

리눅스\_1

2025341034\_kopo36\_B반\_신석환

## 1. 강의 들어가기

1) 개발자들은 무엇을 가장 많이 할까?

⇒ 기업에서 처리되고있는 많은 정보시스템을 개발, 유지보수 등등

⇒ 유닉스, 리눅스 개발자로서 꼭 알아야 하는 것 : 유닉스, 리눅스, 서버, 우분투(ubuntu)

## 2) 용어 찾아보기

=> 리눅스와 유닉스의 차이

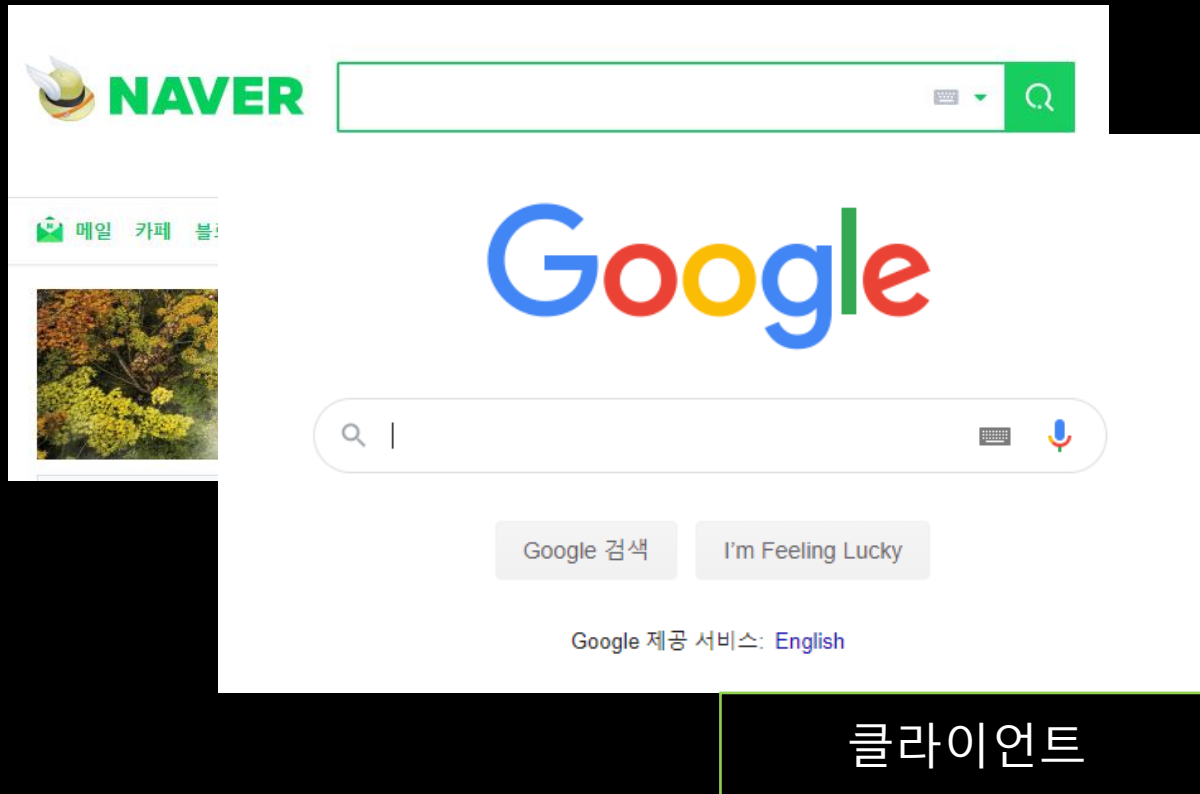
리눅스와 유닉스의 차이점

	리눅스	유닉스
신뢰도	서버들이 수 년간 중단 없이 운행되곤 하지만, 일반적인 사용자에게 있어 리눅스는 매우 믿음직한 운영 체제이다.	포털이나 대기업 사이트 같은 주요한 서버에 사용된다. 그만큼 유닉스는 신뢰도 문제에서 큰 점수를 받는 운영 체제이다.
성능	과중한 네트워크 부하를 처리하여야 할 경우에는 최적화된 성능을 발휘하지는 못하지만, 서버 운영체제로서 적합하다.	서버 운영용 운영체제이다. ID와 비밀번호 관리 등의 기능과 암호화 능력 월등히 좋다. 안정성과 메모리 관리 능력도 뛰어나다.
기술 지원	많은 기관들이 리눅스를 전문적으로 지원한다. 모든 주요 리눅스 업체들은 다양한 방법으로 고객을 지원하며, 몇몇 업체는 중단없이 365일 서비스를 제공한다. 최후의 수단으로 소스를 수정하여 직접 문제를 해결할 수 있다.	하드웨어와 OS 자체가 통합적으로 개발되서 나오므로 IBM사에서 일괄 관리한다. 회사 브랜드가 걸려있어 리눅스에 비해서는 지원이 좋은 편이다.
구축 비용	몇몇 업체들은 상용 배포판을 매우 저렴한 가격에 제공한다. 어플리케이션과 문서들 또한 무상 혹은 저렴하게 구할 수 있다. 리눅스는 사용 제한이 없기 때문에, 구축하고자 하는 모든 시스템에 추가 비용 없이 설치할 수 있다. 리눅스 시스템을 위한 총 구축 비용은 매우 저렴하다고 할 수 있다	굉장히 비싸다. 유닉스는 장비 자체도 따로 나오는 경우가 태반이고, PC용 유닉스도 있지만 보통은 장비가 하나 개발되고 거기에 최적화된 유닉스를 개발해서 상품으로 나오는 경우가 많다. 따라서 장비 값만 해도 몇 억씩 하는 경우가 대부분. 그러나 대기업에서는 큰 비용을 들여서라도 보안 등의 문제로 유닉스를 사용한다.
하드웨어 지원	리눅스는 다양한 하드웨어, 플랫폼들을 지원한다.	많은 상용 유닉스가 있지만, 리눅스에는 못 미친다.

