

리눅스

리눅스의 개념과 구조

Linux 란?

컴퓨터 OS 커널의 일종인 리눅스 커널, 또는 리눅스 커널을 사용하는 운영체제를 가리키는 말이기도 하다.

소스 코드가 공개되어 있는 대표적인 오픈소스 S/W이고 컴퓨터 역사상 가장 많은 참여자가 관여하고 있는 오픈 소스 프로젝트이다. 모바일 운영체제인 안드로이드 역시 리눅스 커널을 가져다가 쓰고 있다.

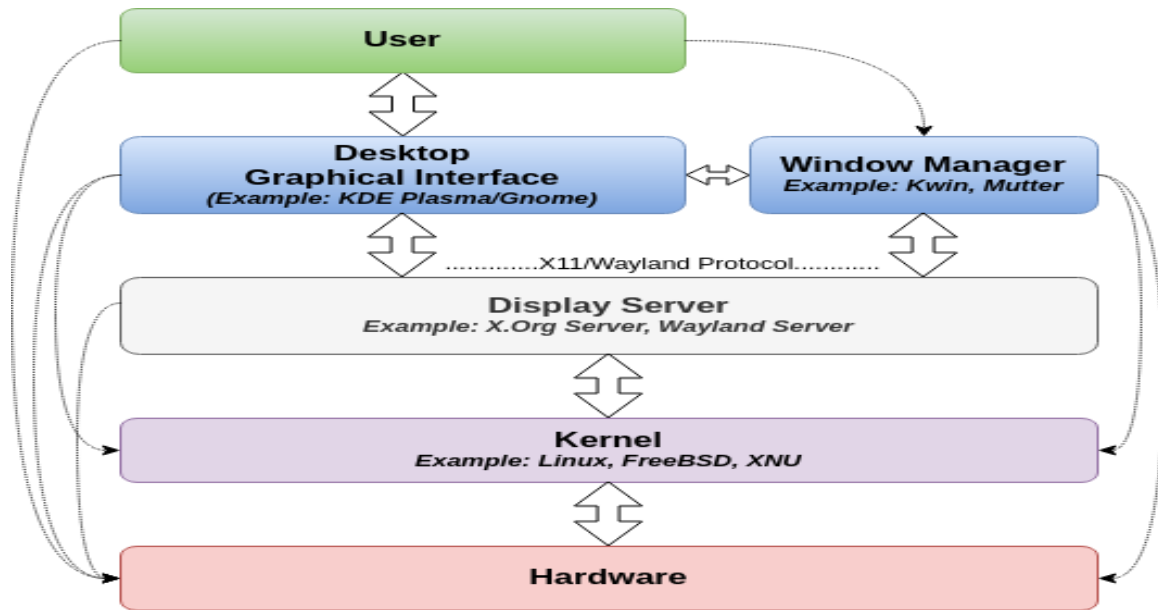
*커널?

Linux® 커널은 Linux 운영 체제(OS)의 주요 구성 요소이며 컴퓨터 하드웨어와 프로세스를 잇는 핵심 인터페이스입니다. 그리고 두 가지 관리 리소스 사이에서 최대한 효과적으로 통신합니다.(하드웨어와의 통신을 담당함)

리눅스 구성요소 <참조 : 위키백과

<https://namu.wiki/w/Linux?from=%EB%A6%AC%EB%88%85%EC%8A%A4>>

- **커널**: 하드웨어와의 통신을 담당한다.
- 디스플레이 서버^[14]: 클라이언트와의 입출력을 담당하며 화면을 표시하는 기초적인 부분이다. (X11, Wayland 등)
- **윈도우 매니저**: 커널과 창, 그리고 창과 창 사이의 통신을 관리한다.
- 디스플레이 매니저: 디스플레이 서버와 데스크톱 환경을 시동시키고 사용자 로그인을 관리한다. 쉽게 생각해서 로그인 화면. (GDM, SDDM, LightDM 등) 사용자가 디스플레이 서버나 데스크톱 환경을 선택할 수 있도록 옵션을 제공한다.
- **UI/UX 킷**: 유저 인터페이스 UI를 구성하기 위해 사용되는 프로그래밍 툴이다. 일반적으로 GUI 응용 프로그램을 만들 때 이를 사용한다.
- **데스크톱 환경**: 사용자가 위와 같은 기술적인 내용을 몰라도 사용할 수 있게끔 세트로 묶고 일관된 경험을 제공한다. 일반적으로 운영체제를 설치하고 볼 수 있는 것들이 바로 데스크톱 환경이다.



