Script
      Kotlin
Linux
Mac OS
      Python
Unix
Window
ios
[root@localhost dir1]#
```

### comm 출력 형식

- 첫 번째 컬럼: file1.txt만의 내용
- 두 번째 컬럼: file2.txt만의 내용
- 세 번째 컬럼: file1.txt, file2.txt의 공통 내용

```
root@localhost:~/dir1
1049 cat -n file1.txt
1050 cat -n file2.txt
1051 history
1052 clear
1053 cat -n file1
1054 clear
1055 history
[root@localhost dir1]# !1049
cat -n file1.txt
1  Android
2  Linux
3  Mac OS
4  SQL
5  Unix
6  Window
[root@localhost dir1]# !1050
cat -n file2.txt
1  HTML
2  Java
3  Java Script
4  Kotlin
5  Python
6  SQL
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# comm file1.txt file2.txt
Android
      HTML
      Java
      Java Script
      Kotlin
Linux
Mac OS
      Python
      SQL
Unix
Window
[root@localhost dir1]#
```

## 2-3. diff 명령어

기본 구문: diff 파일명1 파일명2

```
root@localhost:~/dir1
[root@localhost dir1]# cat -n file1.txt
 1 Android
 2 Linux
 3 Mac OS
 4 SQL
 5 Unix
 6 Window
[root@localhost dir1]# cat -n file2.txt
 1 HTML
 2 Java
 3 Java Script
 4 Kotlin
 5 Python
 6 SQL
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# diff file1.txt file2.txt
1,3c1,5
< Android
< Linux
< Mac OS
---
> HTML
> Java
> Java Script
> Kotlin
> Python
5,6d6
< Unix
< Window
[root@localhost dir1]#
```

## 3. 링크 파일

### 3-1. 🐼 하드링크 (Hard Link)

기본 구문: `ln 파일명1 파일명2`

```
root@localhost:~/dir1
[root@localhost dir1]# ls -li
합계 12
101872809 -rw-r--r-- 1 root root 18  2월 11 00:25 file1.txt
101872807 -rw-r--r-- 1 root root 18  2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18  2월 11 00:26 file3.txt
[root@localhost dir1]# ls -li
합계 16
101872809 -rw-r--r-- 2 root root 18  2월 11 00:25 fl
101872809 -rw-r--r-- 2 root root 18  2월 11 00:25 file1.txt
101872807 -rw-r--r-- 1 root root 18  2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18  2월 11 00:26 file3.txt
[root@localhost dir1]#

root@localhost:~/dir1
[root@localhost dir1]# ln file1.txt fl
[root@localhost dir1]# cat -n file1.txt
 1 This is file1.txt
[root@localhost dir1]# cat -n fl
 1 This is file1.txt
[root@localhost dir1]# rm -rf file1.txt
[root@localhost dir1]# cat -n fl
 1 This is file1.txt
[root@localhost dir1]#
```

### 하드링크 특징

- ✓ i-node 번호가 같다
- ✓ 원본 파일을 삭제해도 복사본을 사용할 수 있다
- ✓ 링크수가 증가한다
- ✓ 동일한 용량을 차지한다

### 링크 분석 예제

변경 전:

```
101872809 -rw-r--r-- 1 root root 18  2월 11 00:25 file1.txt
101872807 -rw-r--r-- 1 root root 18  2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18  2월 11 00:26 file3.txt
```

변경 후:

```
101872809 -rw-r--r-- 2 root root 18  2월 11 00:25 f1
101872809 -rw-r--r-- 2 root root 18  2월 11 00:25 file1.txt
101872807 -rw-r--r-- 1 root root 18  2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18  2월 11 00:26 file3.txt
```

f1과 file1.txt의 링크 수가 2 → 동일한 i-node를 사용하고 있어 같은 파일을 가리키는 하드링크

## 파일 정보 해석

항목	설명
101872809	i-node 번호
-rw-r--r--	파일 권한
2	하드 링크 수 (이 파일을 참조하는 링크의 개수)
root root	소유자 및 그룹
18	파일 크기(바이트)
2월 11 00:25	마지막 수정 시간
file1.txt	파일 이름

## 3-2. 심볼릭 링크 (Symbolic Link)

기본 구문: `ln -s 파일명1 파일명2`

