



LINUX 명령(1)

- LINUX 파일 개요
- 파일 관련 명령
- 프로세스 관리 명령



LINUX 파일 개요

□ LINUX 파일 종류

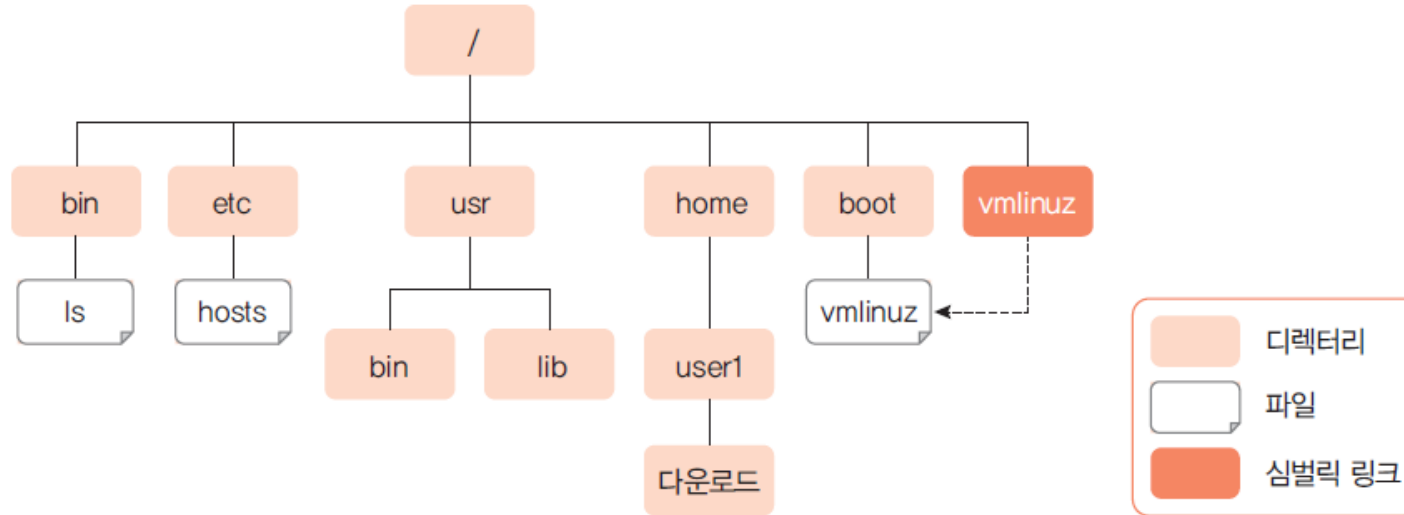
- **file: 관련 있는 정보들의 집합**
- **일반 파일(regular file): 데이터를 저장하는데 주로 사용**
 - 텍스트파일, 실행파일, 이미지파일 등의 대부분의 파일이 해당
 - 실행파일과 이미지파일은 binary 형태로 저장되어 있는 binary file
- **디렉터리(directory): 파일을 효과적으로 관리하기 위해 사용**
 - 리눅스에서는 디렉터리도 파일로 취급
 - 해당 디렉터리에 저장된 파일이나 하위 디렉터리에 대한 정보가 저장
- **심볼릭 링크(symbolic link)**
: 원본 파일을 대신하여 다른 이름으로 파일명을 지정한 파일(바로가기 파일)
- **장치 파일(device file): 하드디스크나 키보드 같은 각종 장치**



LINUX 파일 개요

□ 디렉터리 계층 구조

- 모든 디렉터리의 출발점은 root(/) : **ls -aF /**



- 작업 디렉터리(working directory): 현재 사용 중인 디렉터리('.') 로 표시)
pwd 명령으로 확인 가능
- 상위 디렉터리(parent directory): '..'로 표시
- 홈 디렉터리(home directory): 각 사용자에게 할당된 디렉터리('~'로 표시)



LINUX 파일 개요

□ 주요 디렉터리 역할

디렉터리	역할
dev	장치 파일이 담긴 디렉터리
home	사용자 홈 디렉터리가 생성되는 디렉터리
media	CD-ROM과 USB 같은 외부 장치를 연결(마운트)하는 디렉터리
opt	추가 패키지가 설치되는 디렉터리
sys	리눅스 커널과 관련된 파일이 있는 디렉터리
usr	기본 실행 파일과 라이브러리, 헤더 파일 등이 있는 디렉터리 (Unix System Resource의 약자)
boot	부팅에 필요한 커널 파일이 있는 디렉터리
etc	리눅스 설정을 위한 각종 파일이 있는 디렉터리
mnt	파일 시스템을 임시로 마운트하는 디렉터리
proc	프로세스 정보등 커널 관련 정보가 저장
run	실행 중인 서비스와 관련된 파일이 저장
srv	FTP나 Web 등 시스템에서 제공하는 서비스의 데이터가 저장
var	시스템 운영 중에 발생하는 데이터나 로그 등 자주바뀌는 파일 저장
lost+found	파일 시스템에 문제가 되어 복구될 때 해당 파일이 저장
tmp	시스템 사용 중에 발생하는 임시 데이터가 저장