리눅스 명령어

<기초 명령어>

root 계정

슈퍼계정(admin)으로 접근

\$? 특정계정으로 접근

sudo : 윈도우로 치면 관리자 권한으로 실행 -> permission 관련 오류가 나면 넣는것이 좋을듯

ssh root@ip주소 : 맥에서 리눅스 서버로 접속

shutdown -P now : 종료 == halt -p^C == init 0

shutdown -r now : 재부팅 == reboot == init 6

logout : 로그아웃 == exit

uname -a : 현재 OS정보 출력

hostname : 리눅스설치때 쓴 별명 출력

reboot : 리부트 명령어

halt -p : 완전 셧다운(전원내리기)

리눅스 도움말

man [명령어] : 명령어에 대한 설명 = [명령어] --help = info [명령어]

디렉토리 위치

pwd : 자신의 위치 알아보기

(print working directroy)

ls : 현재 디렉토리의 파일 출력

ls -l : 디렉토리와 파일정보

ls -al : 숨겨진 파일까지도 나오게됨(리눅스에서 파일이름이 . 이 들어가면 숨김파일이라는 뜻)

ls -li : 링크된 파일 정보 출력

cd : 이동디렉토리

/ : 절대 위치 기준

./ : 현재 위치

../ : 현재 위치의 상위

~/ : home 위치 기준

chmod : 권한바꾸기

chmod 773 a.txt : a.txt의 권한을 사용자는 (읽고쓰고실행), 그룹은(읽고쓰고실행), 다른사람은 쓰고실행만 가능함.

chmod o+x a.txt : a.txt의 other에게 실행권한 추가하기

chmod g-r a.txt : a.txt의 group에게 읽기권한 빼기

mkdir : 디렉토리 파일 만들기

touch [파일명] : 파일 만들기

rm -f [파일명] : 파일 지우기

rm -R [디렉토리명] : 디렉토리 지우기

rm : remove

rm [삭제파일 이름] : 파일 삭제

rm -r [디렉토리 이름] : 디렉토리 삭제

rm -f [파일 이름] : 파일 삭제

mv 명령어 (move)

mv [옮길 파일 경로] [옮길 위치] : 이동

cp : copy

cp [복사할 파일 이름] [붙여넣어질 파일 이름] : 복사

cp -r [복사할 디렉토리] [붙여넣어질 디렉토리]

링크파일의 개념 (Link file)

Hard & Soft(Symbolic) Link

Link File

Filename	inode
이력서1.doc	①
조별과제발표.ppt	3
야구동영상.mp4	4
팀발표자료.ppt	3
...	...
library.1.0.so	17
library.2.0.so	18

library.so	10
------------	----

Filesystem

inode	Ref	Data
4	1	0xAABBCC...
①	1	...
3	2	...
2	1	...
...
17	1	...
18		

위의 LinkFile(사람이 보기 편한 파일) <-> Filesystem(물리적 하드디스크에 들어있는 파일, 컴퓨터가 보기 편한 파일)

위 파일들을 inode라는 값으로 서로 1:1매핑하고 있음. inode번호는 파일이 생성될때 유니크하게 부여되는 번호이다. 그래서 링크파일을 읽기, 삭제 등을 하면 해당 inode와 매핑된 inode를 찾아서 읽거나 삭제해준다.

그럼 Reference?? 보통은 1:1 매핑이므로 ref는 늘 1이다.

하지만 Hard Link 라는 개념은 똑같은 inode값을 가진 파일을 하나 이상을 더 만드는 것임. 예를 들어, 조별과제발표.ppt 와 팀발표자료.ppt가 서로 같은 inode값 3을 가지고 있다면 자신을 참조하고 있는 파일이 2개이므로 ref는 2이다. 만약 참조하는 파일이 3개라면 ref도 3이다.

그래서 원본A의 복사본B를 만들게 되면 같은 Inode값에 ref는 2가 되고, 복사한 이후부터는 이 두 파일은 서로에게 영향을 주지 않는다. 해당 파일이 완전히 사라지려면 ref값이 0이 되어야하므로 A를 삭제해도 ref는 1이되어 B는 사라지지 않는다. 대신 입력과 수정은 같은 inode파일이므로 같이 반영된다.

참고로 Hard Link는 dir는 적용되지 않는다(file에만 적용된다.)

Soft Link?

윈도우의 바로가기 개념과 같음. 소프트링크는 링크파일이 링크파일을 참조하는 형식임.

