

# System Programming & OS 실습

## 2. CentOS 기본 조작법

최민국, 정지현, 안석현, 김선재

Dankook University

{mgchoi, wlgjsjames7224, seokhyun, rlatjswo0824}@dankook.ac.kr

# Index

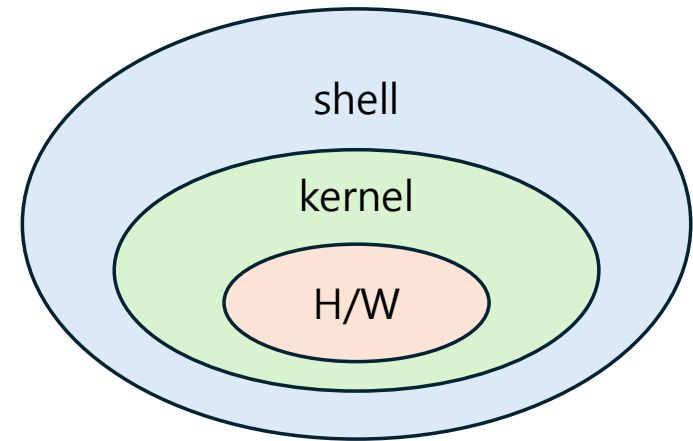
- 셸이란?
- 파일과 디렉토리 개념
- 기본 명령어
  - 기초 명령어
  - 파일 목록/내용 조회 관련 명령어
  - 파일 생성/복사/이동/삭제 관련 명령어
  - 압축/해제 관련 명령어
- 프로세스 기본 개념
  - 프로세스 관련 명령어