# 1. 클라이언트 서버 환경

## 1) 개요

- 일상적 정보처리 시스템은 클라이언트/서버 환경으로 구성되어있음
- 클라이언트는 사용자에게 보여지는 부분
- 서버는 클라이언트가 수행할 수 있는 서비스를 지원하는 부분



## 1. 클라이언트 서버 환경

### 2) 서버의 운영세계

- 클라이언트 : 화면에 보이는 것, 기기의 다양한 기능 사용, 사용자 중심
- 서버 : 데이터처리, 통신, 업무 중심, 대용량, 가용성, 확장성, 안정성

### 3) 유닉스와 리눅스 비교

- 유닉스 : 안정성, 대용량, 기존의 강자, 코어 업무(금융, 국가, 국방)
- 리눅스 : 오픈소스, 최근에 서버의 OS로 신흥강자, 기존 업무 많이 변환(U2L)

### 4) 데스크탑 형식과 서버형식

- 리눅스 : 서버의 OS역할 뿐만 아니라, 현재 윈도우(MacOS)의 PC OS의 역할도 담당할뿐함
- text기반 OS에서 Windows같은 UI기반의 데스크탑 형식에 대한 노력
- 안드로이드 - 리눅스 데스크탑이 진화한 모바일 버전
- Ubuntu : 시장에 강자. 구글링이 쉬움.
- CentOS : 유료 기업용 리눅스인 red Hat과 호환, 오라클 사용가능, 큰 기업용
- ubuntu는 데스크탑, 모바일, 서버, 클라우드 등의 다양한 버전을 제공
- 서버형식은 화면을 보여주는 윈도우UI가 필요하지 않음
- 경량화, 고성능화를 위해 낭비X

## 1. 클라이언트 서버 환경

### 5) 실제 서버, 가상 서버

- 실제서버는 대용량, 고가용성, 다중사용자. 혼자 개발 중 안된다고 켜다 켜다 불가능
- 가상서버 (Virtual Machine/ Cloud기술) 기술이 나오기 전에는 작은 PC/서버에 리눅스를 설치하여 개발용으로 사용함
- 현재 : PC의 성능이나 용량이 커서 자원을 나누어 사용 (개인 가상머신, virtual cloud)  
이 가상서버의 용량이 개발자 1인용도. 범용 서비스를 하기 위해서는 용량확장을 고려
- 퍼블릭 클라우드(AWS, GCP, Azure)에서 작은 용량의 개발 서버를 인터넷공간에 두어 개발 서비스 개시 때 대용량으로 증설된 클라우드 서버 활용



실제 서버는 온전한 장비상태를  
유지하기위한 시설에 있음



# 1. 클라이언트 서버 환경.

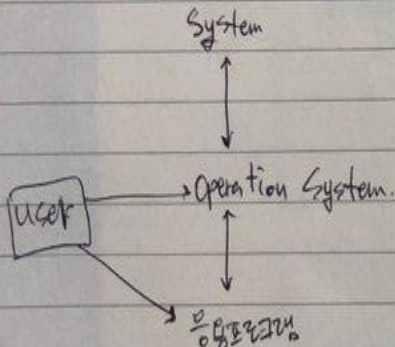
## 1) 개요

(vs Standalone)

- 일방적 정보처리 시스템 클라이언트/서버 환경으로 구성
- 클라이언트는 사용자에게 보여주는 부분.
- 서버는 클라이언트가 수행할 수 있는 서비스를 지원하는 부분.
- 웹브라우저, 서버 응용, 데이터가 저장되는 장소.

## 2) 서버의 운영체제

- 클라이언트는 화면에 보이는 것, 거의 다양한 기종 사용, 서버 환경
- 서버는 데이터처리, 통신, 업무처리, 대용량, 가용성, 확장성, 안정성
- 서버의 운영체제는? Windows 보다 Linux/Unix가 유리



## 3) 유닉스와 리눅스 비교.

- 유닉스 : 안정성, 대용량, 기존의 강자, 코어업의 (금융-계정계) 극대화
- 리눅스 : 오픈소, 최근에 서버의 OS로 인줄가자, SMP는 많은 기종에서도 많이 변화 (ULA)

## 4) 데스크톱형과 서버형

- 리눅스 : 서버의 OS 역할, 윈도우의 PC OS도 담당함.
- 즉 저가 기반 OS에서 Windows 같은 UI기반의 데스크톱 형식에 대한 노력
- 안드로이드의 탄생 -> 리눅스 데스크톱이 전환한 모바일 버전.
- 많은 리눅스 애호가 존재 (명령어들은 비슷)
- 시장점유율 Ubuntu (70~80%), 또다른 강자 CentOS
- Ubuntu : 시장에 강자, 안드로이드, 구글링 쉬운, 오픈소스
- CentOS : 동조하는 기존 기업용 리눅스인 Red Hat과 호환, 호환사용가능. 큰 기업 고려
- Ubuntu는 데스크톱, 모바일, 서버, 클라우드 등의 다양한 배포판 제공
- 서버형은 하먼을 보여주는 윈도우가 격차 X
- 확장성, 고성능화를 위하여 굳이 남발할 필요 X

## 5) 복제, 가상 서버

- 현재 서버는 대용량, 고가용성, 다중사용자, 호환 개변하다 안되면 깃다 컴다름가.
- 가상서버 기술이 나오기전에는 작은 PC/서버에 리눅스를 설치하여 개발용으로 사용.
- 현재 : PC 성능, 용량 커서 자원을 나누어 사용, 하지만 가상서버의 용량이 보통 개발자 1인용도.
- 범용서버를 위해 용량확장 도입해야함.
- 또는 퍼블릭 클라우드 (AWS, GCP, Azure)에서 작은 용량의 개발서버를 번외로 공간에 들어가면.

## 용어. 리눅스 - 1강

### 1. 클라이언트와 서버 (Server)

클라이언트

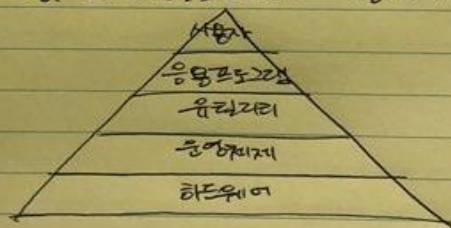
- 네트워크로 연결된 서버로부터 정보를 제공받는 컴퓨터
- 클라이언트는 요청, 서버는 응답의 관계.
- 인터넷과 연결된 거의 모든 소프트웨어들.
- 클라이언트와 반대인 개념.
- ex) 웹 -> 웹브라우저에서 URL을 입력하면 그 URL에 해당하는 웹서버로 웹페이지에 대한 요청이 전달된다. 웹서버는 웹브라우저가 요청한 정보를 제공.