이는 오픈소스 라이브러리에서 많이 활용되는데 예를 들면, 오픈소스 라이브러리를 배포할때는 그 버전의 대표적인 링크로 배포하게 된다(왜냐면 매번 이름을 다르게 링크해서 보내면 매번 다른 이름값을 참조해서 버전업그레이드를 해주어야 하기 때문에 번거롭다.)

그래서 구버전 lib1.0이 있고, 신버전 lib2.0이 있다고 하면 lib이라는 소프트링크를 두어 구버전 lib1.0을 참조했다가, 신버전 lib2.0으로 참조를 바꿔주면 이를 다운받은 개발자는 저절로 업그레이드된 lib를 쓸 수 있다.

Hard Link, Soft Link(SymbolicLink) 만들기

ln : 하드&소프트링크 만들기

소프트링크 만들기

ln -s [원본소스] [링크하고자 하는 타겟]

[순서이해] lib1 생성 -> lib 로 링크 -> lib2 새로 버전 업글 -> lib 기존링크 제거, lib2로 새로 링크

하드링크 만들기

ln [원본소스] [링크하고자 하는 타겟]

ls -li : 링크파일여부 보기 맨 앞에 숫자는 inode번호

cat : 표준입력을 표준출력으로 바꿔줌

cat [파일명] > [복사할 파일명] : 파일을 복사한다. 단, 파일명이 중복되는 것이 있을경우 덮어쓰기된다.

cat [파일명] >> [복사할 파일명] : 파일을 복사한다. 파일명이 중복될 경우 바로 아래 내용이 생성된다.

cat [파일명] | **more** : 파일의 내용이 긴 경우에 화면 단위로 출력해준다.

> : 출력 전환 (표준 출력인 모니터에서 파일이나 기타 장치로 전환)

>> : 특수 출력 (보통 파일로 저장할 때 파일의 뒷부분에 추가되어 덧붙여진다.)

< : 입력 전환 (표준 입력인 키보드에서 파일이나 기타 장치로 전환)

<< : 특수 입력 (제시된 문자열이 입력될 때까지 대기했다가 한 번에 표준 출력으로 보낸다.)

| : 기호 앞선 명령의 출력을 뒤 명령의 입력으로 전환시켜주는 역할을 한다.

head, tail : (앞,뒤 기준) 라인 출력

head -f(-n 숫자) : 앞에 기준으로 10줄 출력(-n 50 으로 하면 50줄 출력)

tail -f(-n 숫자) : 뒤에 기준으로 10줄 출력 (")

grep : 텍스트 파일에서 특정 문자열을 갖는 줄을 찾아서

출력해주는 명령어

grep [검색하고자 하는 문자열] [파일이름:해당 문자열이 있는 파일이름] :

grep -H : 검색하고자 하는 문자열이 있는 파일 이름도 같이 출력

grep -w : 문자열과 한 단어로 일치해야만 출력

grep -v : 문자열과 일치하지 않는 항목 출력

less : 한 화면단위로 출력해주는 명령어

=> 용량이 큰 log파일을 열 때, 자원소모가 적은 less로 열기!

tar : 압축명령어

tar cvfz [압축하고싶은 파일1] [압축하고싶은 파일2] [압축할 파일을 놓을 폴더] : 압축하기

tar xvfz [압축 풀고싶은 파일] : 압축풀기

chown : 소유자와 그룹명을 바꿀 수 있는 명령어

chown [소유자]:[그룹명] [파일이름] : 파일이름의 소유자와 그룹명을 바꾼다(sudo)

find : 파일, 파일내용이나 엄청나게 많은 것이 검색 가능함

find [경로][조건][검색할 내용] : 해당 조건을 가진 경로에 있는 파일 검색하기

ex) find . -name system.log : 현재위치에 system.log라는 파일이름 가진 파일 찾기

which : 명령어의 위치를 찾는 명령어

top : CPU, Memory 상태 확인하는 명령어


(각 top 분석의 내용들 다 알기)