LINUX 파일 개요

□ 경로(path)

- 파일 시스템에서 디렉터리 계층 구조에 있는 특정 파일이나 디렉터리 표시
- 경로 구분자는 '/'(slash) 사용
- 절대 경로명
 - 항상 루트 디렉터리부터 시작(=반드시 '/'로 시작)
 - 특정 파일이나 디렉터리의 모든 중간 디렉터리 이름 표시
 - 예> /bin/lis , /etc/init.d/
- 상대 경로명
 - 현재 디렉터리를 기준으로 시작 (=현재 디렉터리가 어디냐에 따라 달라짐)
 - 예> ./src/debug , ../../etc/init.d/ , ~/src/debug



LINUX 파일 개요

□ 파일과 디렉터리 이름 규칙(naming convention)

- 알파벳, 숫자, 붙임표(-), 밑줄(_), 마침표(.)만 허용
- 공백문자, *, |, ", ', @, #, \$, %, ^, &, / 등의 특수문자 사용 불가
- 영문은 대문자와 소문자를 구별하여 다른 글자로 취급
- 첫 글자가 마침표(.)로 시작하면 숨김 파일로 간주
- 예> &dir, my home, game\, *bad-name
test.test.test, thisisatestfilecreatedbyme



파일 관련 명령

□ 디렉터리 사용 명령(1)

- 현재 디렉터리 확인: **pwd**

```
linuxer@linuxer-PC:~$ pwd  
/home/linuxer
```

- 디렉터리 이동: **cd [디렉터리]**

```
linuxer@linuxer-PC:~$ cd /etc  
linuxer@linuxer-PC:/etc$ cd /usr/lib  
linuxer@linuxer-PC:/usr/lib$ cd ../../..  
linuxer@linuxer-PC:/$ cd  
linuxer@linuxer-PC:~$
```



파일 관련 명령

□ 디렉터리 사용 명령(2)

▪ 디렉터리 내용 확인: **ls [옵션] [디렉터리(파일)]**

linuxer@linuxer-PC:~\$ ls -a	(display list all files)
linuxer@linuxer-PC:~\$ ls -d /*	(display directory information)
linuxer@linuxer-PC:~\$ ls -d /*/	
linuxer@linuxer-PC:~\$ ls -l /*	(display file details)
linuxer@linuxer-PC:~\$ ls -R /	(display even subdirectory)
linuxer@linuxer-PC:~\$ ls -aF /	(/:directory, @:symbolic, *:execution)

▪ 파일 상세 정보의 의미: **-l** 옵션

[종류]	[접근권한]	[링크수]	[소유자]	[소속그룹]	[크기]	[수정시간]	[파일명]
-rw-r--r--	1	linuxer	linuxer	74	8월 13 16:57	readme.txt	
drwxr-xr-x	2	linuxer	linuxer	4096	8월 13 16:57	다운로드	

[종류] **-**: 일반파일, **d**: 디렉터리 파일, **l**: 심볼릭 링크 파일,
b: 블록단위로 읽고 쓰는 장치 파일, **c**: 섹터단위로 읽고 쓰는 문자 장치 파일
p: 파이프 파일(프로세스 간 통신에 사용되는 특수 파일)
s: 소켓파일(네트워크 통신에 사용되는 특수 파일)



파일 관련 명령

□ 디렉터리 사용 명령(3)

■ 디렉터리 생성: **mkdir** [옵션] 디렉터리

```
linuxer@linuxer-PC:~$ mkdir dir1
linuxer@linuxer-PC:~$ mkdir dir2 dir3 dir4
linuxer@linuxer-PC:~$ mkdir -p dir5/dir6/dir7
linuxer@linuxer-PC:~$ ls -R dir2
```

■ 디렉터리 삭제: **rmdir** [옵션] 디렉터리

```
linuxer@linuxer-PC:~$ rmdir dir1
linuxer@linuxer-PC:~$ rmdir dir2 dir3 dir4
linuxer@linuxer-PC:~$ rmdir dir5 (failed to remove)
linuxer@linuxer-PC:~$ rmdir -p dir5/dir6/dir7
linuxer@linuxer-PC:~$ rmdir dir5/dir6/dir7 dir5/dir6 dir5
```