서버

### 2. 운영체제 (OS)

운영체제(OS)

- 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고, 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임.
- 컴퓨터 사용자와 컴퓨터 하드웨어간의 인터페이스로 동작하는 시스템 소프트웨어의 일종. 다른 응용프로그램이 유용한 작업으로 할 수 있도록 환경을 제공.



GUI

### 3. GUI (Graphical User Interface)

- 마우스로 아이콘을 클릭하여 프로그램을 작동시키는 컴퓨터 환경.
- 반대편 지면 -> CLI (Command-Line Interface)

TUI

### 4. TUI (Text User Interface)

- 문자열기반 사용자 인터페이스와 구분하기 위해 그래픽 사용자 인터페이스의 반대편에 만들어진 일명 텍스트 기반. 따라서 사용자 인터페이스라고도 한다.

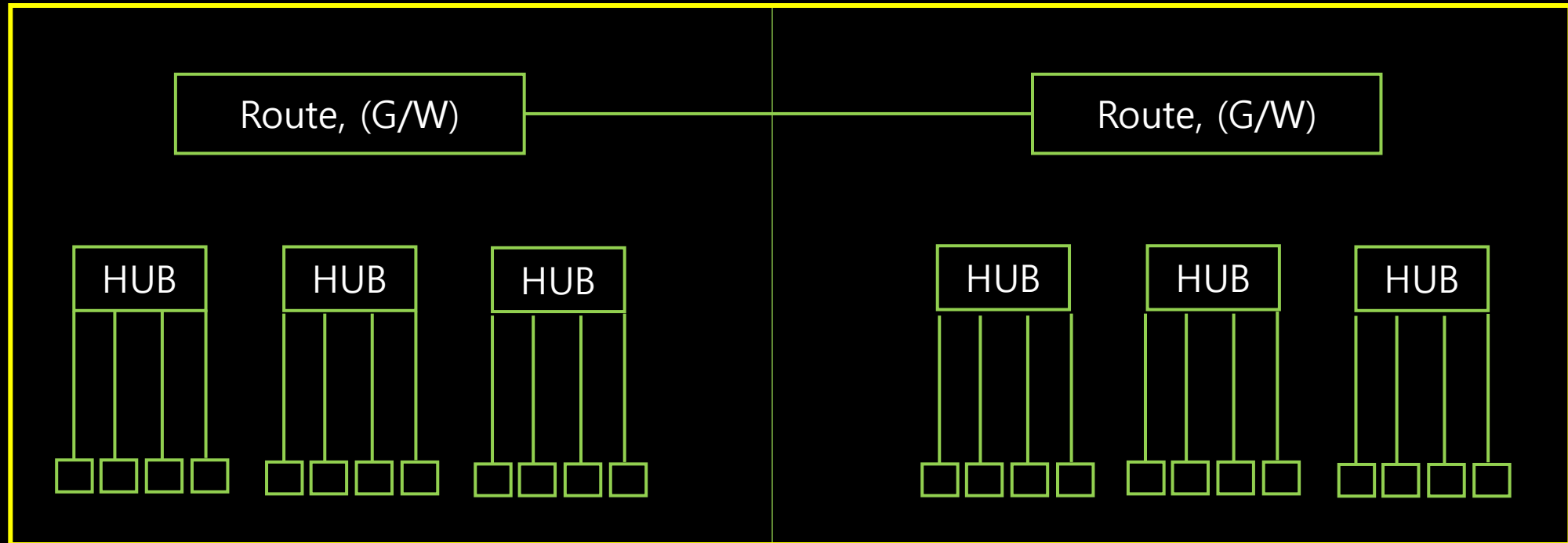
# 리눅스\_2

2025341034\_kopo36\_B반 신석환

## 2. 개발환경 구축

### 1) 서버 사이드 설치

- 네트워크 위에서 주소 : IP(Internet Protocol)
- 같은 네트워크 (허브, 브릿지, L2), 다른 네트워크 (라우터, 게이트웨이, L3), 인터넷 공유기(L2, L3)
- 유선, 무선, Internet working, TCP/IP
- DNS





## 2. 개발환경 구축

### 1) 서버 사이드 설치

- DNS : Domain Name Service, DNS를 서비스하는 서버, 국내 DNS, 해외 DNS

```
C:\Users\401-ST05>ping naver.com
```

```
Ping naver.com [125.209.222.141] 32바이트 데이터 사용:  
요청 시간이 만료되었습니다.  
요청 시간이 만료되었습니다.  
요청 시간이 만료되었습니다.  
요청 시간이 만료되었습니다.
```

```
125.209.222.141에 대한 Ping 통계:  
패킷: 보냄 = 4, 받음 = 0, 손실 = 4 (100% 손실),
```

```
C:\Users\401-ST05>ping google.com
```

```
Ping google.com [172.217.161.174] 32바이트 데이터 사용:  
172.217.161.174의 응답: 바이트=32 시간=41ms TTL=43  
172.217.161.174의 응답: 바이트=32 시간=41ms TTL=43  
172.217.161.174의 응답: 바이트=32 시간=41ms TTL=43  
172.217.161.174의 응답: 바이트=32 시간=40ms TTL=43
```

```
172.217.161.174에 대한 Ping 통계:  
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),  
왕복 시간(밀리초):  
최소 = 40ms, 최대 = 41ms, 평균 = 40ms
```