```
root@localhost:~/dir1
[root@localhost dir1]# ln -s file1.txt f10
[root@localhost dir1]# ls -li
합계 12
101779430 lrwxrwxrwx 1 root root  9  2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18  2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18  2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18  2월 11 00:26 file3.txt
[root@localhost dir1]#
```

## 심볼릭 링크 특징

- ✓ 모든 사용자가 사용할 수 있음
- ⚠ 보안상 굉장히 취약하다
- ✓ i-node가 다르다

## 심볼릭 링크 분석 예제

```
101779430 lrwxrwxrwx 1 root root  9  2월 11 00:41 f10 -> file1.txt
```

```
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
```

## 4. 파일 압축 및 묶기


### 4-1. 파일/디렉토리 압축 및 해제

#### 압축 명령어

##### gzip 압축

기본 구문: `gzip [옵션] [압축할 파일명]`

확장자: `.gz`

 **주의:** 0바이트인 파일을 압축할 경우 압축 헤더정보가 추가되어 용량이 늘어난다.

```
root@localhost:~/dir1

[root@localhost dir1]# ls -li
합계 12
101872811 -rw-r--r-- 1 root root 0 2월 11 00:51 asd
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
[root@localhost dir1]# gzip asd
[root@localhost dir1]# ls -li
합계 16
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
[root@localhost dir1]# S
```

##### bzip2 압축

기본 구문: `bzip2 [옵션] [압축할 파일명]`

확장자: `.bz2`



```
root@localhost:~/dir1
[root@localhost dir1]# ls -li
합계 20
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
101872809 -rw-r--r-- 1 root root 2357 2월 11 01:01 qwe
[root@localhost dir1]# bzip2 qwe
bash: bzip2: 명령을 찾을 수 없습니다...
유사한 명령: 'bzip2'
[root@localhost dir1]# bzip2 qwe
[root@localhost dir1]# ls -li
합계 20
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
101872811 -rw-r--r-- 1 root root 970 2월 11 01:01 qwe.bz2
[root@localhost dir1]#
```

## xz 압축

기본 구문: xz [옵션] [압축할 파일명]

확장자: .xz

```
root@localhost:~/dir1
[root@localhost dir1]# ls -li
합계 20
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
101872809 -rw-r--r-- 1 root root 2357 2월 11 01:01 qwe
[root@localhost dir1]# xz qwe
[root@localhost dir1]# ls -li
합계 20
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
101872811 -rw-r--r-- 1 root root 992 2월 11 01:01 qwe.xz
[root@localhost dir1]#
```

## 2 압축 해제 명령어

```
# gzip 해제
gunzip [파일명]
```

```
gzip -d [압축된것.gz]

# bzip2 해제
bunzip2 [파일명]
bzip2 -d [압축된것.bz2]

# xz 해제
unxz [파일명]
xz -d [압축된것.xz]
```

```
[root@localhost dir1]# bunzip2 qwe.bz2
[root@localhost dir1]# ls -li
합계 20
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
101872809 -rw-r--r-- 1 root root 2357 2월 11 01:01 qwe
```

```
root@localhost:~/dir1
root@localhost dir1]# ls -li
합계 20
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
101872811 -rw-r--r-- 1 root root 992 2월 11 01:01 qwe.xz
root@localhost dir1]# unxz qwe.xz
root@localhost dir1]# ls -li
합계 20
101872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
101779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
101872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
101872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
101872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
101872809 -rw-r--r-- 1 root root 2357 2월 11 01:01 qwe
root@localhost dir1]#
```

## 4-2. 📁 파일/디렉토리 묶기 ( tar )

기본 구문: tar [옵션] [생성파일명] [묶을 파일이나 디렉토리]

### 🔧 주요 옵션

옵션	설명
-c	생성 (아카이브 파일 생성)
-v	묶음/해제 과정 출력

옵션	설명
-f	목록을 파일 또는 해제할 파일
-t	목록을 해제하지 않고 목록만 출력
-r	기존의 아카이브 파일에 파일/디렉토리 추가