# SHELL

## ❖ 리눅스 셸



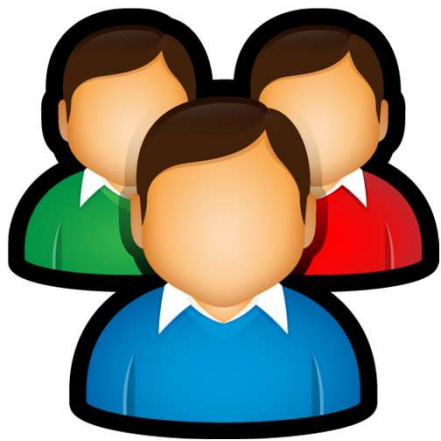
User



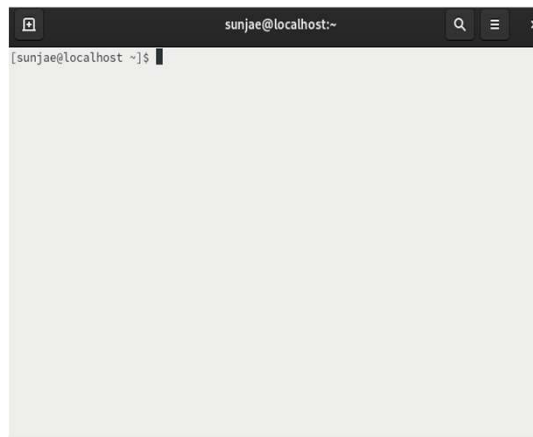
Linux system

# SHELL

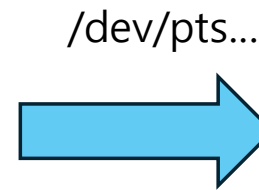
## ❖ 터미널을 통한 쉘의 사용



User



Terminal program

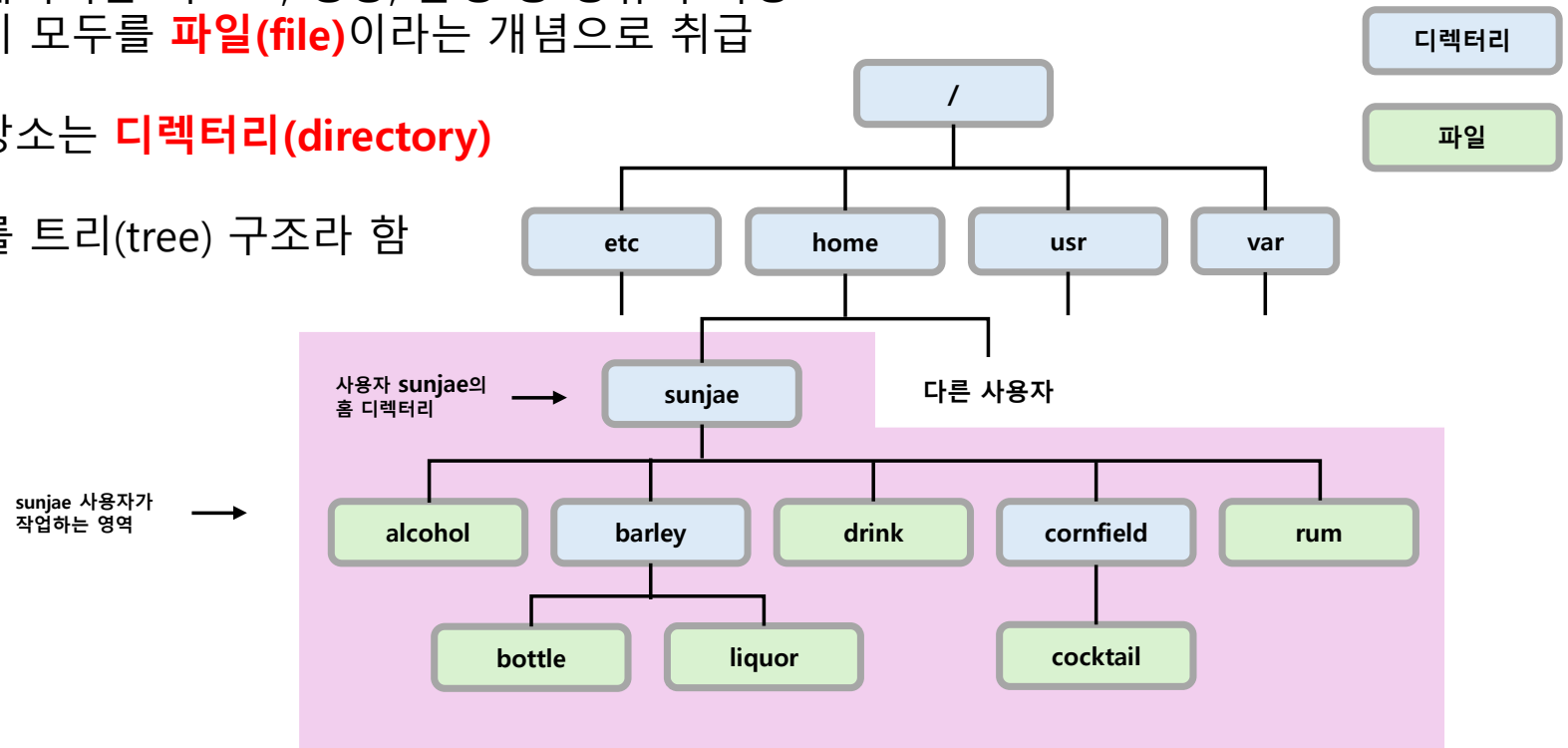


**bash**

# 파일과 디렉터리 개념

## ❖ 파일과 디렉터리의 개념

- 컴퓨터가 다루는 데이터는 텍스트, 영상, 음성 등 종류가 다양하지만 리눅스는 이 모두를 **파일(file)**이라는 개념으로 취급
- 파일을 모아두는 장소는 **디렉터리(directory)**
- 이러한 계층 구조를 트리(tree) 구조라 함



파일과 디렉터리 개념도

# 파일과 디렉터리 개념

## ❖ 파일과 디렉터리의 위치

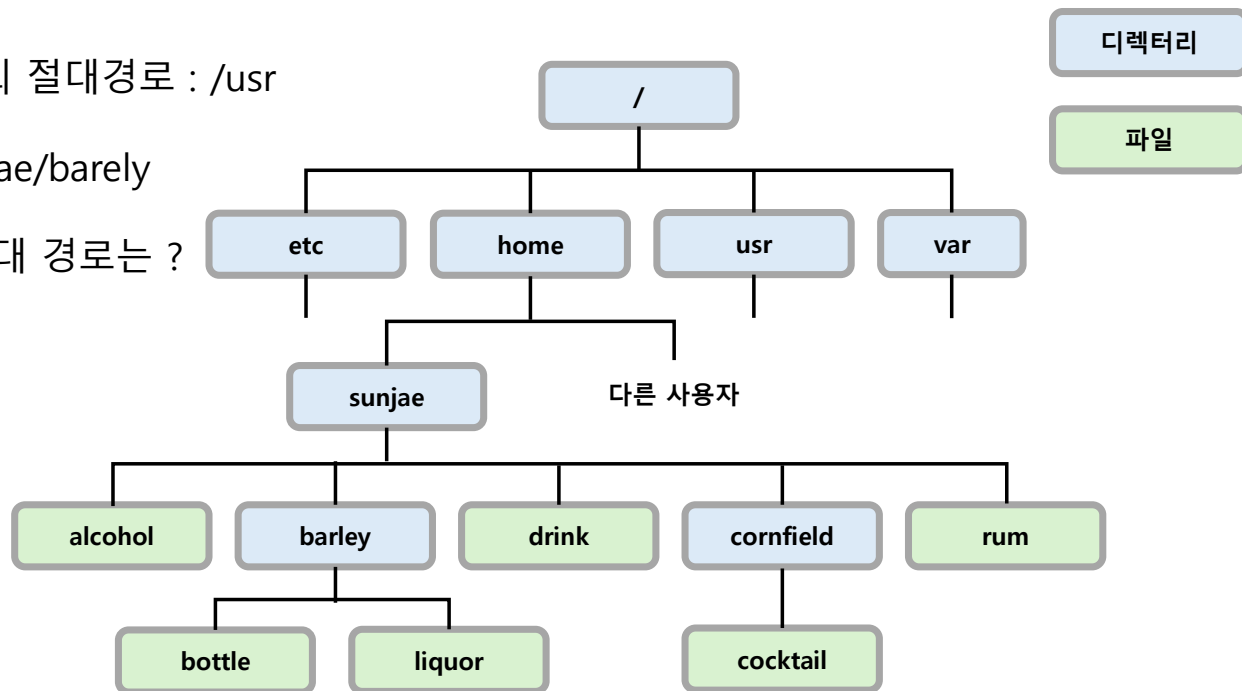
- 사용자는 반드시 트리 구조 어딘가에 존재하는 디렉터리에서 작업을 수행
- 사용자가 지금 위치하고 있는 디렉터를 **현재 디렉터리(current directory)**
- 로그인 시, 홈 디렉터리라고 부르는 개인 전용 디렉터리가 현재 디렉터리가 된다
- 홈 디렉터리 아래는 **서브 디렉터리**
- 현재 디렉터리 있는 파일과 디렉터리만 사용자가 직접 조작 가능하므로 현재 디렉터리에 없는 (외부) 파일과 디렉터리는 다음과 같은 방법으로 사용
  - a. 조작 대상이 존재하는 디렉터리로 이동하기
  - b. 조작 대상을 지정할 때 대상이 존재하는 디렉터리도 지정하기
- a, b 어느 방법이든 대상 파일이나 디렉터리가 트리 내부 어디에 존재하는지 알려주는 장소(디렉터리) 정보가 필요하며 이렇게 다른 장소로 향하는 길을 패스(경로)라 함

▪ 경로 지정에는 **절대 경로**와 **상대 경로** 방식이 있다.