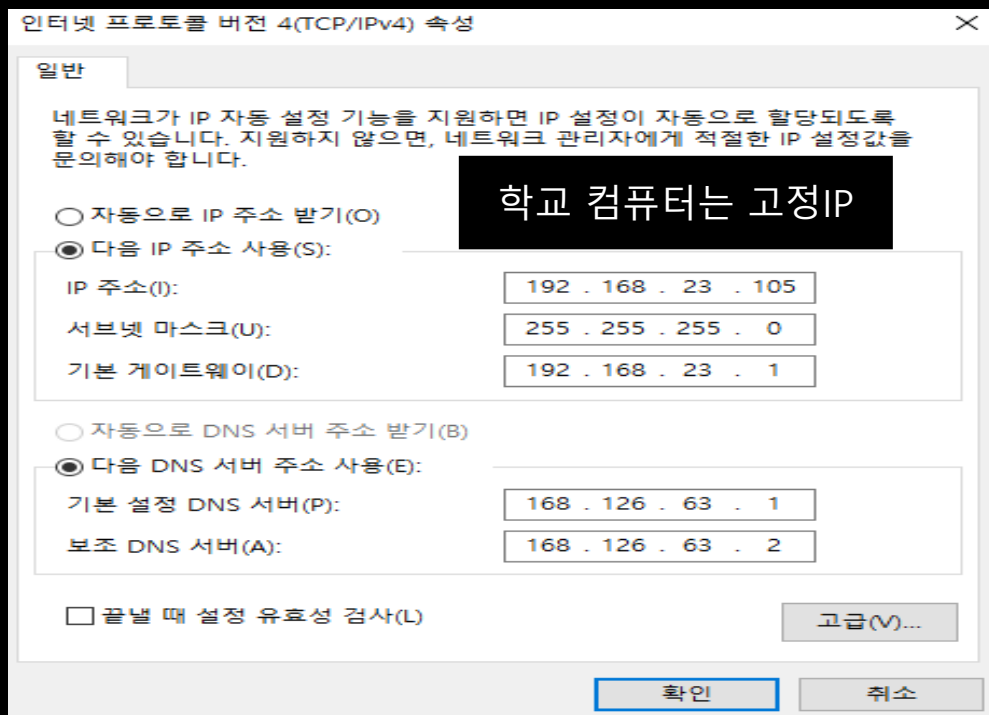
```
C:\Users\401-ST05>_
```

Ex)네이버라고 치면 네이버에 대한 IP로 변환해 주는 것

## 2. 개발환경 구축

### 1) 서버 사이드 설치

- 고정IP(Static IP) : IP를 지정, PC에서 지정, 라우터(공유기)에서 MAC어드레스로 고정
- 유동IP(DHCP – Dynamic Host Configuration Protocol) – 처음 넷에 접속시 아이피를 순차적으로 배정받음. 누가줄까? 라우터...공유기 등의 DHCP기능으로 준다
- NIC(Network Interface Card) 별로 IP부여, 즉 노트북 기준 유선랜(card), 무선랜(card)별 IP가 다르게 준다
- IP확인 -> PC : ipconfig / 리눅스 계열 : ifconfig -a
- 안드로이드/iphone/맥북 : 터미널 App을 설치한 후 거기서 ifconfig -a (rooting필요)



## 2. 개발환경 구축

### 1) 서버 사이드 설치

- 공인IP(public IP) : 인터넷상에서 공식적으로 인식 가능한 IP  
(서비스를 요청하고, 제공하는 것이 가능) – 하지만 공인IP는 개수가 제한적.  
그렇기에 모든 PC등 단말에게 부여할 수 없다.
- 사설IP(private IP) : 내부에서 가상으로 사용하는 IP, 인터넷 서비스를 요청하여 사용할 수는 있지만,  
서버 등으로 인터넷 서비스제공 불가
- 사설IP는 인터넷 구간으로 나가면서 공인IP를 여러단말과 나눠쓰며,  
인터넷 공인IP로 변환(NAT) 하여 사용됨

인터넷에서 통하는 IP

변환되서(NAT) 컴퓨터 내부에서 갖는 IP

이 컴퓨터의 IP주소는 **112.76.112.150** 입니다.

명령 프롬프트

C:\Users\401-ST05>ipconfig

Windows IP 구성

이더넷 어댑터 로컬 영역 연결:

연결별 DNS 접미사 . . . . .	:	
링크-로컬 IPv6 주소 . . . . .	:	fe80::20e7:a7dc:3ae7:f01a%13
IPv4 주소 . . . . .	:	<b>192.168.23.105</b>
서브넷 마스크 . . . . .	:	255.255.255.0
기본 게이트웨이 . . . . .	:	192.168.23.1

이더넷 어댑터 VirtualBox Host-Only Network:

연결별 DNS 접미사 . . . . .	:	
링크-로컬 IPv6 주소 . . . . .	:	fe80::21df:4ae1:7428:194%24
IPv4 주소 . . . . .	:	192.168.56.1
서브넷 마스크 . . . . .	:	255.255.255.0
기본 게이트웨이 . . . . .	:	

#내 IP는 사설IP로 192.168.23.105인데 네이버 서비스를 받을 수 있고,  
네이버서버는 나를 112.76.112.150으로 인지.

#이 NAT변환은 L3 장비에서 이루어지고 해당 공인IP도 유동IP/고정IP일 수 있다.

#고정IP면 당연히 비싸다. 유동IP를 위한 DDNS서비스가 있다.

## 2. 개발환경 구축

### 1) 서버 사이드 설치

- 특별한 IP : 127.0.0.1(일명 loopback) 자기 자신의 IP, localhost라고도 한다.  
관습적 (끝자리 1 : 라우터, 끝자리 255 : 같은 네트워크상 모든 단말. Ex)192.168.23.1  
192.168.23.255
- Port 번호 : IP는 해당 단말(PC, 서버 등)까지의 주소, IP아래 서비스를 위한 포트번호 체계  
EX) 택배시키면, 주소쓰고 받는 사람까지 써야한다. 포트는 받는사람의 개념.
- 몇몇 기본적으로 예약되어 있는 포트번호 : http : 80, https : 8080, mysql : 3306, ssh : 22, netstat -a
- <http://www.naver.com> => [www.naver.com](http://www.naver.com)라는 이름으로 DNS를 찾아 IP를 받아온 후  
http서비스(포트번호 80)를 요청하라.

```
C:\> 명령 프롬프트
C:\Users\401-ST05> ping 127.0.0.1

Ping 127.0.0.1 32바이트 데이터 사용:
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128

127.0.0.1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 0ms, 최대 = 0ms, 평균 = 0ms
```