```

root@localhost:~/dir1/package
root@localhost dir1]# ls -li
합계 20
01872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
01779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file
.txt
01872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
01872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
01872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
35532080 drwxr-xr-x 2 root root 66 2월 11 01:10 package
01872809 -rw-r--r-- 1 root root 2357 2월 11 01:01 qwe
root@localhost dir1]# cd package
root@localhost package]# ls -li
합계 24
35532091 -rw-r--r-- 1 root root 2357 2월 11 01:10 f1

root@localhost:~/dir1
[root@localhost dir1]# tar -cvf backup.tar file1.txt file2.txt
file1.txt
file2.txt
[root@localhost dir1]# ls -li
합계 32
01872796 -rw-r--r-- 1 root root 24 2월 11 00:51 asd.gz
01872811 -rw-r--r-- 1 root root 10240 2월 11 01:15 backup.tar
01779430 lrwxrwxrwx 1 root root 9 2월 11 00:41 f10 -> file1.txt
01872810 -rw-r--r-- 1 root root 18 2월 11 00:40 file1.txt
01872807 -rw-r--r-- 1 root root 18 2월 11 00:25 file2.txt
01872793 -rw-r--r-- 1 root root 18 2월 11 00:26 file3.txt
35532080 drwxr-xr-x 2 root root 66 2월 11 01:10 package
01872809 -rw-r--r-- 1 root root 2357 2월 11 01:01 qwe
[root@localhost dir1]#

```

## 5. 셸 관련 명령어

### 기본 정보

- 리눅스 기본 셸: bash
- 경로 및 파일: /etc/shells
- 환경설정 파일: /etc/bashrc

### 5-1. 셸 변경

기본 구문: chsh [옵션]

```

[root@localhost etc]# chsh
Changing shell for root.
New shell [/bin/bash]: /bin/sh

```

옵션:

- -l : 정의된 셸 목록 출력

### 5-2. alias 설정 및 목록 출력

```

# alias 설정
alias l='ls -l'

# alias 해제
unalias l

```

```
root@localhost:~/dir1
root@localhost dir1]# alias l='ls -li'
root@localhost dir1]# l
계 32
01872796 -rw-r--r-- 1 root root    24  2월 11 00:51 asd.gz
01872811 -rw-r--r-- 1 root root 10240  2월 11 01:15 backup.tar
01779430 lrwxrwxrwx 1 root root     9  2월 11 00:41 f10 -> file1.txt
01872810 -rw-r--r-- 1 root root    18  2월 11 00:40 file1.txt
01872807 -rw-r--r-- 1 root root    18  2월 11 00:25 file2.txt
01872793 -rw-r--r-- 1 root root    18  2월 11 00:26 file3.txt
35532080 drwxr-xr-x 2 root root    66  2월 11 01:10 package
01872809 -rw-r--r-- 1 root root   2357  2월 11 01:01 qwe
root@localhost dir1]#
```

```
[root@localhost dir1]# l
bash: l: 명령을 찾을 수 없습니다...
[root@localhost dir1]#
```

#### 💡 영구 설정 방법:

1. cd /etc 로 이동
2. vi bashrc 로 편집
3. 맨 마지막 fi 밑에 alias l='ls -li' 추가
4. 터미널 재시작 후 l 입력하면 실행됨

## 5-3. 🎨 셸 프롬프트 설정

기본 형태: [root@localhost ~]

### abc 프롬프트 변수

변수	설명
\W	최종 디렉토리만 출력
\w	전체경로 출력
\d	날짜와 시간 출력
\t	시간 출력(24시간제)
\T	시간 출력(12시간제)
\@	오전/오후 출력(12시간제)

설정 예제:

```
PS1="[ \d][\@][\u@\h \W]\\$ "
```

## 6. 프로세스 관련 명령어

💡 최상위 프로세스: systemd

### 6-1. 🖨️ ps 명령어

기본 구문: `ps [옵션]`

주요 옵션: `-a`, `-au`, `-aux`, `-ef`

