

# - 목 차 -

## 제1장

### 운영체제 기초 활용

1. 운영체제 기초
2. Windows 기초
3. Linux 기초

## 제2장

### 데이터베이스 기초 활용

1. 데이터베이스 개념
2. 데이터베이스 관리 시스템
3. 데이터 모델링
4. 데이터베이스 기본 명령어

## 제3장

### 네트워크 기초 활용

1. 네트워크 들어가며
2. Cable
3. 네트워크 장비
4. OSI 7계층
5. TCP/IP
6. IP 기초
7. Packet Tracer 가상 환경구성
8. Routing
9. DHCP
10. NAT

## 제4장

### 기본개발 환경 구축

1. 기본 개발환경 구축하기
2. 컴퓨터 시스템의 구성요소
3. 컴퓨터 시스템의 데이터 표현
4. 프로그래밍 이해
5. 불 대수와 디지털 논리회로



## **제1장**

# **운영체제 기초 활용**



## 1. 운영체제 기초

### (1) 운영체제란?

- OS(Operating System) 라고도 한다.
- 컴퓨터 시스템의 자원(CPU, 주기억장치, 보조기억장치 등)들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 사용자와 하드웨어 사이의 인터페이스 역할을 하는 시스템 소프트웨어의 일종이다.



### (2) 운영체제 종류 및 구분

#### 1) 개인용(클라이언트용) 운영체제

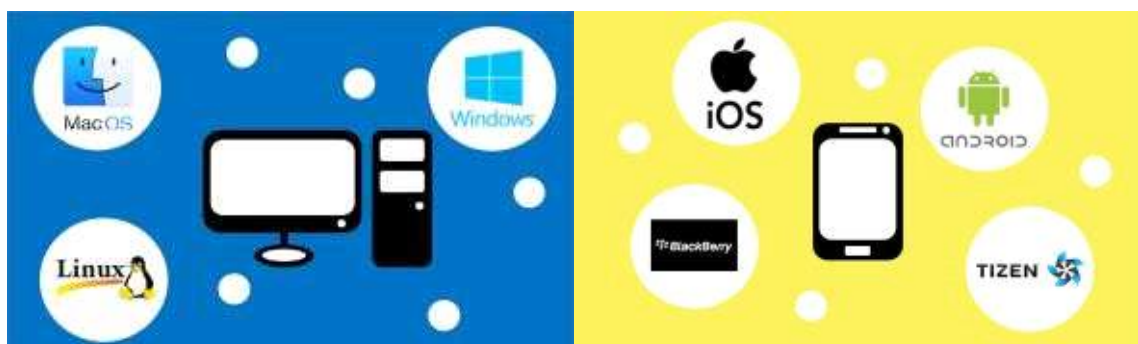
- 일반적인 OS 의 역할에 충실한 운영체제로써, 개인 및 기업에서 클라이언트에 설치해야 사용한다.
- 대표적인 클라이언트용 운영체제 : Windows, Linux, Mac OS 등

#### 2) 서버용 운영체제

- 기본적인 OS 의 역할은 그대로 유지하고, 다수의 사용자에게 서비스를 제공하기 위한 다양한 기능들이 제공되며, 클라이언트용에 비해 강화된 보안 기능을 제공한다.
- 대표적인 서버용 운영체제 : Linux, Windows Server 등

#### 3) 모바일 운영체제

- 개인용 PC 또는 서버 외에 모바일 기기(스마트폰, 태블릿 등)에 사용되는 운영체제로써, 일반 운영체제에 비해 크기(용량)가 작다.
- Android, IOS, Windows Mobile, BlackBerry, TIZEN 등



< 개인용 및 서버용 운영체제 >

< 모바일용 운영체제 >

### (3) 운영체제 설치를 위한 기초 지식

#### 1) CMOS 설정(BIOS설정)

- CMOS(Complementary metal-oxide semiconductor)  
: 부팅에 필요한 각종 설정 내용들을 저장하고 있는 반도체
- BIOS(Basic Input Output System) 또는 ROM BIOS  
: CMOS 에 저장된 설정들을 변경하기 위한 유틸리티가 저장되어 있는 ROM
- 진입 방법은 제조사마다 다르지만, 보통 전원 인가 후 F2 또는 Del 키 눌러서 진입
- CUI 또는 GUI 화면으로 구성됨(제조사에 따라 다름)

CMOS 설정 화면 (삼성전자 PC의 CMOS 화면)



#### < 메인(Main) 메뉴 >

- 주로 기본적인 시스템 정보가 표시됨
- 언어 설정이나 시각, 디스크 설정 등



#### < 메인(Main) 메뉴 - Disk Menu >

- 현재 설치된 디스크 관련 정보가 표시됨
- AHCI Mode 는 Windows 7부터 기본적으로 사용되는 모드이므로 Auto 로 사용하면 됨



#### < 고급(Advanced) 메뉴 >

- CPU Control : CPU 제어 관련 기능
- Peripheral Configuration  
주변 장치 제어 관련 기능  
(USB, 내장 LAN, 사운드 등)
- 고급 환경 제어 가능  
WOL(Wake On Lan) 등 설정 가능



- < 보안(Security) 메뉴 >
- 관리자, 사용자, 디스크 비밀번호 설정



- < 부팅(Boot) 메뉴 >
- 부팅 관련 옵션 설정
  - Boot Device Priority 메뉴에서 장치별 부팅 순서 지정 가능  
(HDD, DVD, USB 의 우선순위 및 HDD 간의 우선순위 등도 설정 가능)