top은 3초정도의 갱신을 하는데,

1초마다 갱신하기 : **d** 키 누르고 엔터 또는 **top -d 1**

```
top - 11:01:02 up 13 days, 17:56, 4 users, load average: 0.13, 0.22, 0.22
Tasks: 349 total, 1 running, 345 sleeping, 3 stopped, 0 zombie
%Cpu(s): 2.1 us, 2.5 sy, 0.0 ni, 95.2 id, 0.1 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 32502980 total, 4303140 free, 11432752 used, 16767088 buff/cache
KiB Swap: 16383996 total, 16383996 free, 0 used. 19145852 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
22954	v75	20	0	6785344	622756	26596	S	3.3	1.9	160:54.19	java
22628	v75	20	0	6780092	632048	26552	S	3.0	1.9	170:30.64	java
22765	v75	20	0	5827060	305224	26272	S	2.6	0.9	117:15.94	java
23115	v75	20	0	6712544	889772	26668	S	2.6	2.7	100:12.35	java
16878	jja	20	0	6775896	622956	26536	S	2.3	1.9	146:43.31	java
799	polkitd	20	0	643412	22040	5264	S	2.0	0.1	2:23.00	polkitd
19438	edu	20	0	6746380	1.415g	26672	S	2.0	4.6	212:03.39	java
20507	root	20	0	157976	2540	1560	R	1.7	0.0	0:00.15	top
795	dbus	20	0	36500	3432	1440	S	1.0	0.0	1:35.39	dbus-daemon
15329	obzents	20	0	1937548	118356	39996	S	1.0	0.4	5:05.85	gnome-shell
19300	edu	20	0	6788592	662372	26576	S	1.0	2.0	196:48.75	java
27029	v70	20	0	6119732	480972	13924	S	0.7	1.5	69:48.94	java
1	root	20	0	193764	6988	3968	S	0.3	0.0	2:19.64	systemd
9	root	20	0	0	0	0	S	0.3	0.0	20:51.50	rcu_sched
43	root	20	0	0	0	0	S	0.3	0.0	0:15.37	ksoftirqd/7
838	root	20	0	24264	1784	1428	S	0.3	0.0	0:34.65	systemd-logind
888	root	20	0	340580	31676	6808	S	0.3	0.1	0:29.34	firewalld
900	root	20	0	119668	1248	700	S	0.3	0.0	0:52.71	kstuned
2559	root	20	0	220068	25948	10772	S	0.3	0.1	6:28.79	Xorg
3002	root	20	0	410396	6572	4652	S	0.3	0.0	0:26.37	packagekitd
3023	root	20	0	371056	5668	3716	S	0.3	0.0	8:25.66	udisksd
9433	root	20	0	0	0	0	S	0.3	0.0	0:00.78	kworker/3:2
10729	root	20	0	0	0	0	S	0.3	0.0	0:02.25	kworker/5:0
10909	root	20	0	0	0	0	S	0.3	0.0	0:04.34	kworker/4:1
15015	root	20	0	524800	7180	4264	S	0.3	0.0	0:14.22	gdm-session-wor