파일 관련 명령

□ 파일 사용 명령(1)

- 파일 내용 출력(연속): **cat [옵션] 파일**

```
linuxer@linuxer-PC:~$ cat /etc/hosts
linuxer@linuxer-PC:~$ cat -n /etc/hosts
```

- 파일 내용 출력(화면단위): **more [옵션] 파일**

```
linuxer@linuxer-PC:~$ more /etc/services
linuxer@linuxer-PC:~$ more +10 /etc/services
```

[화면이동] 스페이스바: 다음화면, 엔터키: 한 줄씩 스크롤, '/' 문자: 해당 문자열 검색, q: 종료

- 파일 내용 출력(화면단위 개선): **less 파일**

```
linuxer@linuxer-PC:~$ less /etc/services
```

[화면이동] j: 한 줄씩 다음 행으로 스크롤, k: 한 줄씩 이전 행으로 스크롤
스페이스바, Ctrl+f: 다음 화면으로 이동, Ctrl+b: 이전 화면으로 이동



파일 관련 명령

□ 파일 사용 명령(2)

- 파일 앞부분 출력: **head [옵션] 파일**

```
linuxer@linuxer-PC:~$ head /etc/services
```

```
linuxer@linuxer-PC:~$ head -7 /etc/services (지정한 행수만큼 출력)
```

- 파일 뒷부분 출력: **tail [옵션] 파일**

```
linuxer@linuxer-PC:~$ tail /etc/services
```

```
linuxer@linuxer-PC:~$ tail -7 /etc/services (지정한 행수만큼 출력)
```

```
linuxer@linuxer-PC:~$ tail -f /etc/services
```

[참고] -f 옵션은 파일에 지속적으로 행이 추가되는 로그 파일을 모니터링할 때 유용함



파일 관련 명령

□ 파일 사용 명령(3)

- 파일(디렉터리) 복사: **cp [옵션] 파일1(디렉터리1) 파일2(디렉터리2)**

```
linuxer@linuxer-PC:~$ cp /etc/hosts text1
```

```
linuxer@linuxer-PC:~$ mkdir dir1
```

```
linuxer@linuxer-PC:~$ cp text1 dir1
```

```
linuxer@linuxer-PC:~$ cp text1 dir1/text2
```

```
linuxer@linuxer-PC:~$ cp text1 /etc (permission denied)
```

```
linuxer@linuxer-PC:~$ cp /etc/hosts /etc/services temp
```

```
linuxer@linuxer-PC:~$ cp -i /etc/hosts text1 (check overwrite)
```

```
linuxer@linuxer-PC:~$ cp dir1 dir2 (copy directory failed)
```

```
linuxer@linuxer-PC:~$ cp -r dir1 dir2 (success)
```

```
linuxer@linuxer-PC:~$ ls -F dir2
```



파일 관련 명령

□ 파일 사용 명령(4)

- 파일 이동 및 파일명 변경: **mv [옵션] 파일1(디렉터리1) 파일2(디렉터리2)**

```
linuxer@linuxer-PC:~$ mv text1 data1
```

```
linuxer@linuxer-PC:~$ mv data1 dir1
```

```
linuxer@linuxer-PC:~$ cp dir1/data1 text1
```

```
linuxer@linuxer-PC:~$ mv text1 dir1/data2
```

```
linuxer@linuxer-PC:~$ mv dir1/data1 dir1/data2 .
```

```
linuxer@linuxer-PC:~$ mv -i data1 data2 (check overwrite)
```

```
linuxer@linuxer-PC:~$ mv dir2 dir3 (change directory name)
```

```
linuxer@linuxer-PC:~$ mv dir3 dir4 (change directory name)
```



파일 관련 명령

□ 파일 사용 명령(5)

▪ 파일 삭제: **rm [옵션] 파일(디렉터리)**

```
linuxer@linuxer-PC:~$ rm data2
```

```
linuxer@linuxer-PC:~$ rm -i data1
```

 (confirm removal)

```
linuxer@linuxer-PC:~$ rm dir4
```

 (remove directory fail)

```
linuxer@linuxer-PC:~$ rm -r dir4
```

 (success)

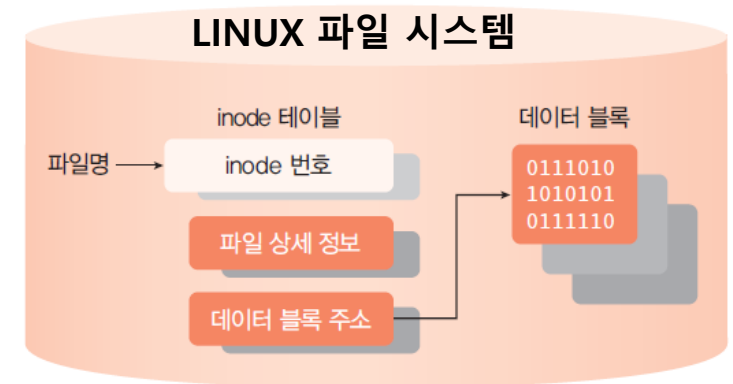
```
linuxer@linuxer-PC:~$ rm -ri dir4
```