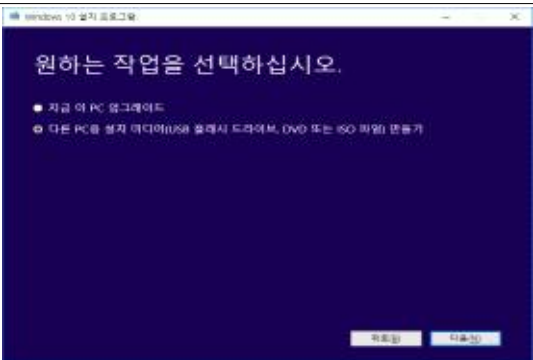
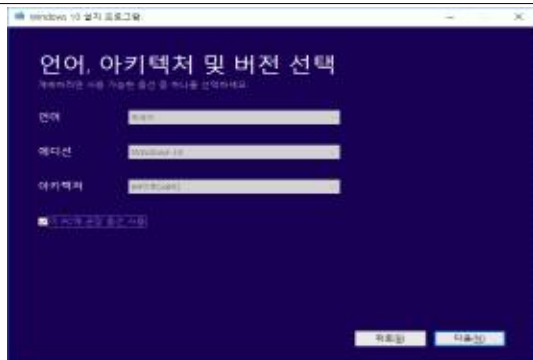
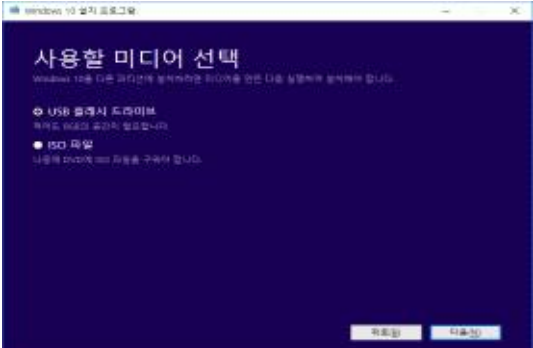
- < 종료(Exit) 메뉴 >
- 변경된 내용을 저장 or 그냥 종료
  - 초기 설정 불러오기
  - 변경 전 내용으로 복구
  - 설정 저장

## 2) DVD 설치 vs USB 설치

- DVD 설치에 비해 USB 설치 속도가 더 빠르다.
- Windows 의 경우 정품 DVD 외에도 Microsoft 제공되는 프로그램을 사용하면 정품 Windows ISO 파일을 다운로드 할 수 있다. (설치 후 정품 인증 필요)
- CentOS Linux 의 경우 CentOS 홈페이지에서 ISO 파일을 다운로드 할 수 있다.

<https://www.microsoft.com/ko-kr/software-download/windows10>



	
	<p>1) USB 플래시 드라이브 선택 시 USB 메모리를 Windows 부팅이 가능한 설치용 USB 로 바꾸어 준다. 단, 기존에 저장된 데이터는 초기화 됨</p> <p>2) ISO 파일 선택 시 차후에 설치 DVD 또는 설치 USB 로 제작할 수 있도록 ISO 파일을 다운로드 할 수 있다.</p>

## 3) 파티션 및 포맷

- 파티션(Partition) : 물리적인 하드디스크를 논리적으로 분할한 단위.  
만약 파티션을 2개로 분할할 경우 1개 디스크를 2개의 디스크처럼 사용 가능
- 포맷(Format) : 해당 디스크(또는 파티션)에 저장된 데이터를 전부 삭제(초기화) 단, 일반적인 포맷은 물리적으로 모든 파일을 지우는 것이 아니라, 파일이 저장된 곳을 담고 있는 테이블 정보만 삭제하므로 경우에 따라 복구가 가능하며, 로우 레벨 포맷(Low Level Format)을 수행할 경우 실제 공장 초기화 상태로 포맷을 수행함



## 2. Windows 기초

### (1) Windows 란?

- 마이크로소프트(Microsoft, MS)사에서 개발한 OS로, 데스크탑에서 사용하는 OS 중 가장 높은 점유율을 가지고 있다. 상용 소프트웨어로써 유료로 구매해야 사용가능하며, Windows 10의 경우 이전 버전으로부터의 무료 업그레이드를 제공해 주고 있다.

### (2) Windows 버전

- 1) Windows Client Version : 개인 또는 기업에서 사용하는 클라이언트용 윈도우즈

기능 \ 에디션	Home	Pro	Enterprise	Education
장치 암호화	✓	✓	✓	✓
도메인 조인		✓	✓	✓
그룹 정책 관리		✓	✓	✓
BitLocker		✓	✓	✓
할당된 액세스 8.1		✓	✓	✓
원격 데스크톱		✓	✓	✓
클라이언트 Hyper-V		✓	✓	✓
직접 액세스		✓	✓	✓
Windows To Go Creator			✓	✓
AppLocker			✓	✓
BranchCache			✓	✓
그룹 정책을 통한 시작 화면 제어			✓	✓

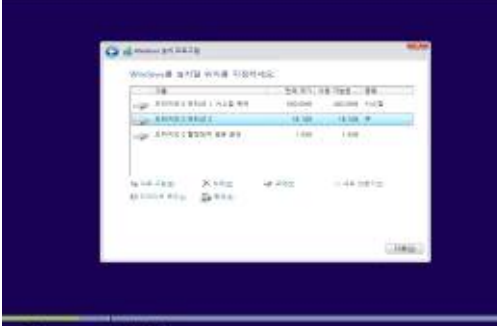






### 2) Windows Server Version

- Windows 8.1 과 Windows Server 2012 버전부터 외관상 차이가 거의 없어지기 시작했으나, 내부적으로 제공되는 기능 및 패키지 등이 다름
- 서버관리자 기본 탑재, Active Directory Domain Controller 역할 수행, 전용 보안 업데이트 등 서버 환경에 필요한 패키지 또는 기능을 제공함
- Windows Server 2012 => Windows Server 2016 => Windows Server 2019
- Standard Edition 과 Datacenter Edition 으로 구분됨