# 파일과 디렉터리 개념

## ❖절대 경로

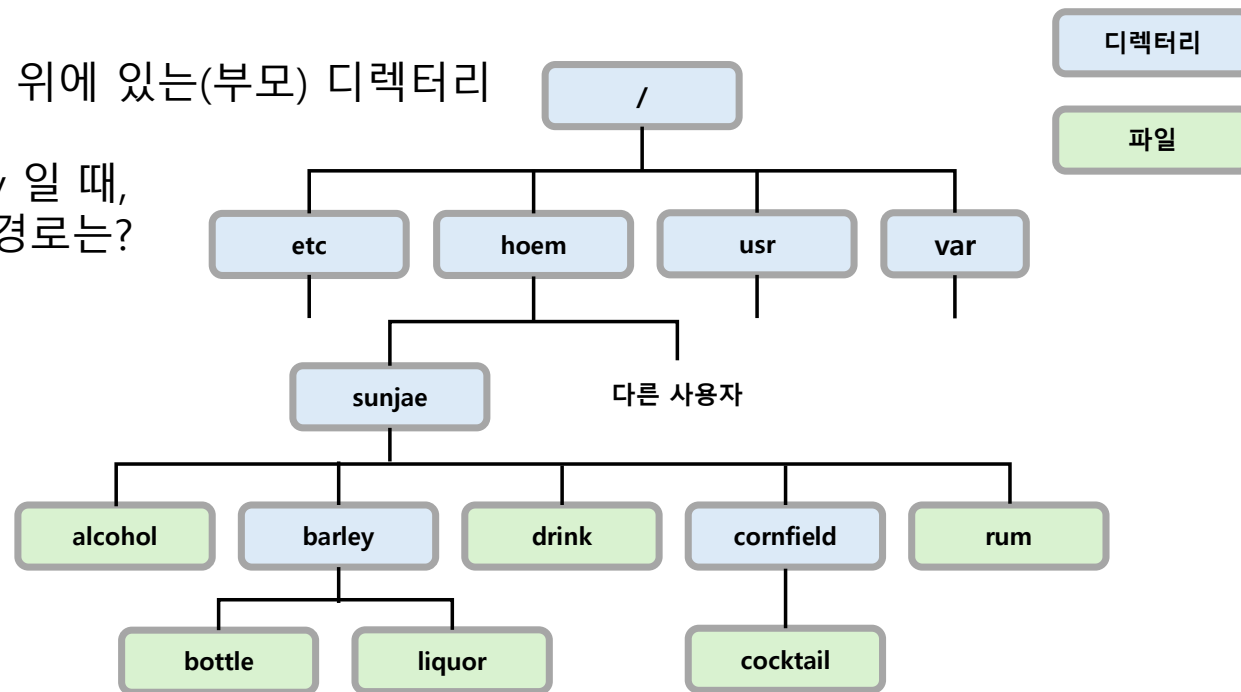
- 루트 디렉터리에서 시작해서 목적지 디렉터리로 향하는 경로(현재 디렉터리가 어디인지 관계없이 해당 파일을 지정하는 경로가 동일)
- 루트 디렉터리 지정 방법 : /
- 루트 디렉터리 아래에 있는 usr 디렉터리의 절대경로 : /usr
- barley 디렉터리의 절대경로 : /home/sunjae/barely
- barely 디렉터리 안에 있는 bottle 파일 절대 경로는 ?



# 파일과 디렉터리 개념

## ❖상대 경로

- 현재 자신의 위치, 즉 현재 디렉터리에서 목적지를 향하는 경로
- 현재 디렉터리가 다르면 목적 디렉터리가 같더라도 경로는 다르다!
- .(닷, dot)이나 .. 같은 기호를 사용하여 표현
- .은 현재 디렉터리를 뜻하고 ..은 1단계 위에 있는(부모) 디렉터리
- 현재 디렉터리가 /home/sunjae/barely 일 때, usr 디렉터리와 bottle 파일의 상대 경로는?





# Basic Command

## ❖ 명령어의 옵션과 인수

- 파일이나 디렉터리 조작, 애플리케이션 실행, 시스템 설정, 정보 표시 같은 대부분의 동작은 명령어 입력으로 가능!!

\$ 명령어

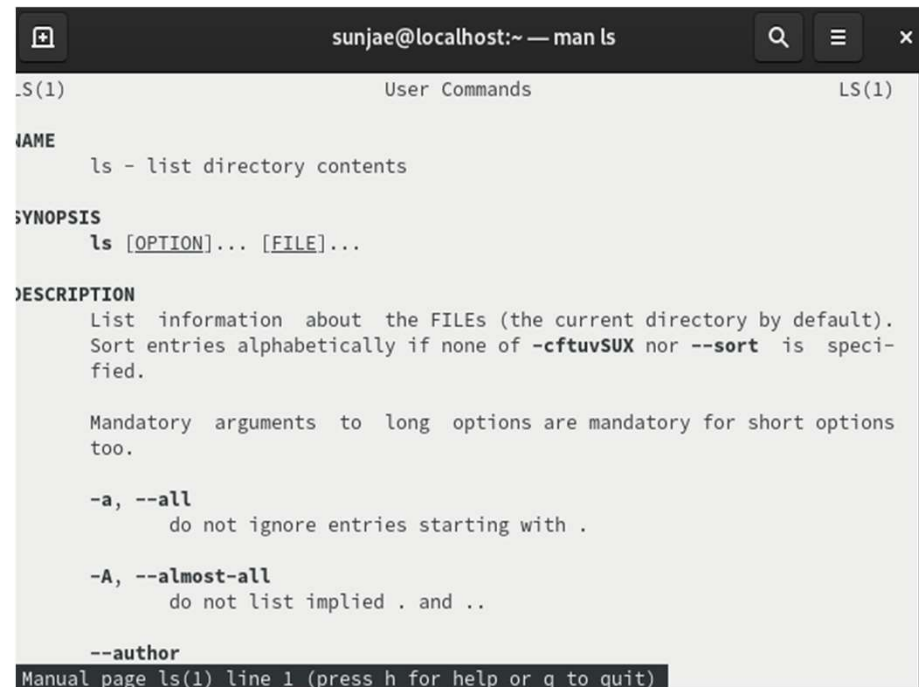
\$ 명령어 [옵션(인수)][옵션(인수)]...