파일 관련 명령

□ LINUX 파일 구조

- 파일 = 파일명 + inode + 데이터 블록
- inode: 파일에 대한 정보를 가진 구조체
 - 외부적으로는 번호로 표시
 - 내부적으로는 파일의 종류, 크기, 소유자, 변경시간, 파일명 등의 상세 정보와 데이터 블록의 주소를 저장
- 파일의 inode 번호 확인: **ls -i**



```
linuxer@linuxer-PC:~$ ls -i
```

1849485 공개	1849486 문서	1982074 비디오	1849487 음악
1849482 다운로드	1849480 바탕화면	1849488 사진	1849484 템플릿



파일 관련 명령

□ 파일 사용 명령(6)

▪ 파일 링크 생성: **ln [옵션] 원본파일 링크파일**

```
linuxer@linuxer-PC:~$ ln dir1/data1 dir1/data1.ln
```

```
linuxer@linuxer-PC:~$ ls -li dir1
```

 (동일한 inode 번호 확인)

```
1443967 data1 1443967 data1.ln 1443969 hosts 1443970 services
```

```
linuxer@linuxer-PC:~$ cat dir/data1
```

```
linuxer@linuxer-PC:~$ cat dir/data1.ln
```

 (내용까지 동일한 것 확인)

```
linuxer@linuxer-PC:~$ ln -s dir1/data1 dir1/data1.sl
```

```
linuxer@linuxer-PC:~$ ls -li dir1
```

 (심볼릭 링크 확인)

```
-rw-r--r-- 2 linuxer linuxer 225 8월 13 17:27 data1
```

```
-rw-r--r-- 2 linuxer linuxer 225 8월 13 17:27 data1.ln
```

```
lrwxrwxrwx 1 linuxer linuxer 5 8월 13 17:28 data1.sl -> data1
```

```
-rw-r--r-- 2 linuxer linuxer 225 8월 13 17:17 hosts
```




파일 관련 명령

□ 파일 사용 명령(7)

- 빈 파일 생성 및 수정 시간 변경: **touch [옵션] 파일**

```
linuxer@linuxer-PC:~$ ls -l dir1/data1
-rw-r--r--  2 linuxer linuxer  225  8월 13 17:27  data1
```

```
linuxer@linuxer-PC:~$ touch  dir1/data1
```

```
linuxer@linuxer-PC:~$ ls -l dir1/data1
-rw-r--r--  2 linuxer linuxer  225  8월 13 17:29  data1
```

```
linuxer@linuxer-PC:~$ touch  -t 12310101  dir1/data1  (시간 변경)
```

```
linuxer@linuxer-PC:~$ ls -l dir1/data1
-rw-r--r--  2 linuxer linuxer  225 12월 31 2020  data1
```



파일 관련 명령

□ 파일 사용 명령(8)

■ 파일 내용 검색: **grep** [옵션] 패턴 파일

```
linuxer@linuxer-PC:~$ grep root /etc/passwd
linuxer@linuxer-PC:~$ cp /etc/services svc.txt
linuxer@linuxer-PC:~$ grep HTTP svc.txt      (패턴이 포함된 행 출력)
linuxer@linuxer-PC:~$ grep -l hello *.c      (패턴이 포함된 파일만 표시)
linuxer@linuxer-PC:~$ grep -n dev ~/.txt     (행 번호를 포함한 행 출력)
```

■ 파일 찾기: **find** [경로 검색 조건] [동작]

```
linuxer@linuxer-PC:~$ find /bin -name ls      (파일명으로 검색)
linuxer@linuxer-PC:~$ find /home -user root   (사용자가 소유한 모든 파일)
linuxer@linuxer-PC:~$ find /tmp -user root -exec rm {} \;
linuxer@linuxer-PC:~$ find /tmp -user root -ok rm {} \;
```



파일 관련 명령

□ 파일 사용 명령(9)

- 명령 위치 찾기: **whereis** [옵션] 파일

```
linuxer@linuxer-PC:~$ whereis mv
linuxer@linuxer-PC:~$ whereis firefox
linuxer@linuxer-PC:~$ whereis ls grep find firefox
```

[검색위치] /bin, /usr/bin, /etc, /usr/etc, /sbin, /usr/sbin, /usr/share/man 등 **정해진 디렉터리**에서만 검색