```
root@localhost: /# ps -a
  PID TTY          TIME CMD
 2224 tty2      00:00:00 gnome-session-b
 6411 pts/0      00:00:00 ps

root@localhost: /# ps -au
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      2218  0.0  0.2 374340 8032 tty2    Ssl+  2월 10   0:00 /usr/libexec/gdm-wayland-session --register-session gnome-sess
root      2224  0.0  0.5 513484 19616 tty2    Sl+   2월 10   0:00 /usr/libexec/gnome-session-binary
root      2903  0.0  0.1 224460 5888 pts/0    Ss    2월 10   0:00 bash
root      3740  0.0  0.1 224256 5760 pts/1    Ss+   2월 10   0:00 bash
root      6416  0.0  0.0 225492 3456 pts/0    R+    01:40   0:00 ps -au

root@localhost: /# ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3 171444 13696 ?        Ss     2월 10   0:01 /usr/lib/systemd/systemd rhgb --switched-root --system --deser
root         2  0.0  0.0      0      0 ?        S      2월 10   0:00 [kthreadd]
root         3  0.0  0.0      0      0 ?        S      2월 10   0:00 [pool_workqueue_]
root         4  0.0  0.0      0      0 ?        I<     2월 10   0:00 [kworker/R-rcu_g]
root         5  0.0  0.0      0      0 ?        I<     2월 10   0:00 [kworker/R-rcu_p]
root         6  0.0  0.0      0      0 ?        I<     2월 10   0:00 [kworker/R-slub_]
root         7  0.0  0.0      0      0 ?        I<     2월 10   0:00 [kworker/R-netns]
root         9  0.0  0.0      0      0 ?        I<     2월 10   0:00 [kworker/0:0H-events_highpri]
root        10  0.0  0.0      0      0 ?        I      2월 10   0:00 [kworker/u512:0-events_unbound]
root        11  0.0  0.0      0      0 ?        I<     2월 10   0:00 [kworker/R-mm_pe]
```

```
root@localhost: /# ps -ef
PID      PPID     C   STIME TTY          TIME CMD
1         0   0   2월 10 ?           00:00:01 /usr/lib/systemd/systemd rhgb --switched-root --system --deserialize 31
2         0   0   2월 10 ?           00:00:00 [kthreadd]
3         2   0   2월 10 ?           00:00:00 [pool_workqueue_]
4         2   0   2월 10 ?           00:00:00 [kworker/R-rcu_g]
5         2   0   2월 10 ?           00:00:00 [kworker/R-rcu_p]
6         2   0   2월 10 ?           00:00:00 [kworker/R-slub_]
7         2   0   2월 10 ?           00:00:00 [kworker/R-netns]
9         2   0   2월 10 ?           00:00:00 [kworker/0:0H-events_highpri]
10        2   0   2월 10 ?           00:00:00 [kworker/u512:0-events_unbound]
11        2   0   2월 10 ?           00:00:00 [kworker/R-mm_pe]
12        2   0   2월 10 ?           00:00:00 [kworker/u512:1-netns]
13        2   0   2월 10 ?           00:00:00 [rcu_tasks_kthre]
14        2   0   2월 10 ?           00:00:00 [rcu_tasks_rude_]
15        2   0   2월 10 ?           00:00:00 [rcu_tasks_trace]
16        2   0   2월 10 ?           00:00:00 [ksoftirqd/0]
```

### 6-2. 🌳 pstree 명령어

기본 구문: `pstree [옵션]`

주요 옵션: `-p` (각각의 프로세스 아이디를 보여줌)

```
root@localhost: /# pstree -p
systemd(1)─ModemManager(901)─{ModemManager}(912)
          │                  │{ModemManager}(920)
          └─NetworkManager(833)─{NetworkManager}(857)
                                │{NetworkManager}(859)
          └─VGAAuthService(858)
          └─accounts-daemon(845)─{accounts-daemon}(851)
                                │{accounts-daemon}(853)
          └─alsactl(882)
          └─atd(977)
          └─auditd(801)─sedispatch(803)
                        │{auditd}(802)
                        │{auditd}(804)
          └─avahi-daemon(834)─avahi-daemon(873)
          └─chronyd(844)
          └─colord(2090)─{colord}(2096)
                        │{colord}(2098)
          └─crond(985)
          └─cupsd(927)
          └─dbus-broker-lau(831)─dbus-broker(832)
          └─fwupd(2772)─{fwupd}(2773)
                        │{fwupd}(2774)
                        │{fwupd}(2775)
                        │{fwupd}(2776)
          └─gdm(987)─gdm-session-wor(2174)─gdm-wayland-ses(2218)─gnome-session-b(2224)─{gnome-session-b}(2255)
                                                           │{gnome-session-b}(2256)
```

### 6-3. top 명령어

실시간 프로세스 모니터링

기본 구문: top [옵션]