15245	obzents	20	0	946356	23900	15780	S	0.3	0.1	0:31.76	gnome-settings-
17019	jja	20	0	6694300	632004	26452	S	0.3	1.9	97:55.31	java
17616	oracle	20	0	2327844	18360	16368	S	0.3	0.1	0:01.49	oracle
22054	root	20	0	0	0	0	S	0.3	0.0	0:04.78	kworker
24467	v50	20	0	6614628	374988	13680	S	0.3	1.2	96:57.11	java
24522	v50	20	0	5790608	354784	13852	S	0.3	1.1	88:18.92	java



< top 세부 정보 필드별 항목 >

- 13 days, 17:56 : 13일 17시간 56분 전에 서버가 구동

- **load average** : 현재 시스템이 얼마나 일을 하는지를 나타냄. 3개의 숫자는 1분, 5분, 15분 간의 평균 실행/대기 중인 프로세스의 수. CPU 코어수 보다 적으면 문제 없음
- **Tasks**
 - **total** : 총 349개의 프로세스가 있음
 - **running** : 1개의 프로세스가 실행중
 - **sleeping** : 345개의 프로세스가 대기중
 - **stopped** : 3개의 프로세스가 멈춤
 - **zombie** : 0개의 프로세스가 좀비상태
 - *좀비상태 : 부모 프로세스의 오류로 인하여 종료한 자식 프로세스의 종료가 적절히 이루어지지 않음, 커널 오류에 의한 비정상적인 상태
- **%Cpu(s)**
 - **%us** : 유저 레벨에서 사용하고 있는 CPU의 비중
 - **%sy** : 시스템 레벨에서 사용하고 있는 CPU비중
 - **%id** : 유휴 상태의 CPU 비중
 - **%wa** : 시스템이 I/O 요청을 처리하지 못한 상태에서의 CPU idle 상태인 비중
- **KiB Mem, Swap** : 각 메모리의 사용량
- **Mem** :
 - **total** : 전체 물리적인 메모리
 - **used** : 사용중인 메모리
 - **free** : 사용되지 않는 여유 메모리
 - **buffers** : 버퍼된 메모리
- **Swap** : RAM에 용량이 부족할 경우 프로세스가 임시 저장되는 공간
 - **total** : 전체 스왑 메모리
 - **used** : 사용중인 스왑 메모리
 - **free** : 남아있는 스왑 메모리
 - **cached** : 캐싱 메모리
- **PID** : 프로세스 ID (**Process ID**의 약자)
- **USER** : 프로세스를 실행시킨 사용자 ID
- **PR** : 프로세스의 우선순위 (**priority**의 약자)
- **NI** : 프로세스의 nice값을 보여주는 항목이다. (**nice value**의 약자)

(* nice : 유저 공간에서 설정한 프로세스 실행 우선 순위를 뜻한다.)

(* -20 ~ +19 까지의 값을 가지고 있으며, -20값이 우선순위가 가장 높은 값이고 양수로 올라갈수록 낮은 값이다.)

- **VIRT** : 가상 메모리의 사용량 (**virtual memory**의 약자)
 - **RES** : 현재 페이지가 상주하고 있는 크기 (**resident size**의 약자)
 - **SHR** : 다른 프로세스와 공유하고 있는 양(**shared memory**의 약자)
- (* VIRT, RES, SHR : 메모리 사용량으로 누수를 확인할 수 있다.)
- **S** : 프로세스의 상태 [S(sleeping), R(running), W(swapped out process), Z(zombies)]
 - **%CPU** : 프로세스가 사용하는 CPU의 사용률

- %MEM : 프로세스가 사용하는 메모리의 사용률
- TIME+ : 프로세스 시작된 이후 경과된 총 시간
- COMMAND : 실행된 명령어

w, who : 접속한 계정 정보에 대한 내용 출력

ping : 네트워크 설정, 인터넷 연결되어 있는지 확인