- 명령 위치 찾기: **which** 명령

```
linuxer@linuxer-PC:~$ which mv
linuxer@linuxer-PC:~$ which firefox
linuxer@linuxer-PC:~$ which ls grep find firefox
```

[검색위치] **PATH 환경변수**로 지정된 경로에서 파일 검색



파일 관련 명령

□ 파일 속성 명령(1)

- 파일 종류 확인: **file** 파일

```
linuxer@linuxer-PC:~$ file /etc/hosts
```

```
linuxer@linuxer-PC:~$ file /dev/tty0 /dev/stdin /etc/passwd
```

- 파일 소유자의 그룹 확인: **groups** [사용자명]

```
linuxer@linuxer-PC:~$ groups
```

```
linuxer@linuxer-PC:~$ groups root
```

```
linuxer@linuxer-PC:~$ groups linuxer      (본인의 계정을 입력해서 확인)
```

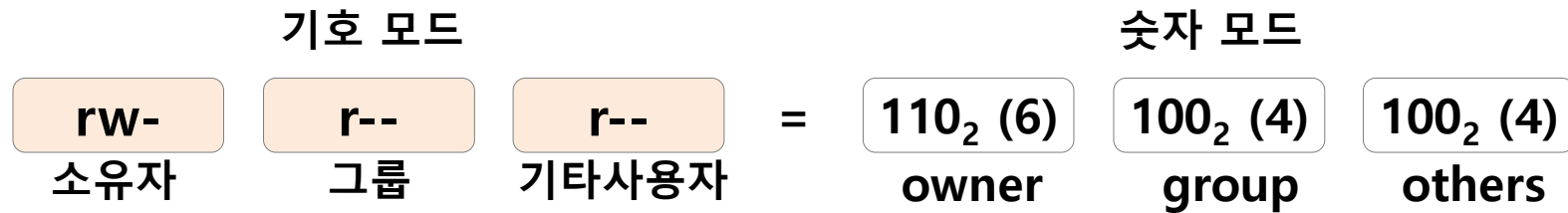


파일 관련 명령

□ 파일 속성 명령(2)

▪ 파일의 접근 권한 표기 방식

```
linuxer@linuxer-PC:~$ ls -l dir1
-rw-r--r-- 2 linuxer linuxer 225 8월 13 17:27 data1
```



접근권한	의미
rwxr-xr-x	소유자는 읽기/쓰기/실행 권한, 그룹과 기타 사용자는 읽기/실행 권한
r-xr-xr-x	소유자, 그룹, 기타 사용자 모두 읽기/실행 권한
rw-----	소유자만 읽기/쓰기 권한을 가지고 그룹과 기타 사용자는 아무 권한 없음
rw-rw-rw-	소유자, 그룹, 기타 사용자 모두 읽기/쓰기 권한
rw-rw-rwx	소유자, 그룹, 기타 사용자 모두 읽기/쓰기/실행 권한
r-----	소유자만 읽기 권한을 가짐



파일 관련 명령

□ 파일 속성 명령(3)

- 파일 접근 권한 변경: **chmod** [옵션] 설정권한 대상파일(디렉터리)

```
linuxer@linuxer-PC:~$ ls -l dir1
-rw-r--r-- 2 linuxer linuxer 225 8월 13 17:27 data1
linuxer@linuxer-PC:~$ chmod u-w dir1/data1      (기호를 이용한 권한설정)
linuxer@linuxer-PC:~$ ls -l dir1
-r--r--r-- 2 linuxer linuxer 225 8월 13 17:27 data1
linuxer@linuxer-PC:~$ chmod ugo+rwx dir1/data1   (a+rwx, a=rwx 가능)
```

```
linuxer@linuxer-PC:~$ chmod 111 dir1/data1      (숫자를 이용한 권한설정)
---x--x--x 2 linuxer linuxer 225 8월 13 17:27 data1
linuxer@linuxer-PC:~$ chmod 222 dir1/data1
--w--w--w- 2 linuxer linuxer 225 8월 13 17:27 data1
linuxer@linuxer-PC:~$ chmod 444 dir1/data1
-r--r--r-- 2 linuxer linuxer 225 8월 13 17:27 data1
linuxer@linuxer-PC:~$ chmod 777 dir1/data1
-rwxrwxrwx 2 linuxer linuxer 225 8월 13 17:27 data1
```



파일 관련 명령

□ 파일 속성 명령(4)