- 옵션은 명령어가 제공하는 부가 기능을 실행
  - 명령어가 어떤 식으로 동작하는지 제어하기도 결과 출력 형식을 변경하기도 한다
  - 옵션은 -(하이픈) 뒤에 알파벳이 따라오는 형식
- 인수는 명령어나 옵션에 넘기는 값
  - 입력한 값에 대응하는 명령어 실행 결과를 얻을 수 있다

# Basic Command

## ❖ 도움말

- 명령어 사용법을 설명하는 도움말(manual)이 존재
- `man ls` 입력 후 Enter! (ls 명령어는 뒤에서 설명)
- 명령어 입력 시, 명령어가 제공하는 기능, 사용 방법 개요, 지정 가능한 옵션, 인수 관련 설명이 화면에 표시
- 참조를 종료하려면 "q" 키를 입력
- 또한 `--help` 옵션과 함께 명령어도 실행해보자!



```
sunjae@localhost:~ — man ls
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  List information about the FILES (the current directory by default).
  Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      show author names for files with sUID or sGID bits set

Manual page ls(1) line 1 (press h for help or q to quit)
```

# Basic Command

## ❖ 현재 경로와 디렉토리 이동

- pwd : 현재 경로 보기
  - pwd 명령은 현재 작업중인 디렉토리의 절대 경로를 보여준다
  - Syntax : pwd
- cd : 원하는 디렉토리로 이동한다
  - 디렉토리간 이동명령
  - Syntax : cd [directory]

인자값	의미
Directory	이동하기 원하는 디렉토리
.	현재 디렉토리
..	상위 디렉토리
\$변수	변수에 지정된 정보를 이용하여 디렉토리 이동
~	로그인 된 사용자의 홈 디렉토리로 이동
~계정명	지정된 계정의 홈 디렉토리로 이동

# Basic Command

## ❖파일 목록/내용 조회 관련 명령어

- ls : 디렉토리의 목록 보기
  - 도스의 dir과 같은 역할을 하며, 해당 디렉토리에 있는 내용을 출력한다
  - Syntax : ls [ option ] [directory / file ]
- option

옵션	의미
<b>-a</b>	.을 포함한 경로안의 모든 파일과 디렉토리 표시
<b>-l</b>	지정한 디렉토리의 내용을 자세히 출력
-d	지정된 디렉토리의 정보 출력
-n	파일 및 디렉토리 정보 출력시 UID, GID를 사용
-F	파일 형식을 알리는 문자를 각 파일 뒤에 추가 "*", "/", "@", " ", "=", "NULL"
-R	하위 경로와 그 안에 있는 모든 파일들도 같이 나열

# Basic Command

## ❖파일 목록/내용 조회 관련 명령어

- cat : 파일 내용 출력
  - 텍스트 파일 내용을 표준 출력장치로 출력하는 명령
  - Syntax : cat [ file ]
  - 파일 내용 출력하기

```
[root@localhost sunjae]# cat /etc/passwd
```

- 기존의 파일 내용을 다른 파일로 입력하기

```
[root@localhost sunjae]# cat /etc/passwd > /testfile
```

- 기존 파일에 내용 추가하기

```
[root@localhost sunjae]# cat >> /testfile
```

# Basic Command

## ❖파일 목록/내용 조회 관련 명령어

- head : 파일 내용 출력
  - 파일의 내용중 처음부터 아래로 10줄 출력
  - head 명령어만 사용시 기본값이 위에서 10줄을 출력

```
[root@localhost sunjae]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

- head -<n> /etc/passwd

```
[root@localhost sunjae]# head -1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

# Basic Command

## ❖파일 목록/내용 조회 관련 명령어

### ▪ tail

- 파일의 내용중 마지막부터 위로 10줄 출력
- tail 명령어만 사용시 기본값인 아래에서 10줄을 출력

```
[root@localhost sunjae]# tail /etc/passwd
setroubleshoot:x:987:987:SELinux troubleshoot server:/var/lib/setroubleshoot:/usr/sbin/nologin
pipewire:x:986:986:PipeWire System Daemon:/run/pipewire:/usr/sbin/nologin
flatpak:x:985:985:User for flatpak system helper:/usr/sbin/nologin
gdm:x:42:42:/var/lib/gdm:/usr/sbin/nologin
gnome-initial-setup:x:984:983:/run/gnome-initial-setup:/usr/sbin/nologin
chrony:x:983:982:chrony system user:/var/lib/chrony:/usr/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
dnsmasq:x:982:981:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
tcpdump:x:72:72:/usr/sbin/nologin
sunjae:x:1000:1000:sunjae:/home/sunjae:/bin/bash
```

- tail -<n> /etc/passwd

```
[root@localhost sunjae]# tail -1 /etc/passwd
sunjae:x:1000:1000:sunjae:/home/sunjae:/bin/bash
```

# Basic Command

## ❖파일 목록/내용 조회 관련 명령어

- more : 내용이 많은 파일을 화면단위로 끊어서 출력
  - more /etc/passwd
  - ls 명령어와 조합해서 사용도 많이 한다 (ls -l /etc | more)

```
[root@localhost sunjae]# ls -l /etc | more
total 1304
drwxr-xr-x. 3 root root      28 Jul 22 13:25 accountsservice
-rw-r--r--. 1 root root      16 Jul 22 13:31 adjtime
-rw-r--r--. 1 root root    1529 Jun 23  2020 aliases
drwxr-xr-x. 3 root root      65 Jul 22 13:27 alsa
drwxr-xr-x. 2 root root    4096 Jul 22 13:28 alternatives
-rw-r--r--. 1 root root     541 Nov 30  2023 anacrontab
-rw-r--r--. 1 root root     833 Feb 11  2023 appstream.conf
-rw-r--r--. 1 root root      55 Jun 17 21:52 asound.conf
-rw-r--r--. 1 root root       1 Apr  4  2022 at.deny
drwxr-x---. 4 root root     100 Jul 22 13:26 audit
drwxr-xr-x. 3 root root    4096 Jul 22 13:31 authselect
drwxr-xr-x. 4 root root       71 Jul 22 13:25 avahi
drwxr-xr-x. 2 root root     124 Jul 22 13:28 bash_completion.d
-rw-r--r--. 1 root root    2658 Feb  7  2024 bashrc
-rw-r--r--. 1 root root     535 Mar  5 18:24 bindresvport.blacklist
drwxr-xr-x. 2 root root        6 Jun 19 20:35 binfo.d
drwxr-xr-x. 2 root root      23 Jul 22 13:25 bluetooth
-rw-r-----. 1 root brlapi    33 Jul 22 13:27 brlapi.key
drwxr-xr-x. 7 root root       84 Jul 22 13:27 brltty
-rw-r--r--. 1 root root  28974 Aug 10  2021 brltty.conf
-rw-r--r--. 1 root root       24 Mar 21 02:58 centos-release
drwxr-xr-x. 3 root root       36 Jul 22 13:27 chromium
-rw-r--r--. 1 root root   1371 Dec  5  2023 chrony.conf
-rw-r-----. 1 root chrony   540 Jun 19 15:23 chrony.keys
drwxr-xr-x. 2 root root       26 Jul 22 13:26 cifs-utils
drwxr-xr-x. 4 root root       66 Jul 22 13:26 cockpit
--More--
```