```
root@localhost: /— top

top - 01:47:25 up 2:33, 2 users, load average: 0.00, 0.00, 0.00
tasks: 250 total, 1 running, 249 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.8 us, 0.3 sy, 0.0 ni, 98.5 id, 0.0 wa, 0.3 hi, 0.0 si, 0.0 st
MiB Mem : 3626.0 total, 925.7 free, 1431.3 used, 1598.6 buff/cache
MiB Swap: 4012.0 total, 4012.0 free, 0.0 used, 2194.7 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 2278 root        20   0 3987288 233812 127532 S   2.7   6.3   2:07.23 gnome-shell
 2855 root        20   0 1008956 84280  57248 S   0.7   2.3   0:49.81 gnome-terminal-
    1 root        20   0 171444   13696  9856 S   0.0   0.4   0:01.95 systemd
    2 root        20   0      0      0      0 S   0.0   0.0   0:00.05 kthreadd
    3 root        20   0      0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_
    4 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_g
    5 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
    6 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_
    7 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
    9 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
   10 root        20   0      0      0      0 I   0.0   0.0   0:00.00 kworker/u512:0-events_unbound
   11 root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_pe
   12 root        20   0      0      0      0 I   0.0   0.0   0:00.03 kworker/u512:1-netns
```

## 프로세스 우선순위 관리

### 우선순위 기본 개념

- 수치가 낮을수록 우선순위가 높다
- PR 값으로 우선순위 확인
- PR 자체는 변경 불가, NI 값은 변경 가능
- nice값 범위: -20 ~ 19

## 6-4. 🎯 nice 명령어 (프로그램 실행 시 우선순위 설정)

기본 구문: nice [옵션] [설정값] [프로그램명]

```
# +10 우선순위 설정 (낮춤)
nice -10 top
```

```
# -10 우선순위 설정 (높임)
nice --10 top
```

## 6-5. 🛠️ renice 명령어 (실행중인 프로세스 우선순위 변경)

기본 구문: renice [옵션] [설정값] [PID]

```
# 우선순위를 10만큼 낮춤
renice 10 7034
```

```
# 우선순위를 10만큼 높임
renice -10 7304
```

## 🚫 프로세스 종료

## 6-6. ⚡ kill 명령어

기본 구문: kill [옵션] [PID]

### 📋 주요 시그널 목록

1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL	5) SIGTRAP
2) SIGABRT	7) SIGBUS	8) SIGFPE	9) SIGKILL	10) SIGUSR1
3) SIGSEGV	12) SIGUSR2	13) SIGPIPE	14) SIGALRM	15) SIGTERM
4) SIGSTKFLT	17) SIGCHLD	18) SIGCONT	19) SIGSTOP	20) SIGTSTP
5) SIGTTIN	22) SIGTTOU	23) SIGURG	24) SIGXCPU	25) SIGXFSZ
6) SIGVTALRM	27) SIGPROF	28) SIGWINCH	29) SIGIO	30) SIGPWR
7) SIGSYS	34) SIGRTMIN	35) SIGRTMIN+1	36) SIGRTMIN+2	37) SIGRTMIN+3

🔥 중요한 시그널: 1, 2, 3, 9, 15, 18, 19, 20

## 6-7. 🎯 killall 명령어

기본 구문: killall [옵션] [프로세스명]

```
# 프로세스명으로 강제 종료
killall -9 프로세스명
```



**Linux 명령어 마스터하기!** 이 가이드를 통해 리눅스 시스템을 더욱 효율적으로 관리해보세요.

[Linux 2.2](#)