- 기본 접근 권한 확인 및 변경: **umask** [옵션] [마스크값]

```
linuxer@linuxer-PC:~$ umask
0002          (-----w- , 그룹과 기타 사용자에게 쓰기 권한은 부여하지 않겠다)
```

```
linuxer@linuxer-PC:~$ umask -S
u=rwx, g=rwx, o=rx
```

```
linuxer@linuxer-PC:~$ umask 077
```

```
linuxer@linuxer-PC:~$ umask
0077          (u=rwx, 그룹과 기타 사용자에게 아무런 권한을 부여하지 않겠다)
```

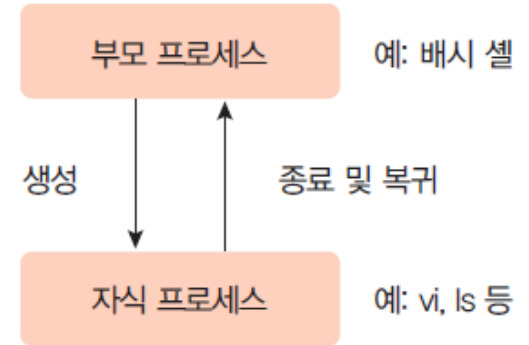
```
linuxer@linuxer-PC:~$ touch masktest.txt
-rwx----- 1 linuxer linuxer 0 8월 13 18:11 masktest.txt
```



프로세스 관리 명령

□ LINUX 프로세스 개념

- 프로세스: 실행 중인 프로그램
- 프로세스는 부모-자식 관계를 가짐
 - systemd, kthreadd를 제외하고
모든 프로세스는 부모 프로세스를 가짐
 - 자식 프로세스는 할 일이 끝나면 부모 프로세스에 결과 반환 후 종료
- 각 프로세스는 고유 번호로 관리 됨: PID
- 프로세스 구분
 - general process: 특정 기능을 수행하기 위해 부모 프로세스가 실행
 - daemon process: 특정 서비스를 제공하기 위해 커널에 의해 실행
 - orphan process: 자식 프로세스가 실행 중인데 부모 프로세스가 종료
 - zombie process: 자식 프로세스가 종료되었는데
프로세스 테이블 목록에 남아 있는 프로세스





프로세스 관리 명령

□ 프로세스 관리 명령(1)

▪ 프로세스 목록 보기: **ps** [옵션]

- 유닉스 옵션: 묶어서 사용할 수 있고, 붙임표로 시작(예: -ef)
- BSD 옵션: 묶어서 사용할 수 있고, 붙임표 없이 시작(예: aux)
- GNU 옵션: 붙임표 두 개로 시작(예: --pid)

```
linuxer@linuxer-PC:~$ ps
```

PID	TTY	TIME	CMD
20816	pts/3	00:00:00	bash
25787	pts/3	00:00:00	ps

```
linuxer@linuxer-PC:~$ ps -f
```

(프로세스 상세 정보 출력)

UID	PID	PPID	C	STIME	TTY	TIME	CMD
linuxer	20816	3626	0	04:03	pts/3	00:00:00	bash
linuxer	25818	20816	0	21:07	pts/3	00:00:00	ps -f



프로세스 관리 명령

□ 프로세스 관리 명령(2)

▪ 프로세스 목록 보기: **ps** [옵션]

linuxer@linuxer-PC:~\$ ps -e	(전체 프로세스 목록 출력)
linuxer@linuxer-PC:~\$ ps -ef	(전체 프로세스 상세목록 출력)
linuxer@linuxer-PC:~\$ ps a	(터미널에서 실행한 프로세스 정보 출력)
linuxer@linuxer-PC:~\$ ps au	(터미널에서 실행한 프로세스 상세정보 출력)
linuxer@linuxer-PC:~\$ ps ax	(시스템에서 실행 중인 모든 프로세스 출력)
linuxer@linuxer-PC:~\$ ps aux	(시스템에서 실행 중인 모든 프로세스 상세출력)
linuxer@linuxer-PC:~\$ ps -u linuxer	(특정 사용자의 프로세스 목록 출력)
linuxer@linuxer-PC:~\$ ps -fu linuxer	(특정 사용자의 프로세스 상세목록 출력)
linuxer@linuxer-PC:~\$ ps -p 25818	(특정 PID에 해당하는 프로세스 정보 출력)
linuxer@linuxer-PC:~\$ ps -ef grep bash	(특정 프로세스에 대한 정보 검색)



프로세스 관리 명령

□ 프로세스 관리 명령(3)

- 특정 프로세스 정보 검색: **pgrep** [옵션] 패턴

```
linuxer@linuxer-PC:~$ pgrep -x bash
```

```
linuxer@linuxer-PC:~$ pgrep -l bash
```

```
3637 bash
```

```
11120 bash
```

```
linuxer@linuxer-PC:~$ ps -fp $(pgrep -x bash) (PID를 이용한 상세 정보 출력)
```

```
linuxer@linuxer-PC:~$ ps -fp $(pgrep -u linuxer bash)
```