space : 다음페이지  
b : 이전 페이지  
enter : 한 줄 씩 넘기기  
/word : word 검색  
q : 종료



# Basic Command

## ❖파일 목록/내용 조회 관련 명령어

- grep : 파일에서 내가 원하는 단어를 찾는다
  - Syntax : grep [-option ] [ word ] [ file or directory ]
  - -n 옵션을 주어 몇 번째 라인에 있는지 검색

```
GNU nano 5.6.1      introduce.txt
hello
my name is Cent
Nice to meet you
bte~
```

grep -n Cent introduce.txt

```
[sunjae@localhost ~]$ grep -n Cent introduce.txt
2:my name is Cent
```

- -r 옵션을 주어 하위 디렉토리까지 포함하여 검색

```
[sunjae@localhost ~]$ grep -r "hello" /home
/home/sunjae/process.sh:      echo "hello, pid $$"
/home/sunjae/test.sh:printf("hello");
/home/sunjae/introduce.txt:hello
```

# Basic Command

## ❖파일 목록/내용 조회 관련 명령어

- find : 지정된 디렉토리 아래의 모든 파일과 디렉토리를 검색하고 출력
  - Syntax : find [ path ] [ option ] [expression]
  - 파이프라인( | )과 grep 명령어와 함께 자주 쓰인다(ex. find . -name "\*.log" | xargs grep "error" )

옵션	의미
-name	파일 이름이 지정한 패턴과 일치하는 파일을 찾는다
-iname	대소문자를 구분하지 않고 찾는다
-type	파일 타입을 지정하여 검색한다
-size	파일 크기가 지정한 크기와 일치하는 파일을 찾는다
-mtime	마지막으로 수정된 시점을 기준으로 파일을 찾는다
-atime	마지막으로 접근한 시점을 기준으로 파일을 찾는다
-ctime	마지막으로 상태가 변경된 시점을 기준으로 파일을 찾는다

# Basic Command

## ❖파일 생성/복사/이동/삭제 관련 명령어

- touch : 빈 파일을 생성한다
  - Syntax : touch [filename]
- nano : 유닉스와 리눅스 기반 시스템에서 사용할 수 있는 간단한 텍스트 에디터
  - sudo yum install nano 명령어로 설치 가능
  - Syntax : nano [filename]

단축키	의미
Ctrl + x	에디터 종료
Ctrl + o	파일 저장
Ctrl + w	검색(윈도우 ctrl + f)
Ctrl + k	텍스트 자르기(커서 있는 라인 삭제)
Ctrl + u	자른 텍스트 붙여넣기
Ctrl + j	현재 행 정렬



# Basic Command

## ❖파일 생성/복사/이동/삭제 관련 명령어

- mkdir : 디렉토리 생성
  - Syntax : mkdir [directory]
  - -p 옵션 시 하위 디렉토리까지 생성 가능
  
- rmdir : 비어있는 디렉토리 삭제
  - 디렉토리를 삭제하는 명령
  - Syntax : rmdir [ directory ]

# Basic Command

## ❖파일 생성/복사/이동/삭제 관련 명령어

- cp(copy) : 파일 / 디렉토리 복사
  - 파일이나 디렉토리를 복사하는 명령어
  - Syntax : cp [ -option ] [ sources ] [ target ]
- option

옵션	의미
-i	복사대상 파일이 있을 경우, 사용자에게 복사에 대한 실행 여부를 묻는다
-f	복사대상 파일이 있을 경우, 사용자에게 확인없이 강제로 복사한다
-r	디렉토리를 복사할 경우 하위 디렉토리나 파일을 모두 복사한다
-v	복사진행 상태를 출력한다
-d	복사대상 파일이 심볼릭 파일이면, 심볼릭 정보를 그대로 유지한 상태로 복사한다
-p	원본 파일의 소유주, 그룹, 권한, 시간정보를 보존하여 복사한다
-a	원본 파일의 속성, 링크 정보들을 그대로 유지하면서 복사한다

# Basic Command

## ❖파일 생성/복사/이동/삭제 관련 명령어

- mv : 파일 / 디렉토리 이동
  - 파일이나 디렉토리를 이동하거나 이름을 바꾼다
  - Syntax : mv [ -option ] [ sources ] [ target ]
- option

옵션	의미
-u	이동할 파일이 이동할 위치에 있는 파일보다 최근 파일일 경우에만 이동한다
-b	대상 파일이 이미 있어, 지워지는 것을 대비해 백업파일을 생성한다
-f	대상 파일이 이미 있어도 사용자에게 어떻게 처리할지를 묻지 않는다
-v	파일을 옮기는 과정을 자세하게 보여준다
-S	-b 옵션을 이용하여 백업할 경우 백업파일에서 사용할 파일 이름의 꼬리 문자를 지정한다

# Basic Command

## ❖파일 생성/복사/이동/삭제 관련 명령어

- rm : 파일 및 디렉토리 삭제
  - 파일이나 디렉토리를 삭제하는 명령(권한이 있을 경우)
  - Syntax : rm [ - option ] [ directory / file ]
- option