ping [도메인 또는 ip주소] :

- latency : 내 PC와 도메인간의 네트워크 왕복시간

nslookup [도메인주소] : ip주소 출력

ps : 프로세스 상태 출력 (process status)

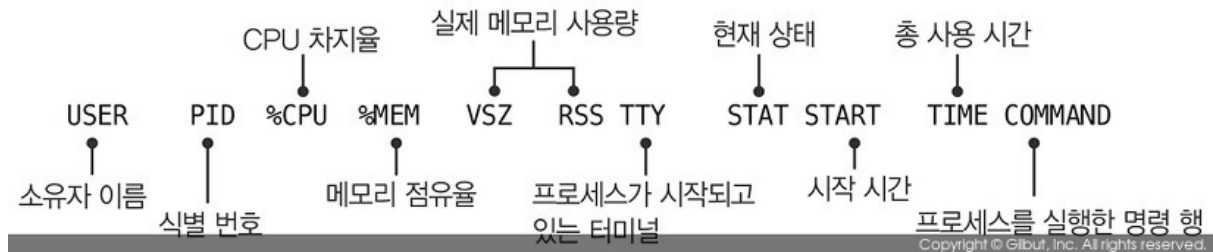
processor : CPU를 의미

process : 실행파일이 메모리에 업로드된 상태를 의미함

ps -ef : 현재 내 pc의 모든 프로세스 상태 출력

ps -ef | grep '문자열' :

주목해야 할 것 **PID, PPID**에 주목해야 함



CPU: cpu 사용 비율(추정치)

MEM: 메모리 사용 비율(추정치)

VSZ: k단위 또는 페이지 단위의 가상 메모리 사용량

RSS: 실제메모리 사용량

TT: 프로세스와 연결된 터미널

STAT: 현재 프로세스의 상태코드

kill -9 : 프로세스 멈추기

kill -9 [process.id] :

adduser : 새로운 계정 만들기(home 계정에 만들어짐)

deluser : 계정 삭제하기

su - 사용자계정 : 원하는 사용자계정으로 계정바꾸기

df : 현재 마운트된 하드디스크의 용량을 보여준다.

df -h : 사람이 읽기 편하게 하드 디스크의 용량을 보여준다.

du : 파일이나 디렉토리들이 디스크에서 차지하는 크기 출력

du -h : 현재 디렉토리에 있는 모든 파일 및 디렉토리에 대한 크기를 출력

du -sh : 현재 디렉토리의 총 사용량을 출력

wget [다운받을 주소] : 웹에서 다운로드 받아오자

fdisk : 디스크 파티션을 확인하고 추가/삭제 하는 명령

** 방화벽 설정

RHEL 7 부터는 방화벽을 관리하는 데몬이 **firewalld**로 변경되었고, 방화벽 설정은 복잡한 **iptables** 명령어 대신 **firewall-cmd** (콘솔), **firewall-config**(X-Windows) 명령어를 사용하는 것을 권장합니다.

방화벽 관련 정보

: <https://www.lessstif.com/system-admin/rhel-centos-firewall-22053128.html>

설치 및 서비스 구동

yum install firewalld : 방화벽 설치

systemctl start firewalld : 방화벽 시작

systemctl enable firewalld : 서버 부팅시 방화벽 서비스를 자동으로 시작

systemctl stop firewalld : 서비스 중지

systemctl disable firewalld : 재부팅시 자동 중지

systemctl status firewalld : 서비스 상태 확인

설정

/usr/lib/firewalld/ 에 위치하고 있고 미리 설정된 **zone**과 **zone**별 허용하는 서비스들은 이 폴더를 보면 확인할 수 있다.

sudo firewall-cmd --state : 방화벽 상태 체크

sudo firewall-cmd --list-all : 사용 가능한 모든 서비스/포트 목록 조회

sudo firewall-cmd --permanent --zone=public --add-port=80/tcp : 포트 추가

sudo firewall-cmd --permanent --zone=public --remove-port=80/tcp : 포트 삭제

sudo firewall-cmd --permanent --zone=public --add-service=http : 서비스 추가

sudo firewall-cmd --permanent --zone=public --remove-service=http : 서비스 삭제

sudo firewall-cmd --reload : 방화벽 재시작

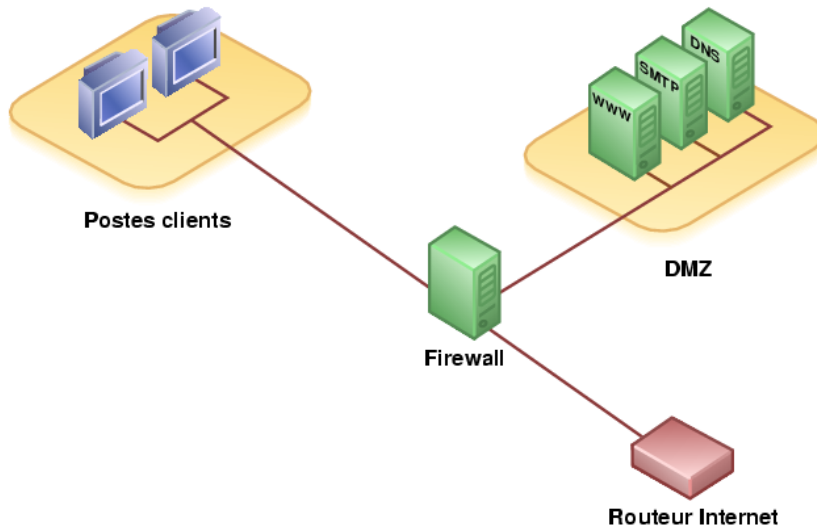
80포트 열기

firewall-cmd --zone=public --permanent --add-port=80/tcp (80 포트 입력)