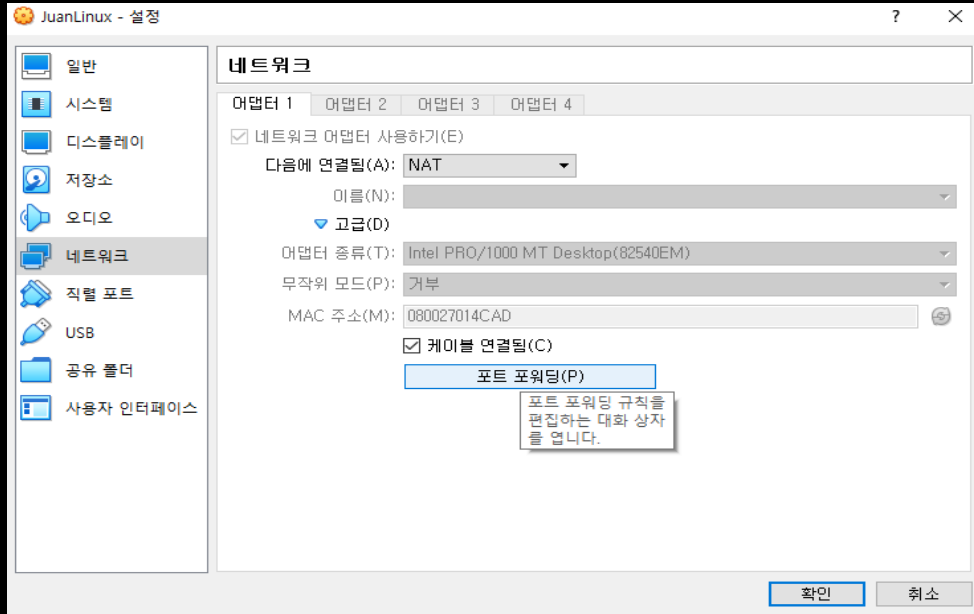
```
C:\Users\401-ST05> ping localhost

Ping 401-ST05-HP [::1] 32바이트 데이터 사용:
::1의 응답: 시간<1ms
::1의 응답: 시간<1ms
::1의 응답: 시간<1ms
::1의 응답: 시간<1ms

::1에 대한 Ping 통계:
패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
최소 = 0ms, 최대 = 0ms, 평균 = 0ms
```

# 실습

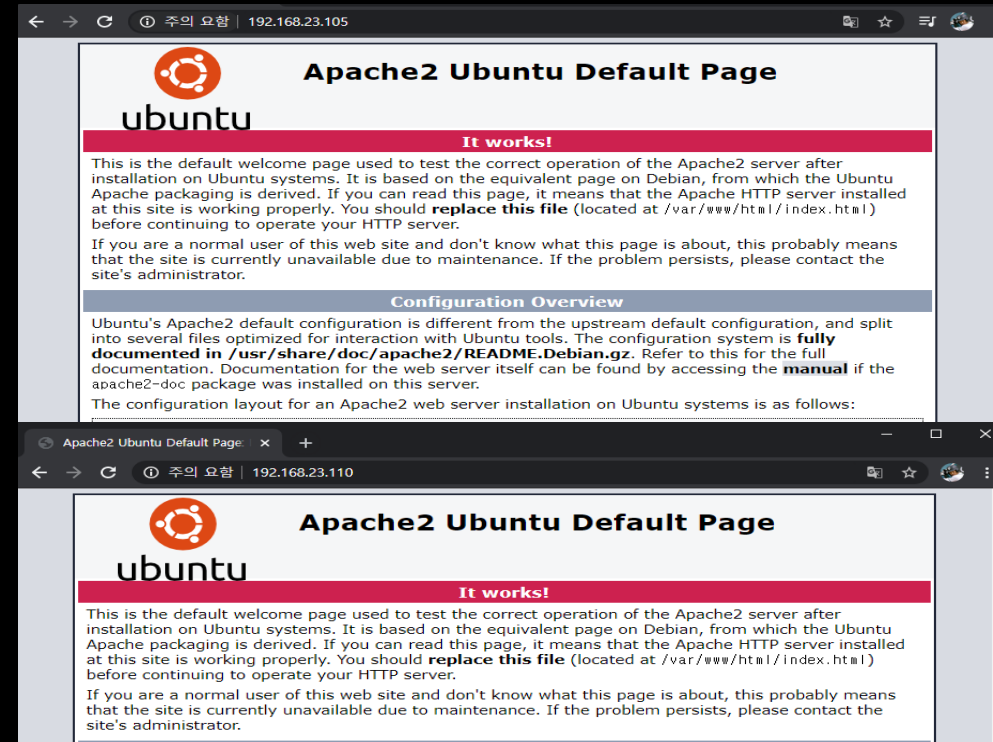
1 : virtualBox 포트포워딩 설정 : 22(ssh), 80(http), 8080(https), 3306(mysql)



이름	프로토콜	호스트 IP	호스트 포트	게스트 IP	게스트 포트
Rule 1	TCP		80		80
Rule 2	TCP		22		22
Rule 3	TCP		8080		8080
Rule 4	TCP		3306		3306

2 : 리눅스에서 아파치2 깔기

리눅스실행 -> root로 접속 -> apt install apache2 하면 깔리고  
웹브라우저에서 내 IP를 입력해본다.





# 실습

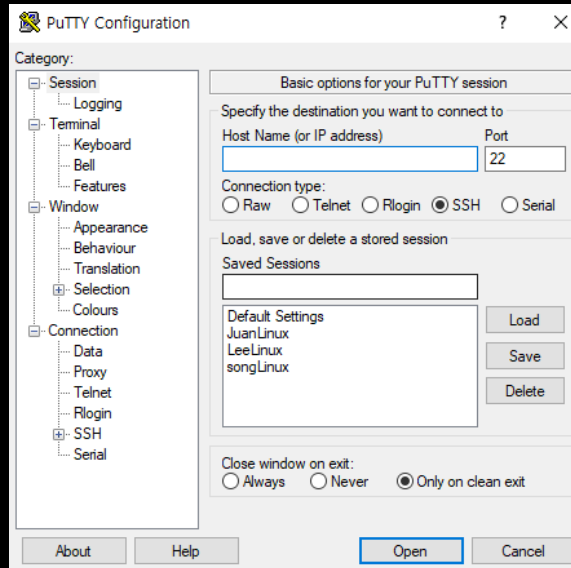
## 3 : 리눅스 서버에 보안터미널서비스(ssh) 설치

JuanLinux 시작 -> root로 로그인 -> apt install ssh

원격 터미널(한글도 지원해주는)에 연결하기 위해 PuTTY설치([www.naver.com](http://www.naver.com) -> putty검색후 설치)