### (3) Windows 설치


Windows 10 설치 과정	
	<p>Window 디스크(DVD) 또는 USB를 넣고 전원을 켜면 다음과 같은 화면이 출력되며, 이 때 아무키나 누르면 Windows 설치 프로그램이 구동되기 시작한다. (단, CMOS 설정 내의 부팅 우선순위에서 DVD 또는 USB가 HDD 보다 낮을 경우 HDD로 부팅됨.</p>

	
	
<p>시리얼 입력하거나 [제품 키 없음]으로 차후 인증 가능</p>	<p>자신에게 맞는 버전(Home 또는 Pro) 선택</p>
	
<p>라이선스 동의</p>	<p>새로 설치할 경우 사용자 지정 클릭</p>
	
<p>새 디스크일 경우 파티션 설정 필수 - 새로 만들기</p>	<p>생성할 파티션 크기 입력(전체 사용 또는 분할)</p>

	
<p>파티션 생성 후 남은 공간은 다른 파티션으로 생성</p>	<p>설치 드라이브 선택 후 다음 클릭 시 설치 진행</p>
	
<p>기본 설정 사용</p>	<p>일반적으로 개인의 경우 [내가 소유합니다] 선택</p>
	
<p>Microsoft 계정 입력하거나, [이 단계 건너뛰기] 선택</p>	<p>생성할 계정명과 패스워드 입력 후 설치 완료됨</p>
	
<p>설치 완료 후 윈도우 부팅 화면</p>	

#### (4) Windows 기본 설정

##### 1) 기본 시스템 정보 확인 : 시작메뉴 우클릭 - 시스템(Y)



[ 디바이스 사양 ]

디바이스 이름 - 컴퓨터 이름 (변경 가능)

프로세서 - CPU 정보

설치된 RAM - 메모리 용량

시스템 종류 - CPU, OS 종류 (32bit, 64bit)

[ Windows 사양 ]


에디션 - Windows 에디션(Home, Pro)

버전 - 현재 Windows 설치 버전

※ Windows 버전별로 표시 형태 다름

##### 2) 장치 관리자 확인 : 시작메뉴 우클릭 - 장치 관리자(M)

- 컴퓨터에 설치되어 있는 각종 하드웨어들을 Windows에서 인식 후 표시해 놓은 곳
- 일반적으로 장치에 알맞은 드라이버가 자동으로 설치되지만, 간혹 올바른 드라이버가 설치되지 않아 오류가 표시되어 있는 경우가 있음 => 해당 장치의 제조사 홈페이지에서 각 Windows 버전에 맞는 드라이버를 다운로드 후 설치필요



[ 네트워크 어댑터 ]

설치된 NIC(= 랜카드) 표시됨.

이 장치에 오류가 표시되어 있을 경우 IP 설정을 하더라도 인터넷 접속이 불가능

[ 디스플레이 어댑터 ]

설치된 그래픽카드 표시됨.

이 장치에 오류가 표시되어 있을 경우 화면 해상도 등이 정상적으로 설정되지 않음

##### 3) IP 주소 및 통신 연결 상태 확인


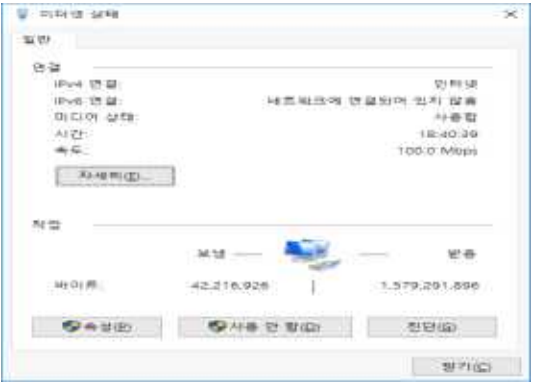
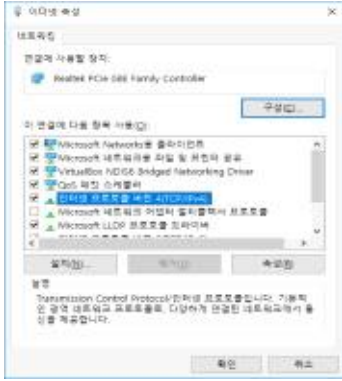


#### IP 주소 설정

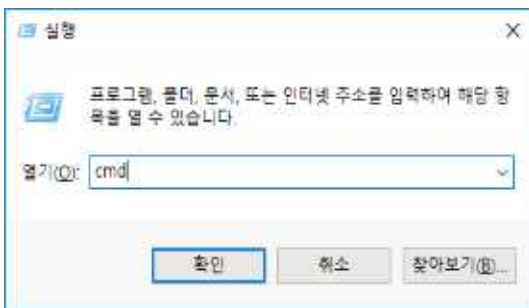

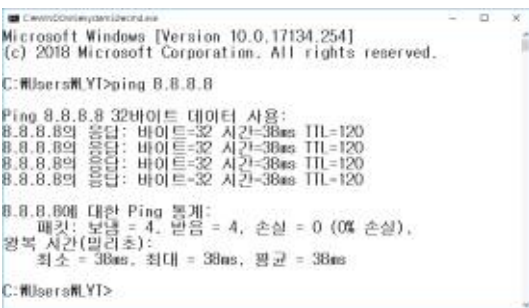
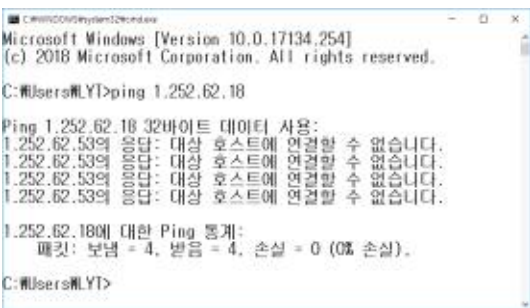