firewall-cmd --zone=public --permanent --add-port=http

firewall-cmd --zone=public --permanent --add-port=https

**** zone=dmz?** 내부 네트워크(일반적으로 인터넷)와 외부 네트워크 사이에 위치한 서버이다. 내부 네트워크와 외부 네트워크가 **dmz**로 연결할 수 있도록 허용하면서도, **dmz** 내의 컴퓨터 안에 있는 호스트들은 내부 네트워크로 연결할 수 없다.



시스템 개별 설정

/etc/firewalld/ 에 위치하고, default zone 등 firewall의 동작은 /etc/firewalld/firewalld.conf 에서 지정할 수 있습니다.

DefaultZone = dmz

커스터마이징된 zone 설정은 /etc/firewalld/zones/[ZONE].xml 에서 확인할 수 있습니다.

예로 public존은 etc/firewalld/zones/public.xml에 설정합니다.

firewall-cmd --reload : 서비스 재구동

***zone**

: 존은 서버의 용도에 맞게 사전에 정의된 네트워크의 신뢰 레벨을 의미

firewall-cmd --get-zones : 사전 정의된 존 목록 출력

firewall-cmd --list-all-zones : 전체 존 목록을 상세하게 출력

firewall-cmd --get-default-zone : default 로 설정된 존 출력

firewall-cmd --get-active-zone : 활성화된 존 출력

firewall-cmd --permanent --new-zone=webserver(존 name) : 새로운 존 추가

firewall-cmd --permanent --delete-zone=webserver(존 name) : 존 삭제

서비스

firewall-cmd --get-services : 사전에 정의된 서비스 목록

firewall-cmd --permanent --list-all --zone=dmz : dmz존에 영구적으로 등록된 서비스 목록 확인

포트 (포트 추가/변경, ip 추가/변경 은 --reload 옵션을 실행해야 반영됨)

firewall-cmd --permanent --zone=public --add-port=8080/tcp : 포트 추가

firewall-cmd --permanent --zone=public --add-port=4000-4100/tcp : 4000~4100 포트추가

firewall-cmd --permanent --zone=public --remove-port=8080/tcp : 포트 삭제

firewall-cmd --permanent --zone=public --add-source=192.168.1.0/24 --add-port=22/tcp
: 192.168.1 대역에서 ssh 접근을 허용(--add-source=<ip 범위>/netmask

****netstat 명령어 : 네트워크 상태 확인**

netstat [옵션] | grep 포트번호 or 서비스명

netstat -an | grep 8080 : 8080port 현재상태 확인(열었는지 아닌지)

netstat -nap : 연결을 기다리는 목록과 프로그램을 보여줌

netstat -nlpt : TCP listening 상태의 포트와 프로그램을 보여줌

**** systemd(system daemon)을 관리하는 systemctl 명령어 사용법**

: systemd(system daemon)은 전통적으로 Unix 시스템이 부팅 후에 가장 먼저 생성된 후에 다른 프로세스를 실행하는 init 역할을 대체하는 데몬입니다. 지금은 RHEL / CentOS / Ubuntu / Arch 등 대부분의 리눅스 시스템에 공식적으로 채택되었습니다.

서비스 관리

서비스 상태 확인

systemctl status 서비스명 : 확인할 서비스

systemctl status nginx : nginx 서비스 상태 출력