PuTTY에 Saved Sessions에 원하는 이름 설정하고, Host Name에 내 PC의 IP주소를 입력하고 Save누르기

```
JuanLinux [실행 중] - Oracle VM VirtualBox
파일  편집  보기  입력  장치  도움말
Preparing to unpack .../0-openssh-sftp-server_1%3a8.2p1-4_amd64.deb ...
Unpacking openssh-sftp-server (1:8.2p1-4) ...
Selecting previously unselected package libwrap0:amd64.
Preparing to unpack .../1-libwrap0_7.6.q-30_amd64.deb ...
Unpacking libwrap0:amd64 (7.6.q-30) ...
Selecting previously unselected package openssh-server.
Preparing to unpack .../2-openssh-server_1%3a8.2p1-4_amd64.deb ...
Unpacking openssh-server (1:8.2p1-4) ...
Selecting previously unselected package ssh.
Preparing to unpack .../3-ssh_1%3a8.2p1-4_all.deb ...
Unpacking ssh (1:8.2p1-4) ...
Selecting previously unselected package ncurses-term.
Preparing to unpack .../4-ncurses-term_6.2-0ubuntu2_all.deb ...
Unpacking ncurses-term (6.2-0ubuntu2) ...
Selecting previously unselected package ssh-import-id.
Preparing to unpack .../5-ssh-import-id_5.10-0ubuntu1_all.deb ...
Unpacking ssh-import-id (5.10-0ubuntu1) ...
Setting up openssh-sftp-server (1:8.2p1-4) ...
Setting up ssh-import-id (5.10-0ubuntu1) ...
Attempting to convert /etc/ssh/ssh_import_id
Setting up libwrap0:amd64 (7.6.q-30) ...
Setting up ncurses-term (6.2-0ubuntu2) ...
Setting up openssh-server (1:8.2p1-4) ...
Creating config file /etc/ssh/sshd_config with new version
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
Setting up ssh (1:8.2p1-4) ...
Processing triggers for ufw (0.36-6) ...
Processing triggers for systemd (245.4-4ubuntu3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
root@kopo36:~#
```



```
kopo36@kopo36: ~
login as: kopo36
kopo36@192.168.23.105's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-31-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu 28 May 2020 07:15:48 AM UTC

System load:  0.0               Processes:            103
Usage of /:   43.3% of 9.78GB   Users logged in:     1
Memory usage: 19%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%

* MicroK8s passes 9 million downloads. Thank you to all our contributors!

https://microk8s.io/

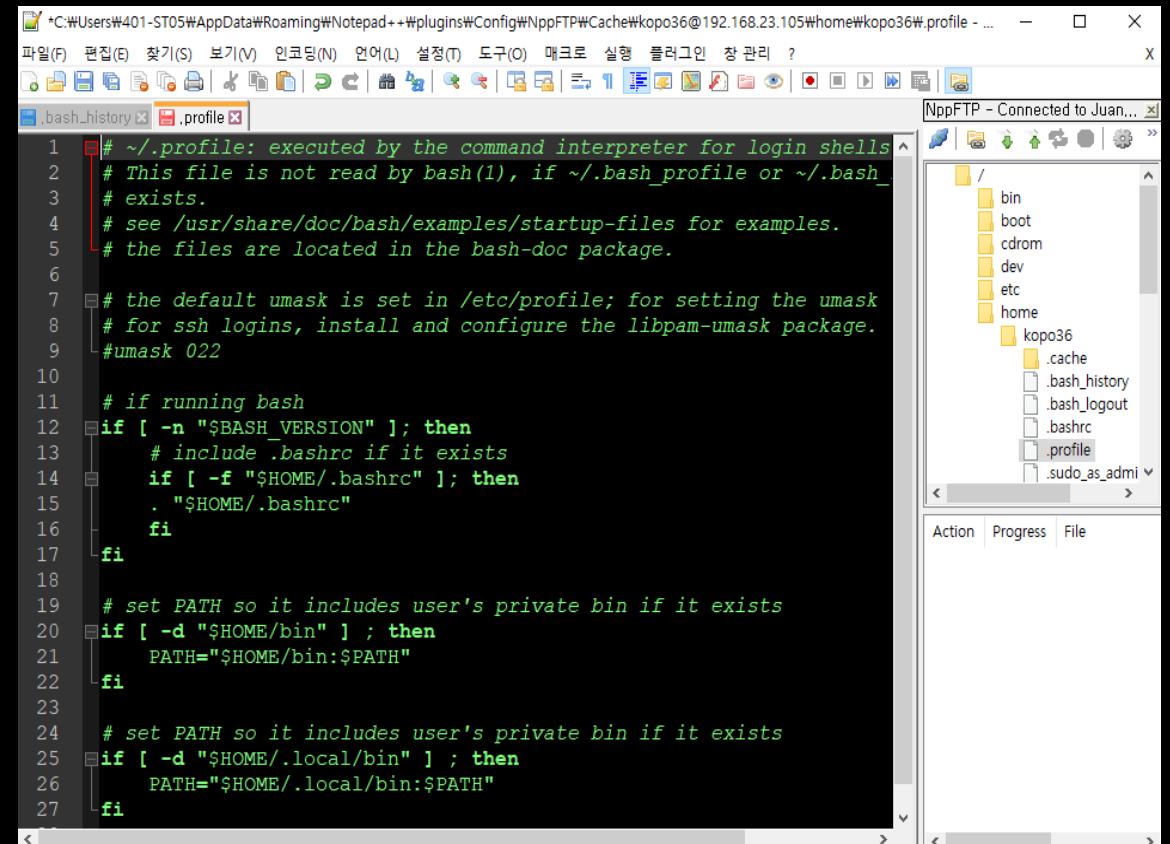
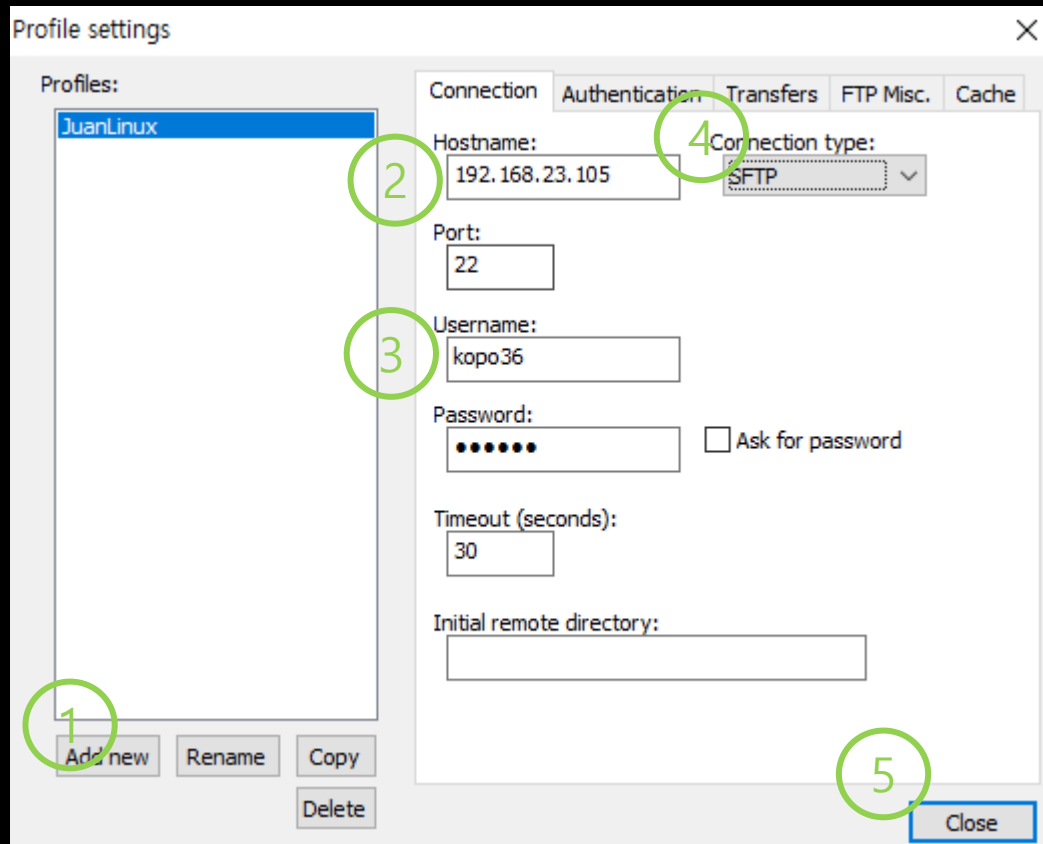
22 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu May 28 06:37:16 2020
kopo36@kopo36:~$
```