네트워크 아이콘에 느낌표 있을 경우  
인터넷 접속 불가

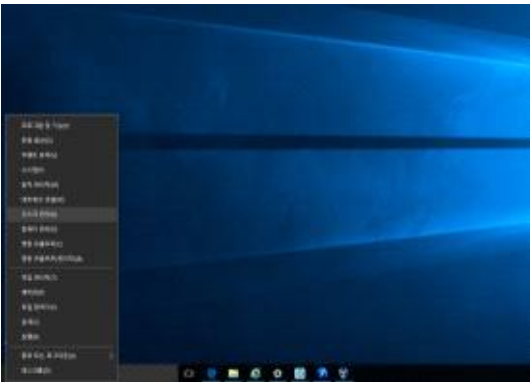
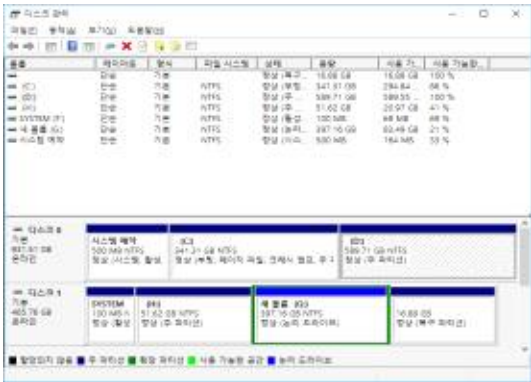


[네트워크 및 공유 센터] 클릭

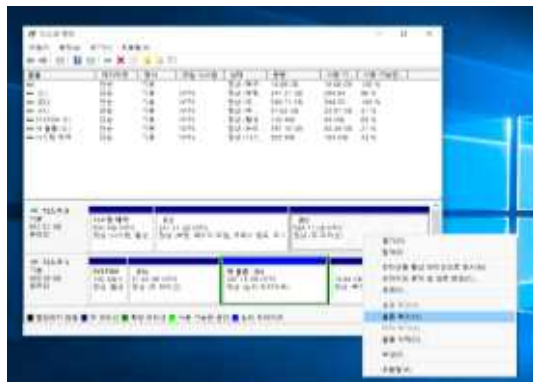
<p>우클릭 - [네트워크 및 인터넷 설정 열기] 선택</p>  <p>활성 네트워크 목록 중 해당 네트워크의 [이더넷] 클릭</p>	 <p>[자세히] 클릭 시 현재 설정 정보 조회 가능 [속성] 클릭 시 설정 화면 이동</p>
 <p>[TCP/IPv4] 선택 후 더블클릭 또는 [속성] 클릭</p>	 <p>처음 설정 시 자동으로 IP 주소 받기 모드(DHCP)가 선택되어 있으므로 공유기 등의 환경 사용 시 IP 주소가 자동으로 설정됨 (주로 192.168.x.x)</p>
 <p>정상적으로 설정이 완료되고, 네트워크 연결이 성공하면 느낌표 아이콘이 사라진다.</p>	

통신 연결 상태 확인	
<p>PING : 목적지 IP 주소까지의 통신 가능 여부를 확인하는 명령어이자 프로그램</p> <p>Window + R 키를 눌러 실행 창을 띄운 후 cmd 입력(명령프롬프트 실행)</p> <p>ping [목적지IP주소] 의 형태로 사용</p>	
	
명령프롬프트 실행	명령프롬프트 실행 결과
	
8.8.8.8 주소로 ping 시도 후 성공했을 때 메세지	1.252.62.18 주소로 ping 시도 후 실패했을 때 메세지

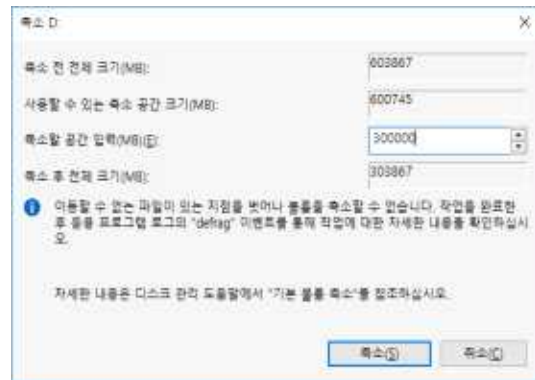
#### 4) 디스크 관리(정보 확인, 파티션, 포맷, 오류 검사, 조각 모음)

파티션 관리	
윈도우 설치 후에도 사용자가 하드디스크 파티션을 직접 관리할 수 있다.	
	
시작메뉴 우클릭 - 디스크 관리	디스크 관리창

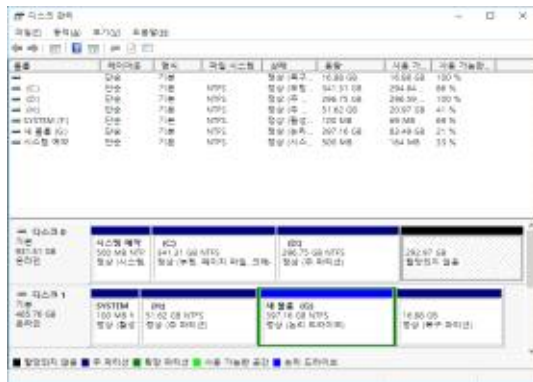




## 기존 파티션 우클릭 - 볼륨 축소



기존 디스크에서 축소할 크기  
(새 파티션 크기) 입력



축소된 공간만큼 새로운 공간  
표시됨(할당되지 않음)