옵션	의미
-f	파일 / 디렉토리 삭제시 사용자에게 어떻게 처리할지 묻지 않는다
-r	일반 파일이면 그냥 지우고, 디렉토리면 디렉토리를 포함한 하위 경로와 파일을 모두 지운다
-v	각각의 파일 지우는 정보를 자세하게 보여준다

# Basic Command

## ❖파일 시스템 용량 확인 명령어

- df : 파일 시스템의 디스크 사용량을 표시
  - 파일 시스템의 총 용량, 사용된 용량, 남은 용량 등을 확인
  - Syntax : df [ - option ] [ directory ]
- option

옵션	의미
-h	사람이 읽기 쉬운 형태로 출력 (KB, MB, GB 등 단위)
-a	모든 파일 시스템을 표시
-T	파일 시스템의 타입도 함께 표시



# Basic Command

## ❖파일 모드(권한) 관련 명령어

### ▪ chmod : 파일 압축/해제

- 파일에 저장된 데이터를 읽기 (r = read)
- 파일에 데이터를 쓰기 (w = write)
- 파일 실행 (x = execute)



읽기 (r)	⇒	4	---	: 0 (0+0+0)
쓰기 (w)	⇒	2	r--	: 4 (4+0+0)
실행 (x)	⇒	1	-w-	: 2 (0+2+0)
권한 없음 (-)	⇒	0	--x	: 1 (0+0+1)
			rwx	: 7 (4+2+1)

# Basic Command

## ❖파일 모드(권한) 관련 명령어

- 스크립트 파일 경우 초기 생성 시 실행 권한 x
  - 프로세스 관련 명령어 실행 시 필요하니 숙지

```
[sunjae@localhost ~]$ ls -l
total 156
drwxr-xr-x. 2 sunjae sunjae   6 Jul 23 12:42 barley
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Desktop
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Documents
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Downloads
-rw-r--r--. 1 root  root  71426 Aug 14 15:26 filelist.gz
drwxr-xr-x. 2 sunjae sunjae   33 Aug 23 15:50 gcc_practice
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Music
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Pictures
-rwxr-xr-x. 1 sunjae sunjae   64 Aug 27 11:26 process.sh
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Public
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Templates
drwxr-xr-x. 3 sunjae sunjae   30 Aug 24 17:36 test
drwxr-xr-x. 4 root  root    55 Aug 14 16:59 testdir
-rw-r--r--. 1 sunjae sunjae   29 Aug 27 14:54 test.sh
-rw-r--r--. 1 root  root  71799 Aug 14 16:58 test.tar.gz
-rw-r--r--. 1 sunjae sunjae    1 Aug 14 11:34 test.txt
drwxr-xr-x. 2 sunjae sunjae   6 Jul 22 14:18 Videos
```

# Basic Command

## ❖ 권한 및 계정 변경

- sudo

- 다른 사용자의 권한을 사용하여 명령을 실행하도록 하는 명령어
- "sudo 명령어"는 root 권한으로 명령어를 실행
- /etc/sudoers 파일에 지정되어 있는 사용자만 사용가능

- su

- 다른 사용자로 로그인하는 명령어
- su 사용자명으로 사용 ( "su"는 "su root"와 동일 )
- 로그인 하려는 계정의 패스워드가 필요

# Basic Command

## ❖ 압축/해제 관련 명령어

- gzip : 파일 압축
  - 파일을 압축해준다
  - Syntax : gzip [ filename ]
- gunzip : 파일 압축
  - 압축 파일을 해제해준다
  - Syntax : gunzip [ filename ]

# Basic Command

## ❖ 압축/해제 관련 명령어

### ▪ tar : 파일 압축/해제

- 옵션값에 따라 파일을 압축하거나 해제시켜준다
- Syntax : tar -czf [ archive\_name.tar.gz ] [ file1 ] [ directory1 ] . . . or tar -zxf [ archive\_name.tar.gz ]

### ▪ 실습

- find 명령어 입력 시 출력되는 내용을 리다이렉션하여 filelist.txt 파일로 저장
- gzip 명령어로 filelist.txt 파일 압축
- gzip 명령어로 생성된 압축 파일과 다른 디렉토리 두 개 묶어서 압축
- 새로운 디렉토리 생성
- 생성된 디렉토리에 압축 파일을 풀어서 확인해보기

-c : 파일을 tar로 묶음  
-z : gzip으로 압축하거나 해제  
-f : 파일 이름을 지정  
-x : tar 압축을 풀기

# Basic Command

## ❖ 날짜와 시간

- date : 현재 날짜와 시간을 출력

```
[sunjae@localhost ~]$ date  
Tue Aug 27 04:26:22 PM KST 2024
```

## ❖ 화면 지우기

- clear : 화면을 깨끗하게 지우기

```
drwxr-xr-x. 2 sunjae sunjae 33 Aug 23 15:50 gcc_practice  
drwxr-xr-x. 2 sunjae sunjae 6 Jul 22 14:18 Music  
drwxr-xr-x. 2 sunjae sunjae 6 Jul 22 14:18 Pictures  
-rw-r--r--. 1 sunjae sunjae 64 Aug 27 11:26 process.sh  
drwxr-xr-x. 2 sunjae sunjae 6 Jul 22 14:18 Public  
drwxr-xr-x. 2 sunjae sunjae 6 Jul 22 14:18 Templates  
drwxr-xr-x. 3 sunjae sunjae 30 Aug 24 17:36 test  
drwxr-xr-x. 4 root root 55 Aug 14 16:59 testdir  
-rw-r--r--. 1 root root 71799 Aug 14 16:58 test.tar.gz  
-rw-r--r--. 1 sunjae sunjae 1 Aug 14 11:34 test.txt  
drwxr-xr-x. 2 sunjae sunjae 6 Jul 22 14:18 Videos  
[sunjae@localhost ~]$ cd /  
[sunjae@localhost /]$ ls  
afs boot etc lib media opt root sbin sys tmp var  
bin dev home lib64 mnt proc run srv testfile usr  
[sunjae@localhost /]$ cd ~  
bash: cd: ~/: invalid option  
cd: usage: cd [-L|[-P [-e]] [-@]] [dir]  
[sunjae@localhost /]$ cd ~  
[sunjae@localhost ~]$ ls  
barley Downloads Music Public testdir test.txt  
Desktop filelist.gz Pictures Templates test.sh  
Documents gcc_practice process.sh test test.tar.gz  
[sunjae@localhost ~]$ clear
```



```
[sunjae@localhost ~]$
```