## 실습

### 4 : Npp(Note Pad Plus) 다운 (64비트)

실행 -> 플러그인 -> 플러그인 관리 -> NppFTP(파일을 다운로드 받아오는 기능) 설치 -> 플러그인에 NppFTP설치된거 확인 -> show NppFTP window -> 오른쪽에 뜬 설정 클릭 -> Profile settings



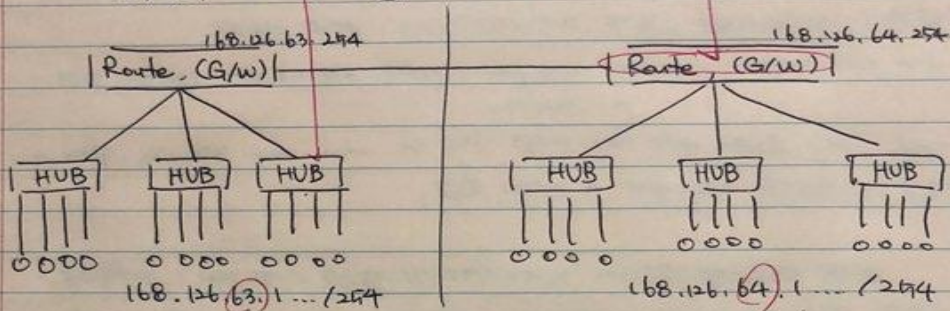


## 2. 개방화경 구조

IP, 라우팅, DNS, DHCP/고급IP, 공인/사설

### 1) 서버 사이드 설치 (네트워크 이야기 - 1)

- 네트워크 위에서 주소 : IP (인터넷 프로토콜)
- 같은 네트워크 (라브, 브리지, L2), 다른 네트워크 (라우터, 게이트웨이, L3), 인터넷 플러그 (L)
- 원선, 무선, Internet working, TCP/IP



- DNS: Domain Name Service, DNS를 서비스하는 서버, 근원 DNS, 보조 DNS

- 고정 IP (Static IP): IP를 지정, PC에서 지정, 라우터(공공)에서 MAC 어드레스로 고정

- 유동 IP (DHCP - Dynamic Host Configuration Protocol)

유동 IP의 장

- 처음 넷에 접속 시 IP를 순차적으로 배정 받음.

(누가 줄까? 라우터...공공...등의 DHCP 기능으로 준다.)

- NIC (Network Interface card) 번호 IP 부여, 즉 노트북 같은 무선랜 (카드), 무선랜 (카드) 번호 IP가 다르게 부여 받음.

- IP 확인 → PC : ip config / 리눅스 계열 : ifconfig -a

- 안드로이드 / iPhone / 맥북 : 인터넷 App을 설치한 후 거기서 ifconfig -a (rooting 필요)

- 공인 IP (Public IP): 인터넷 상에서 공개적으로 인식 가능한 IP

(서비스를 요청하고, 제공받는 것이 가능) - 하지만 공인 IP는 개수가 제한적, 그렇기에 모든 PC를 단말에게 부여할 수 없다.

- 사설 IP (Private IP): 내부에서 가장으로 사용하는 IP, 인터넷 서비스를 요청하여 사용할 수는 있지만, 서버들로 인터넷 서비스 제공 불가.

- 사설 IP는 인터넷 구조로 나가면서 공인 IP를 여러 단말과 나눠쓰며, 인터넷 공인 IP로 변환 (NAT) 하여 사용된다.