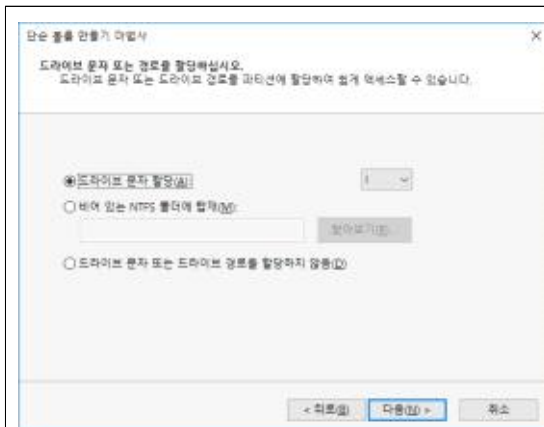
새로운 파티션 우클릭 - 새 단순 볼륨



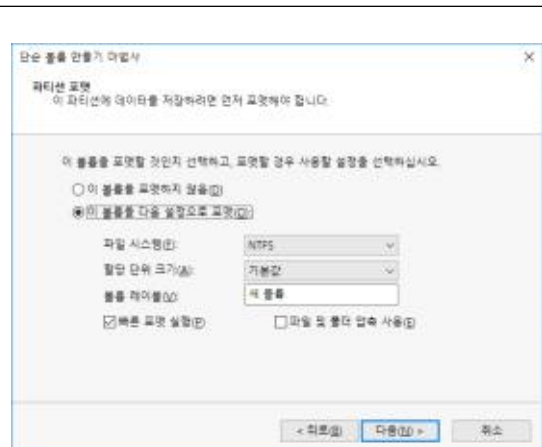
## 마법사 실행



새 파티션 크기 입력



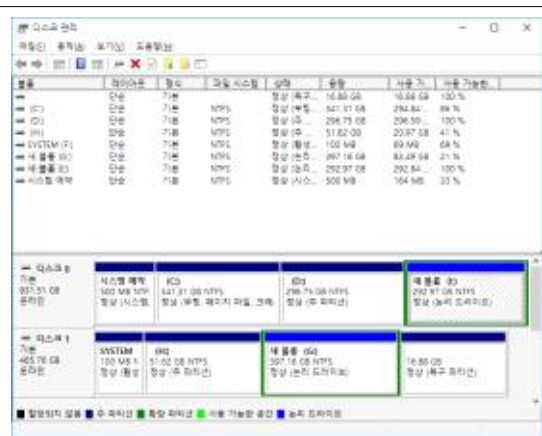
드라이브 문자(C, D...) 선택  
(비어있는 문자 중 선택)



새 파티션 포맷(볼륨 레이블 변경 가능)



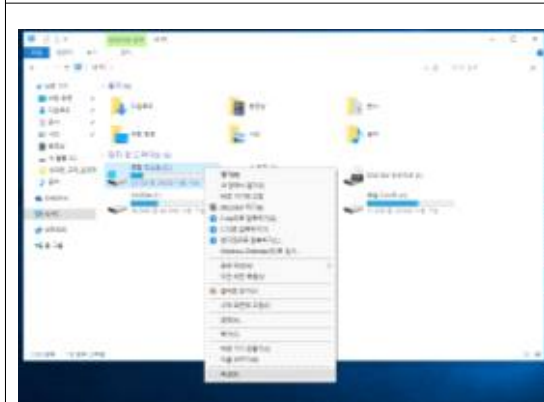
파티션 생성 완료



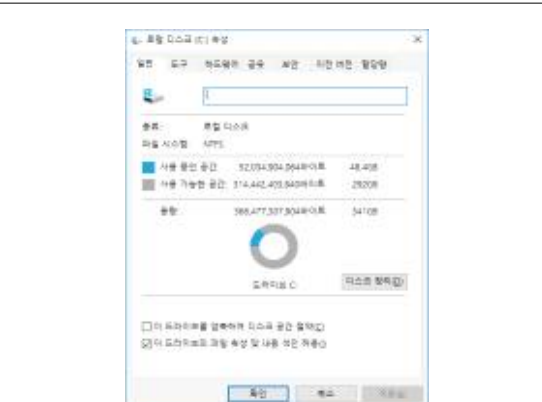
디스크 목록에 새 파티션(새 볼륨)  
표시됨

### 디스크 관리 도구(오류 검사, 최적화 및 조각 모음)

- 오류 검사 : 하드디스크의 논리적인 오류를 검색하여 수정할 수 있다.
- 조각 모음 : 디스크 사용 시 발생하는 파일들의 조각을 모아 떨어진 성능을 향상시킨다.