# Basic Command

## ❖ 문자열 출력 명령어

- echo : 문자열이나 변수 값을 화면에 출력
  - 쉘 스크립트에서 메시지 출력, 변수 값 확인 등에 사용
  - Syntax : echo [옵션] [문자열]

- option

옵션	의미
-n	줄 바꿈 없이 출력
-e	이스케이프 문자를 해석하여 출력

# Basic Command

## ❖ 실습 (파일 다루기)

- test1.txt 파일 생성
- test1.txt 파일명을 apple.txt로 변경
- apple.txt 파일을 /tmp 경로로 이동시키고 파일명은 pineapple.txt로 변경



# Basic Command

## ❖파일 생성/복사/이동/삭제 관련 명령어

- testdir1 디렉토리 생성
- testdir1 디렉토리에 haha.txt 파일 생성
- testdir1 디렉토리 삭제
- testdir2 디렉토리 생성
- testdir2 디렉토리를 /tmp 경로로 이동
- 다시 기존 경로로 디렉토리를 이동

# Basic Command

## ❖ Yum (Yellodog Update Modified)

- 레드햇 계열의 리눅스 배포판에서 사용하는 프로그램(패키지) 설치 관리 도구
- 패키지 의존성 문제를 자동으로 처리하면서 설치, 업데이트, 삭제를 진행
- 패키지 의존성 (package dependency)



C 패키지를 설치 하기 위해서는, A와 B 패키지를 순서대로 설치 한 후에 설치가 가능하다.

# Basic Command

## ❖ Yum (Yellodog Update Modified)

yum	[옵션]	명령어	패키지명
-----	------	-----	------

패키지 list 확인

```
[choi_gunhee@localhost ~]$ yum list
```

xxhash-libs.x86_64	0.8.1-3.el9	appstream
xz-devel.i686	5.2.5-8.el9	appstream
xz-devel.x86_64	5.2.5-8.el9	appstream
xz-java.noarch	1.8-14.el9	appstream
xz-libs.i686	5.2.5-8.el9	baseos
xz-lzma-compat.x86_64	5.2.5-8.el9	appstream
yajl.i686	2.1.0-21.el9	appstream
yara.i686	4.2.1-4.el9	appstream
yara.x86_64	4.2.1-4.el9	appstream
yelp-libs.i686	2:40.3-2.el9	appstream
yum-utils.noarch	4.1.0-2.el9	baseos
zchunk-libs.i686	1.1.9-4.el9	baseos
zchunk-libs.x86_64	1.1.9-4.el9	baseos
zlib.i686	1.2.11-33.el9	baseos
zlib-devel.i686	1.2.11-33.el9	appstream
zlib-devel.x86_64	1.2.11-33.el9	appstream
zram-generator.x86_64	0.3.2-7.el9	appstream
zsh.x86_64	5.8-9.el9	baseos
zziplib.i686	0.13.71-9.el9	appstream
zziplib.x86_64	0.13.71-9.el9	appstream
zziplib-utils.x86_64	0.13.71-9.el9	appstream

```
[choi_gunhee@localhost ~]$
```

# Basic Command

## ❖ Yum (Yellodog Update Modified)

- vim 설치
  - 옵션 y : 사용자에게 yes/no 여부를 묻지 않고 넘어감

```
[choi_gunhee@localhost ~]$ yum -y install vim
```

```
Error: This command has to be run with superuser privileges (under the root user on most systems).
```

```
[choi_gunhee@localhost ~]$
```

- 위 에러는 vim 설치 시 root 권한이 필요하다는 것을 알려준다.

```
[choi_gunhee@localhost ~]$ sudo yum -y install vim
```

```
We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:
```

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for choi_gunhee: █
```

```
[sudo] password for choi_gunhee:
```

```
Last metadata expiration check: 0:21:28 ago on Sun 14 Aug 2022 08:34:54 PM KST.
```

```
Package vim-enhanced-2:8.2.2637-16.el9.x86_64 is already installed.
```

```
Dependencies resolved.
```

```
Nothing to do.
```

```
Complete!
```

```
[choi_gunhee@localhost ~]$
```

# Basic Command

## ❖ Yum (Yellodog Update Modified)

- yum check-update : 업데이트 가능한 패키지 목록 확인

```
[choi_gunhee@localhost ~]$ yum check-update
Last metadata expiration check: 1:08:57 ago on Sun 14 Aug 2022 07:48:59 PM KST
.
[choi_gunhee@localhost ~]$
```

- yum update 패키지명  
yum update

- 패키지명을 입력하면 해당 패키지 업데이트, 안한다면 모든 패키지 업데이트

```
[choi_gunhee@localhost ~]$ sudo yum update
Last metadata expiration check: 0:24:26 ago on Sun 14 Aug 2022 08:34:54 PM KST
.
Dependencies resolved.
Nothing to do.
Complete!
[choi_gunhee@localhost ~]$
```

# Basic Command

## ❖ Yum (Yellodog Update Modified)

- yum remove 패키지명 : 해당 패키지를 삭제

```
[choi_gunhee@localhost ~]$ sudo yum remove vim
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
Removing:				
vim-enhanced	x86_64	2:8.2.2637-16.el9	@AppStream	3.8 M
Removing unused dependencies:				
gpm-libs	x86_64	1.20.7-29.el9	@AppStream	28 k
vim-common	x86_64	2:8.2.2637-16.el9	@AppStream	30 M
vim-filesystem	noarch	2:8.2.2637-16.el9	@anaconda	40