systemctl start mariadb : mariadb 시작

서비스 자동 시작

systemctl enable mariadb : 부팅 시 mariadb 자동 시작

서비스 목록보기

sudo systemctl list-units : 설치된 모든 unit 파일을 보려면 list-unit-files

서비스 마스킹 : 동일한 용도로 사용하는 서비스가 동시에 설치되어 있을 경우 충돌할 수 있어, 서로 충돌하는 서비스로 시작하지 않게 합니다.

sudo systemctl mask ntpd : ntpd 구동되지 않음

=> sudo systemctl start ntpd : 시작해도 Unit is masked라고 구문이 뜨며 실행되지 않음

sudo systemctl unmask ntpd : ntpd 마스킹 해제

조건에 따라 서비스 보기

sudo systemctl list-units --state=enabled : enabled된 모든 서비스

sudo systemctl list-units --state=failed : 구동에 실패한 서비스

sudo systemctl list-units --state=active : 모든 active인 목록

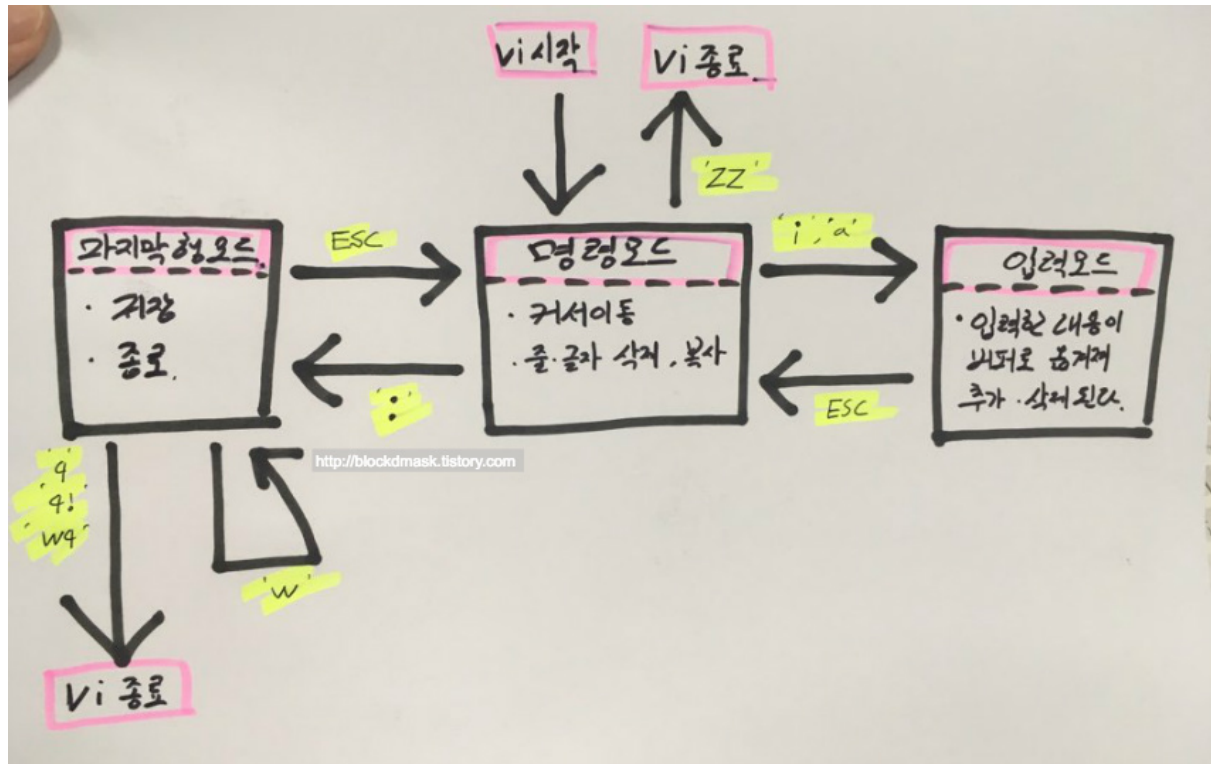
sudo systemctl list-units --state=inactive : 상태가 inactive인 목록

sudo systemctl list-units --type=service --state=running : 서비스중에 상태가 running인 목록

sudo systemctl is-active nginx : 특정 서비스가 active 상태인지 조회

sudo systemctl is-enabled nginx : 서비스가 부팅때 구동되도록 설정 여부

Linux / Unix Vi 편집기 명령어



i) 명령 모드(command mode) - 처음 vi 명령어로 vi를 시작하게 되면 들어가게 됩니다. 여기서는 방향키를 이용하여 커서를 이동할 수 있으며, "dd" 나 "yy" 로 한 줄 삭제 및 한 줄 붙여넣기, 또는 "x" 명령어로 글자 하나를 삭제할 수 있습니다. (더 많은 명령어는 아래에)

- i : insert 의 명령어로 입력모드로 전환된다.
- a : 현재 커서 바로 다음위치에서 입력 모드 전환.
- o : 현재 줄 다음 위치에서 입력모드로 전환.
- s : 현재 커서 위치에서 한 글자를 지우고 입력모드로 전환
- u : 방금 입력한 명령을 취소할 수 있다.(ctrl+z와 동일)
- x : 현재 커서가 위치한 문자를 삭제한다.
- p : 복사한 내용을 현재 행 이후에 붙여 넣는다.
- ctrl + f : 다음 화면으로 이동 (forward 의 약자)
- ctrl + b : 이전 화면으로 이동 (back 의 약자)
- ctrl + y : 한 줄씩 내려간다.
- ctrl + e : 한 줄씩 올라간다.
- ctrl + d : 반 정도 화면이 이동한다.

ii) 입력 모드(insert mode) - 명령모드에서 "i" 나 "a" 명령을 통해서 입력 모드로 넘어 갈 수 있습니다. 입력모드로 가게되면, 자유롭게 코드나 글을 작성을 하시면 됩니다. 명령 모드로 다시 돌아오려면 "ESC"를 누르면 됩니다.

iii) 마지막 행 모드(**Last line mode**) - 마지막행 모드는 명령모드에서 ":" (콜론) 을 입력하면 화면 맨 밑단에 :_____ 하며 입력을 할수 있는 공간이 나옵니다. 여기서 현재까지 내가 작성한 이 내용을 저장하고 vi를 종료(wq)할지, 그냥 종료(q, q!)할지 등을 입력할 수 있습니다

- :q : vi에서 작업한 것이 없을 때 vi 종료된다.
- :q! : 작업한 내용을 저장하지 않고 종료된다.
- :w [파일명] : 작업한 내용을 저장만 한다. 파일명을 지정하면 새 파일로 저장
- :wq, wq! : 작업한 내용을 저장하고 vi 종료
- ZZ(대문자) : 작업한 내용을 저장하고 vi 종료