관리할 디스크 우클릭 - 속성

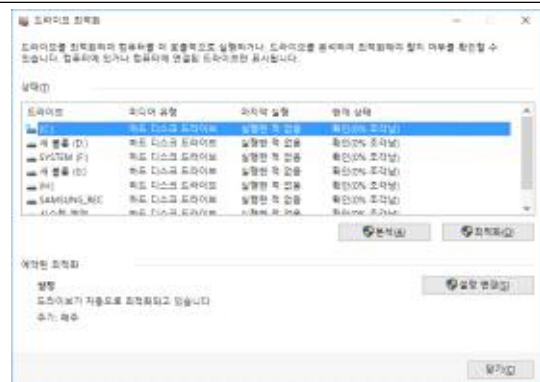


[도구] 탭 클릭

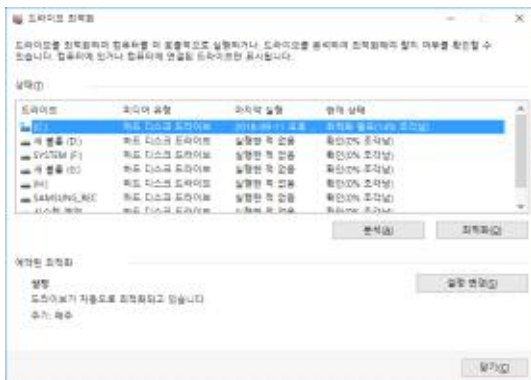




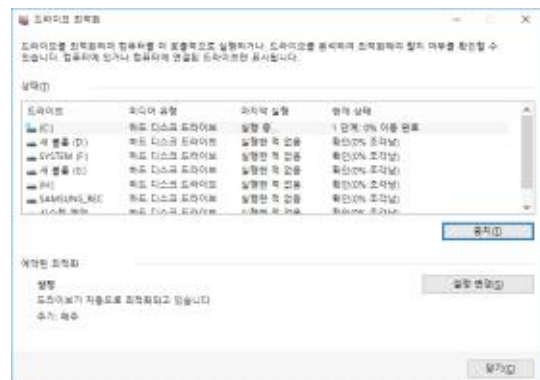
원하는 작업 항목 실행



최적화의 경우 원하는  
드라이브 클릭 - 분석 버튼 클릭하여  
분석 실행

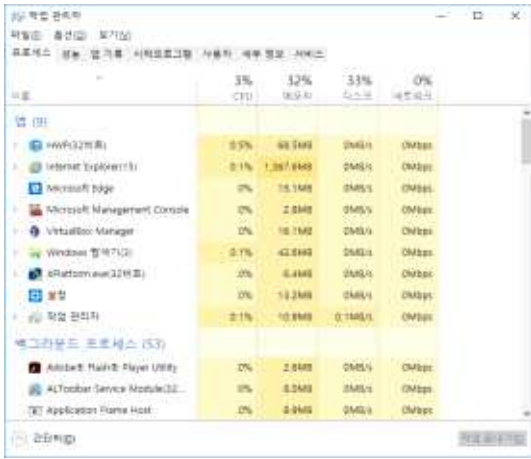
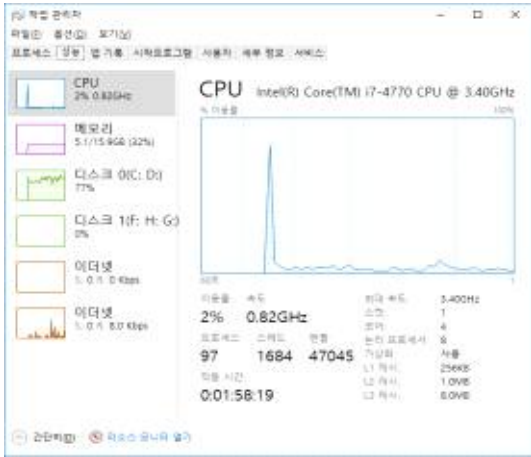
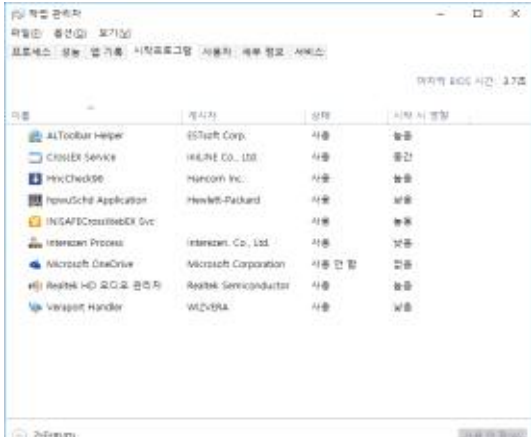


분석 완료 시  
현재 상태에 분석 결과 표시됨  
최적화 버튼 클릭하여 최적화 실행



최적화 실행 중일 경우  
현재 상태에 작업 진행 상황 표시됨

## 5) 작업 관리자

작업 관리자 실행 방법	
<ul style="list-style-type: none"> <li>- 시작메뉴 우클릭 &gt; 작업관리자 실행</li> <li>- 윈도우키 + R 키를 눌러 실행창에 taskmgr 입력하여 작업관리자 실행</li> <li>- Ctrl + Shift + ESC 키를 눌러 작업관리자 실행</li> </ul>	
작업 관리자 메뉴 설명	
	<p>[프로세스] 탭</p> <p>현재 사용 중인 어플리케이션에 대한 관련 프로세스 목록과 해당 프로세스의 CPU, 메모리, 디스크, 네트워크 사용량을 표시하는 탭</p> <p>정상 종료되지 않은 프로세스도 표시되므로 해당 프로세스에 대한 강제 종료 (작업 끝내기)를 수행할 수 있다.</p>
	<p>[ 성능 ] 탭</p> <p>현재 PC의 리소스 사용량에 대한 모니터링이 가능하다.</p> <p>CPU, 메모리, 디스크, 네트워크에 대한 현재 및 누적 사용량 등에 대한 각종 통계들이 그래프 및 수치 데이터로 표시되어 쉽게 확인할 수 있다.</p>
	<p>[ 시작프로그램 ] 탭</p> <p>이전의 Windows 7 까지는 별도로 관리하였으나, Windows 8부터 시작프로그램 관리가 작업관리자와 결합되었으며, 이 메뉴에서 시작프로그램에 대한 사용 여부를 결정할 수 있다.</p>

### [ 세부 정보 ] 탭

현재 실행 중인 프로세스에 대한 상세한 정보를 표시하는 탭

PID(프로세스 ID), 현재 상태, 사용자명, 점유율, 프로세스 상세 설명 등이 표시되어 프로세스에 대해 좀 더 많은 정보를 확인하고 강제 종료 등의 작업을 수행할 수 있다.