→ 인터넷 공인 IP로 변환

### 1) 서버 사이드 설치

- 특별한 IP : 127.0.0.1 (명명 loopback) 자기 자신의 IP, 라우터 (공공) : 라우터, 공인 IP : 같은 네트워크상 모든 단말. (예) 192.168.23.1, 192.168.23.255

다른사람과 연결 (Port 번호) : IP는 해당 단말 (PC, 서버, 등) 가지지 않는, IP 자체 서비스를 위한 포트번호 체계. ex) 웹서버라면, 주소보고 방문자랑 가지 써야함.

- 몇몇 기본값으로 예약되어 있는 포트번호 : http: 80, https: 8080, mysql: 3306, ssh: 22, netstat -a

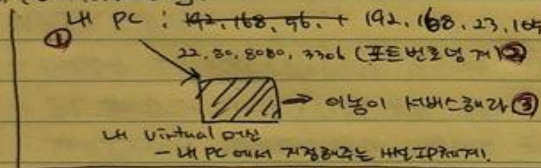
- http://www.naver.com → www.naver.com 라는 이름으로 DNS를 찾아

IP를 받아온 후 http://www (포트번호 80) 를 요청하라.

- 포트포워딩: 사설 IP가 공인 IP (상당 IP) 체계의 서비스를 받기 위한 방법.

- 상단에 위치한 장치에 특정 포트를 등록하여 해당 포트의 요청이 오면 하단 지정된 단말에게 서비스한다.

### #Port Forwarding



- 실행 - Linux (불러)

- apt ← 프로그램 설치

apt install apache2

- console : 서버에 붙어있는 단말. (Linux 명령어 인터프리터)



## 용어 리눅스 - 2강

- IP**
- Internet Protocol.
  - 네트워크 상에서 컴퓨터는 다른 컴퓨터와 구별 될 수 있도록 고유번호를 가지게 되는데 이것을 IP라 한다.
- 라우팅**
- 목적지까지 갈 수 있는 여러 경로를 한가지 경로를 설정해 주는 과정.
  - 하나의 컴퓨터로 자신이 속하지 않은 네트워크에 속한 PC와 통신할때 발생된 패킷을 목적지까지 전달해 주는 과정.
- DNS**
- Domain Name Service
  - 도메인에 연결된 서버 IP를 찾아주는 역할.
  - ex) test.co.kr 도메인을 입력 → 도메인에 연결된 서버 IP로 접속하여 서버 IP를 요청. → 요청받은 네임서버는 test.co.kr의 서버 IP주소를 찾아서 컴퓨터에 주소로 알려줌 → 즉 컴퓨터는 서버 IP로 접속 → test.co.kr 접속.

### NAT / 고정 IP

#### NAT

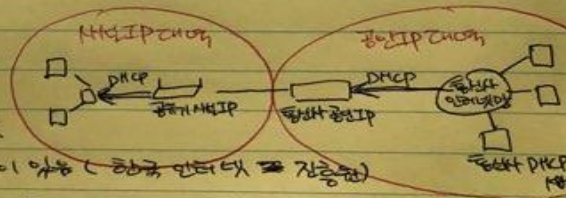
- 네트워크 주소 변환 (Network Address Translation)
- IP 패킷의 TCP/UDP 포트 숫자와 소스 및 목적지의 IP 주소 등을 재기록하면서 라우터를 통해 네트워크 트래픽을 주고받는 기술.
- \* TCP : 인터넷 상에서 데이터를 메시지의 형태로 보내기 위한 IP와 함께 사용하는 프로토콜.
- \* UDP : 데이터를 데이터그램 단위로 처리하는 프로토콜.

#### 고정 IP

- (개의 아이피만 변경되지 않고 사용하는 것.)
- 특성에 따라 고정 IP가 필요한 경우가 있다. ex) 서버 운영 (호스팅)
- ex) 베이비, 쇼핑몰 등

### 공인 IP / 사설 IP

- 인터넷을 통한 통신을 하기 위한 IP.
- 각 나라마다 공인 IP를 관리하는 기관이 있음 (한국 인터넷진흥원)
- 인터넷 상에서 서로 다른 PC끼리 통신하기 위해 필요한 IP.
- 내부 네트워크 상에서 각 컴퓨터간 통신을 하기 위한 IP
- 외부망인 인터넷과 연결이 되는 아이피는 내부망을 구축할때 활용하는 IP.



## 용어 정리 리눅스 - 2강

### 프록시

- 컴퓨터 네트워크에서 패킷이 라우터나 방화벽 같은 네트워크 게이트웨이를 가로지르는 동안 하나의 IP 주소나 포트 번호 집합의 통신 요청을 다른 곳으로 넘겨주는 NAT의 응용.
- 원격 컴퓨터 (예: 인터넷상의 컴퓨터)가 근거리 통신망 (LAN) 내에 위치한 특정 컴퓨터나 서비스에 연결할 수 있게 함.

### SSH

- Secure Shell Protocol
- 네트워크 프로토콜 중 하나로 컴퓨터와 컴퓨터가 인터넷과 같은 Public Network를 통해 서로 통신을 할 때 보안적으로 안전하게 통신을 하기 위해 사용하는 프로토콜.

### PUTTY

- 가상 단말기 프로그램.
- 리눅스에는 Telnet 이라는 있는데, 리눅스 서버의 IP에 접속하여, 원격으로 작업을 할 수 있게 해주는 서비스.

### 포트

- 컴퓨터의 주변 장치와 연결하기 위한 연결단.
- 소프트웨어에선 네트워크 서비스나 특정 프로세스를 식별하는 논리적인.
- 웹 서비스를 할 수 있는 환경을 구축하기 위해 사용하는 소프트웨어.
- 하드웨어적 : Website의 컴포넌트 파일들을 저장하는 컴퓨터
- 소프트웨어적 : 웹 사용자가 어떻게 호스트 파일들에 접근하는지 관리.

### 웹 서버 (Apache)

아파치 재단에서 개발하는 Java 기반의 웹 브라우저

### WAS (Tomcat)

- Web Application Server.
- Server 단에서 Application을 동작할 수 있도록 지원함.
- 동적인 요구에 대응하기 위해 처리할 명령으로 변환, Web Client (브라우저)에게는 결과값만 전송함.

아파치 재단에서 개발하는 Java 기반의 웹 브라우저

웹 브라우저, TCP/IP, HTTP, HTTPS, SSL, TLS, etc.