UID	PID	PPID	C	STIME	TTY	STAT	TIME	CMD
linuxer	3637	3626	0	04:03	pts/0	Ss+	0:00	bash
linuxer	11120	3626	0	04:03	pts/0	Ss+	0:00	bash

[STAT 의미] R: 실행 중(Running), S: 인터럽트 가능한 대기 상태(Sleep)
T: 작업 제어에 의해 정지된 상태(Stopped)
Z: 좀비 프로세스(defunct), s: 세션 리더 프로세스
+: 포그라운드 프로세스 그룹, l: 멀티 스레드



프로세스 관리 명령

□ LINUX 시그널(signal) 개념

- 프로세스에 어떤 사건이 발생 했음을 알리는 간단한 메시지
- 주로 사용되는 시그널

시그널	번호	기본처리	의미
SIGHUP	1	종료	터미널과의 연결이 끊어졌을 때 발생
SIGINT	2	종료	인터럽트 사용자가 Ctrl+C 입력하면 발생
SIGQUIT	3	종료, 코어덤프	종료 신호로 사용자가 Ctrl+₩을 입력하면 발생
SIGKILL	9	종료	강제 종료 발생 (이 시그널을 받은 프로세스는 무시할 수 없음)
SIGALRM	14	종료	알람에 의해 발생
SIGTERM	15	종료	kill 명령이 보내는 기본 시그널



프로세스 관리 명령

□ 프로세스 관리 명령(4)

- 프로세스에 시그널 전송: **kill [-시그널] PID**

[터미널 A에서]

```
linuxer@linuxer-PC:~$ man ps
```

[터미널 B에서]

```
linuxer@linuxer-PC:~$ ps -fp $(pgrep -x man)      (man ps 명령 PID 조회)
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
-----	-----	------	---	-------	-----	------	-----

linuxer	26496	20816	0	00:30	pts/2	00:00:00	man ps
---------	-------	-------	---	-------	-------	----------	--------

```
linuxer@linuxer-PC:~$ kill 26496                (프로세스 종료)
```

```
linuxer@linuxer-PC:~$ ps -fp $(pgrep -x man)      (종료 확인)
```

```
error: list of process IDs must follow -p          (종료되어 PID 획득 실패)
```

...



프로세스 관리 명령

□ 프로세스 관리 명령(5)

▪ 프로세스 강제 종료: **kill -9 PID**

[터미널 A에서]

```
linuxer@linuxer-PC:~$ sh
```

(셸 실행 명령)

[터미널 B에서]

```
linuxer@linuxer-PC:~$ ps -fp $(pgrep -x sh)
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
linuxer	26541	20816	0	00:40	pts/2	00:00:00	sh

```
linuxer@linuxer-PC:~$ kill 26541
```

(종료 시도)

```
linuxer@linuxer-PC:~$ ps -fp $(pgrep -x sh)
```

linuxer	26541	20816	0	00:40	pts/2	00:00:00	sh
---------	-------	-------	---	-------	-------	----------	----

(종료되지 않음)

```
linuxer@linuxer-PC:~$ kill -9 26541
```

(강제 종료)

```
linuxer@linuxer-PC:~$ pgrep -x sh
```

(종료 확인)



프로세스 관리 명령

□ 프로세스 관리 명령(6)

- 프로세스 이름으로 종료: **kill** 명령이름

[터미널 A에서]

```
linuxer@linuxer-PC:~$ man ps
```

[터미널 B에서]

```
linuxer@linuxer-PC:~$ kill man
```

(종료 시도)

```
linuxer@linuxer-PC:~$ pgrep -x man
```

[터미널 A에서]

```
linuxer@linuxer-PC:~$ sh
```

[터미널 B에서]

```
linuxer@linuxer-PC:~$ kill sh
```

(종료 시도)

```
linuxer@linuxer-PC:~$ pgrep -x sh
```

(어떻게 될까요?)