작업 우선순위 지정을 통해 특정 프로세스에 대한 리소스 할당 우선순위를 높이거나 낮춤으로써 실행 성능에 대한 제어가 가능하다.

이름	PID	상태	사용자 이름	CPU	메모리(가)	설명
ApplicationFrameHost.exe	3520	실행 중	SYSTEM	00	9,094 K	Application Frame...
atdsvc.exe	4712	실행 중	SYSTEM	00	8,142 K	AtD Service
atdsvc.exe	1872	실행 중	SYSTEM	00	5,512 K	atdsvc.exe
audiodg.exe	4004	실행 중	LOCAL S...	00	4,744 K	Windows 오디오 경...
BrowserBroker.exe	4408	실행 중	SYSTEM	00	4,196 K	Browser Broker
ClientServerMain.exe	6668	실행 중	SYSTEM	00	15,884 K	Client Server Main...
CrossKService.exe	3452	실행 중	SYSTEM	00	392 K	CrossK Service
CSOService.exe	568	실행 중	SYSTEM	00	712 K	Client Server Servi...
CSOService.exe	884	실행 중	SYSTEM	00	1,264 K	Client Server Servi...
DeviceAssociation.exe	2480	실행 중	LOCAL SE...	00	3,212 K	Device Association...
DeviceAssociation.exe	8848	실행 중	SYSTEM	01	11,716 K	Client Host Servi...
DeviceAssociation.exe	9804	실행 중	SYSTEM	00	2,900 K	COM+ Eventlog
DeviceAssociation.exe	98	실행 중	Device-1	00	121,888 K	이스라엘 및 문장...
DeviceAssociation.exe	7848	실행 중	SYSTEM	00	836 K	Device Association...
DeviceAssociation.exe	4288	실행 중	SYSTEM	00	43,864 K	Windows 탐색기
DeviceAssociation.exe	8808	실행 중	SYSTEM	00	2,852 K	Address Flash R...
DeviceAssociation.exe	8020	실행 중	SYSTEM	00	1,800 K	Universal Print Dr...
DeviceAssociation.exe	2428	실행 중	SYSTEM	00	848 K	Hercules PrintTray...
DeviceAssociation.exe	812	실행 중	SYSTEM	00	700 K	Forwarded Applicat...
DeviceAssociation.exe	1804	실행 중	SYSTEM	00	72,876 K	Task Manager
DeviceAssociation.exe	1812	실행 중	SYSTEM	00	912 K	Internet Service P...
DeviceAssociation.exe	3904	실행 중	SYSTEM	00	2,696 K	Internet Process
DeviceAssociation.exe	3720	실행 중	SYSTEM	00	18,904 K	Internet Explorer


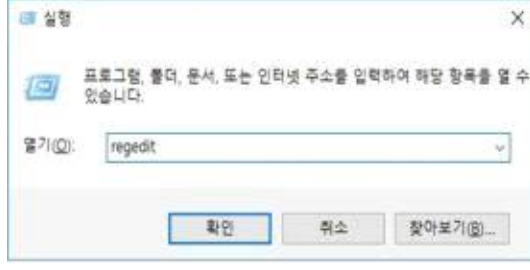
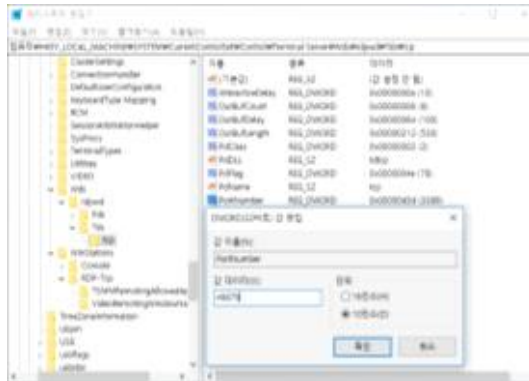


### [ 서비스 ] 탭

현재 윈도우에서 수행되는 각종 서비스에 대한 정보 확인 및 실행, 중지 여부를 설정할 수 있다.

자세한 작업을 수행하기 위해서는 아래에 있는 [서비스 열기]를 클릭하거나, 윈도우 + R 키를 눌러 실행 창에 services.msc 명령어를 입력하여 서비스 창을 띄울 수 있다.

이름	PID	상태	그룹
WSearch	9968	Windows Search	실행 중
WinProtonSvc		Windows Media Player Network...	중지됨
WinAgSvc		Win Performance Adapter	중지됨
WinDefend	2092	Windows Defender Service	실행 중
WinDefend	1028	Windows Defender Network Ins...	실행 중
WinDefend	2856	Block Level Backup Engine Servi...	중지됨
VSS	3132	Volume Shadow Copy	실행 중
VSS	808	Credential Manager	실행 중
UACService	8720	Interactive Services Detection	중지됨
TrustedInstaller		Windows Modules Installer	실행 중
TrustedInstaller		Storage Tiers Management	중지됨
TrustedInstaller		Software Protection	중지됨
TrustedInstaller	1844	Print Spooler	실행 중
TrustedInstaller		SNMP Trap	중지됨
TrustedInstaller		Service Data Service	중지됨
TrustedInstaller	2448	Windows Remediation Service	실행 중
TrustedInstaller	808	Security Accounts Manager	실행 중
TrustedInstaller		Abuse Safe Transaction Service	중지됨
TrustedInstaller		Remote Procedure Call (RPC) L...	중지됨
TrustedInstaller		Performance Counter (DUL) Host	중지됨
TrustedInstaller		Office At Source Control	중지됨

6) 원격 데스크톱 (Server 설정은 Windows Pro 버전 이상에서 설정 가능)