### Transaction Summary

```
Remove 4 Packages
```

```
Freed space: 34 M
```

```
Is this ok [y/N]: █
```

# 프로세스

## ❖ 프로세스란?

- 시스템에서 실행 중인 프로그램
- 리눅스는 기본적으로 다중 프로세스 시스템

## ❖ 프로세스의 부모-자식 관계

- 리눅스의 모든 프로세스는 부모 - 자식 관계 (ex. 쉘에서 nano 실행 -> 부모 : 쉘, 자식 : nano)
- 자식 프로세스는 할 일이 끝나면 부모 프로세스에게 결과를 돌려주고 종료

## ❖ 프로세스의 번호(PID)

- 각 프로세스의 고유 번호로, 1번부터 시작해서 프로세스 실행마다 증가된다

## ❖ 프로세스 종료

- 데몬 프로세스 : 특정 서비스 제공용으로 평소에 대기하다가 요청 시 동작. 커널에 의해 실행
- 고아 프로세스 : 부모 프로세스가 먼저 종료된 경우, 1번 프로세스가 부모가 되어 종료를 돕는다
- 좀비 프로세스 : 종료되었는데도 프로세스 테이블 목록에 있는 경우. 부모 프로세스가 종료 정보를 제대로 처리하지 않아 발생하며 부모를 종료시켜 고아 프로세스로 만들어 처리

# Basic Command

## ❖ 프로세스 관련 명령어

- ps : 프로세스 목록 출력
  - 실행 중인 프로세스의 목록을 출력해준다
  - Syntax : ps [ -option ]
  - 현재 프로세스의 ID(PID)를 출력하는 명령어 = echo \$\$
- option

옵션	의미
-a	현재 실행 중인 모든 프로세스 출력
-e	모든 프로세스 정보 표시
-f	전체 포맷으로 출력



# Basic Command

## ❖ 프로세스 관련 명령어

ps 실행 결과

```
[sunjae@localhost ~]$ ps
```

PID	TTY	TIME	CMD
2660	pts/0	00:00:00	bash
2704	pts/0	00:00:00	ps

ps -f 실행 결과

```
[sunjae@localhost ~]$ ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
sunjae	2660	2642	0	10:44	pts/0	00:00:00	bash
sunjae	2819	2660	0	10:46	pts/0	00:00:00	ps -f

ps -ef 실행 결과

```
[sunjae@localhost ~]$ ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	10:44	?	00:00:01	/usr/lib/systemd/systemd rhg
root	2	0	0	10:44	?	00:00:00	[kthreadd]
root	3	2	0	10:44	?	00:00:00	[pool_workqueue_]
root	4	2	0	10:44	?	00:00:00	[kworker/R-rcu_g]
root	5	2	0	10:44	?	00:00:00	[kworker/R-rcu_p]
root	6	2	0	10:44	?	00:00:00	[kworker/R-slub_]
root	7	2	0	10:44	?	00:00:00	[kworker/R-netns]
root	8	2	0	10:44	?	00:00:00	[kworker/0:0-cgroup_destroy]
root	9	2	0	10:44	?	00:00:00	[kworker/0:0H-events_highpri]
root	10	2	0	10:44	?	00:00:00	[kworker/u8:0-events_unbound]
root	11	2	0	10:44	?	00:00:00	[kworker/R-mm_pe]
root	12	2	0	10:44	?	00:00:00	[kworker/u8:1-netns]
root	13	2	0	10:44	?	00:00:00	[rcu_tasks_kthre]
root	14	2	0	10:44	?	00:00:00	[rcu_tasks_rude_]
root	15	2	0	10:44	?	00:00:00	[rcu_tasks_trace]
root	16	2	0	10:44	?	00:00:00	[ksoftirqd/0]
root	17	2	0	10:44	?	00:00:00	[rcu_preempt]
root	18	2	0	10:44	?	00:00:00	[migration/0]
root	19	2	0	10:44	?	00:00:00	[idle_inject/0]
root	20	2	0	10:44	?	00:00:00	[kworker/0:1-events]
root	21	2	0	10:44	?	00:00:00	[cpuhp/0]

# Basic Command

## ❖ 실습(PID 출력)

- 3초마다 hello, pid \*\*\*\* 를 출력하는 sh를 작성해보자

### Hint code

nano process.sh (스크립트 생성)

```
#!/bin/bash
```

```
while [ ? ];do  
    echo "hello, pid ??"  
    sleep ?  
done
```

```
[sunjae@localhost ~]$ ./process.sh  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326  
hello, pid 3326
```

# Basic Command

## ❖ 프로세스 관련 명령어

- kill : 프로세스 종료
  - PID를 확인 후 kill 명령어로 종료
  - Syntax : kill [ -option ]
  - -9는 옵션은 프로세스를 강제 종료(프로세스가 정상 종료가 안될 경우 사용)
  - -15 옵션은 자신이 하던 작업을 모두 안전하게 종료하는 절차를 밟으며 종료
- 실습
  - ps 명령어로 현재 실행중인 프로세스 PID 확인
  - sleep 100 & 입력 (테스트용 백그라운드 프로세스)
  - ps 명령어로 sleep 프로세스의 PID 확인
  - sleep 프로세스 종료하기

# Basic Command

## ❖ IPC 상태 확인 명령어

- ipcs : 시스템의 IPC (Inter-Process Communication) 자원 상태를 출력
- ipcmk: 새로운 IPC 리소스를 생성
- lpcrm: 기존의 IPC 리소스를 삭제

- option

옵션	의미
-m/M	공유 메모리 정보 출력
-q/Q	메시지 큐 정보 출력
-s/S	세마포어 정보 출력
-a	모든 IPC 자원 정보 출력

# Basic Command

## ❖ 네트워크 상태 확인 명령어

- netstat : 네트워크 연결, 라우팅 테이블, 네트워크 인터페이스 상태 등을 출력
  - sudo yum install net-tools
  - 네트워크 진단 및 트러블슈팅에 사용
  - Syntax : netstat [옵션]
- option

옵션	의미
-a	모든 네트워크 연결과 포트를 표시
-n	주소와 포트를 숫자 형태로 표시
-t	TCP 연결 상태만 출력
-u	UDP 연결 상태만 출력