프로세스 관리 명령

□ 프로세스 관리 명령(7)

■ 프로세스 관리 도구: **top**

현재 실행 중인 프로세스에 대한 정보를 주기적으로 출력

```

linuxer@linuxer-PC: ~
top - 01:17:16 up 2 days, 10:00, 1 user, load average: 0.27, 0.11, 0.04
Tasks: 197 total, 2 running, 195 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3936.2 total, 693.6 free, 1317.5 used, 1925.1 buff/cache
MiB Swap: 1873.4 total, 1873.4 free, 0.0 used. 2322.1 avail Mem

  PID USER      PR  NI   VIRT    RES    SHR  S  %CPU  %MEM   TIME+ COMMAND
 1837 linuxer   20   0 165252    2744   2372  S   0.3   0.1   3:58.77 VBoxCli+
 1992 linuxer   20   0 4337556  432216 133064  S   0.3  10.7   5:51.10 gnome-s+
 3626 linuxer   20   0 1075432   81632  51472  S   0.3   2.0   0:54.60 gnome-t+
    1 root      20   0 167732    11716   8512  S   0.0   0.3   0:04.30 systemd
    2 root      20   0      0          0      0  S   0.0   0.0   0:00.02 kthreadd
    3 root      0 -20      0          0      0  I   0.0   0.0   0:00.00 rcu_gp
    4 root      0 -20      0          0      0  I   0.0   0.0   0:00.00 rcu_par+
    6 root      0 -20      0          0      0  I   0.0   0.0   0:00.00 kworker+
    9 root      0 -20      0          0      0  I   0.0   0.0   0:00.00 mm_perc+

```

[내부명령 키]

- Enter, Space : 다시 출력
- h, ? : 도움말 화면 출력
- n : 출력 프로세스 갯수 변경
- m : 메모리 정보(토글방식)
- u : 사용자 기준 정렬
- k : 프로세스 종료
- ↑ : 목록 위로 스크롤
- ↓ : 목록 아래로 스크롤
- q : top 명령 종료

■ GUI 응용프로그램 : [프로그램표시]-[시스템 감시]





프로세스 관리 명령

□ LINUX 프로세스 작업 제어

▪ Foreground Task(process)

- 입력한 명령의 실행 결과가 출력될 때까지 기다려야 하는 작업
- 일반적인 명령들은 포그라운드 작업으로 진행됨

```
linuxer@linuxer-PC:~$ sleep 100
```

▪ Background Task(process)

- 입력한 명령의 실행 결과가 출력되지 않아도 다른 작업 가능
- 명령 뒤에 '**&**' 입력하면 백그라운드 작업으로 처리됨

```
linuxer@linuxer-PC:~$ sleep 100 &  
[1] 27736
```



프로세스 관리 명령

□ 프로세스 관리 명령(8)

- 백그라운드 작업 목록 보기: **jobs [%작업 번호]**

```
linuxer@linuxer-PC:~$ sleep 100 &
```

```
linuxer@linuxer-PC:~$ sleep 200 &
```

```
linuxer@linuxer-PC:~$ jobs
```

```
[1]-  실행중    sleep 100 &           ( -는 +보다 이전에 접근한 작업)
```

```
[2]+  실행중    sleep 200 &           ( +는 가장 최근에 접근한 작업)
```



프로세스 관리 명령

□ 프로세스 관리 명령(9)

- 백그라운드로 작업 보내기: **bg [%작업 번호]**

```
linuxer@linuxer-PC:~$ sleep 100
```

```
^Z
```

```
[1]+  정지됨    sleep 100
```

```
linuxer@linuxer-PC:~$ bg %1
```

```
[1]+  sleep 100 &
```

```
linuxer@linuxer-PC:~$ jobs
```

```
[1]+  실행중    sleep 100 &
```



프로세스 관리 명령

□ 프로세스 관리 명령(10)

- 포그라운드로 작업 가져오기: **fg [%작업 번호]**

```
linuxer@linuxer-PC:~$ jobs  
[1]+  실행중    sleep 100 &  
linuxer@linuxer-PC:~$ fg  
sleep 100
```

- 실행 중인 작업 종료: **^C** or **kill %작업번호**

```
linuxer@linuxer-PC:~$ sleep 100  
^C  
linuxer@linuxer-PC:~$ sleep 100 &  
[1] 27736  
linuxer@linuxer-PC:~$ kill %1  
linuxer@linuxer-PC:~$ jobs  
[1]+  종료됨    sleep 100
```



프로세스 관리 명령

□ 프로세스 관리 명령(8)

- 로그아웃 이후에도 백그라운드 작업 계속 진행: **nohup 명령&**

```
linuxer@linuxer-PC:~$ nohup find / -name passwd &
```

```
[1] 27741
```

```
linuxer@linuxer-PC:~$ nohup: 입력 무시 및 'nohup.out'에 출력 추가  
exit (터미널 종료)
```

```
linuxer@linuxer-PC:~$ nohup find / -name passwd > pw.dat 2>&1 &
```

```
[1] 27741
```

```
linuxer@linuxer-PC:~$ exit
```

[다시 터미널 열어서 확인]

```
linuxer@linuxer-PC:~$ more pw.dat
```

[셸 리디렉션] n>&m (0: 표준입력, 1: 표준출력, 2: 표준오류)