원격 데스크톱 설정 및 접속	
 <p>시작메뉴 우클릭 - 시스템 원격 데스크톱 - 원격 데스크톱 활성화 항목 켜기</p>	 <p>윈도우 + R 키 눌러 실행 창 띄운 뒤 regedit 입력하여 레지스트리 편집기 실행</p>
	<p>컴퓨터\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\Tds\tcp 항목과 컴퓨터\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp 항목을 찾아 PortNumber 더블클릭하여 포트번호 편집 기본 포트는 3389 이지만 보안을 위해 1024 ~ 65535 사이의 다른 포트번호 입력 (10진수 항목 선택하여 변경)</p>
<p>앞에서 설정한 포트의 경우 방화벽에서 허용해줘야 접속이 가능하며 방화벽 설정 방법은 [ 방화벽 및 Windows Defender ] 부분에서 다루도록 한다.</p>	
 <p>다른 클라이언트에서 앞서 설정한 원격 데스크톱 서버에 접속하기 위해 원격 데스크톱 프로그램 실행 시작 - Windows 보조프로그램 - 원격 데스크톱 연결을 실행하거나 실행창에서 mstsc 입력하여 실행</p>	 <p>IP주소:포트번호(서버에서 설정한 포트번호) 형식을 사용하여 접속할 서버의 정보 입력 후 연결 (접속에 필요한 계정은 서버에서 생성)</p>

## 7) 방화벽 및 Windows Defender

### 방화벽 - 특정 패킷을 차단 또는 허용



[제어판] - [Windows Defender 방화벽] 실행



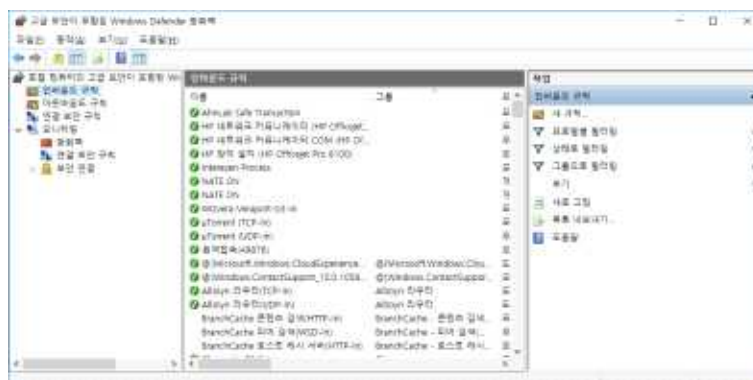
방화벽 실행 시 기본 상태 확인 가능



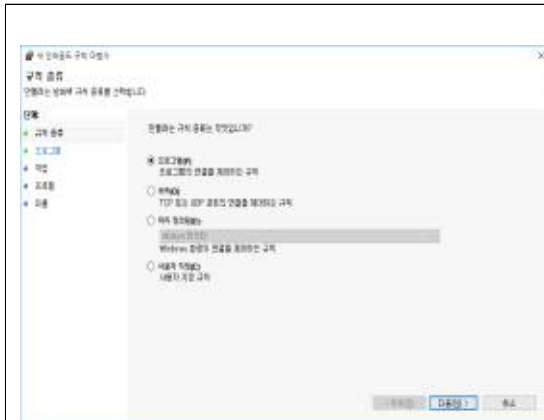
Windows Defender 방화벽 설정 또는 해제 메뉴 선택하여 방화벽 On/Off 설정 가능



Windows Defender 방화벽을 통해 앱 또는 기능 허용 설정 변경 버튼 누른 후 방화벽에서 허용하고자 하는 프로그램 찾아 체크 표시 후 확인



고급 설정 클릭 시 상세한 방화벽 기능 컨트롤이 가능  
 인바운드 규칙 : 외부에서 현재 PC로 들어오는 방향에 대한 방화벽 설정  
 아웃바운드 규칙 : 현재 PC에서 외부로 나가는 방향에 대한 방화벽 설정  
 원하는 항목 선택 후 우측 작업 메뉴의 [새 규칙] 클릭



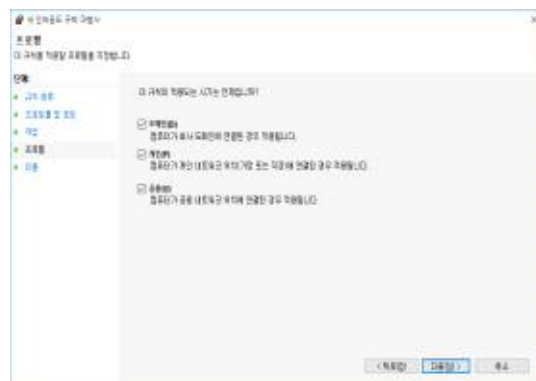
프로그램 선택 시 특정 프로그램에 대한 방화벽 설정이 가능하며, 주로 포트 항목을 선택하여 원하는 포트를 직접 지정하는 방법을 사용함



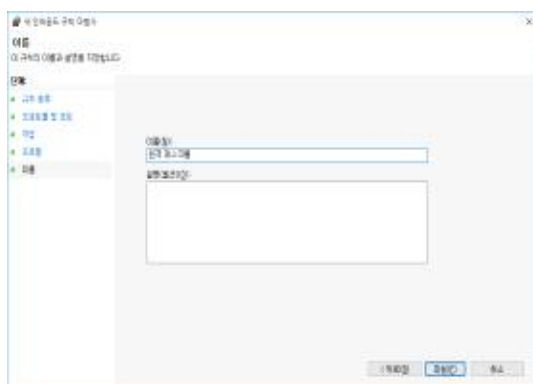
TCP 또는 UDP 포트 타입 선택 후 특정 로컬 포트 항목에 지정할 포트 번호 입력 (ex. 원격 데스크탑 허용 시 3389번 입력)



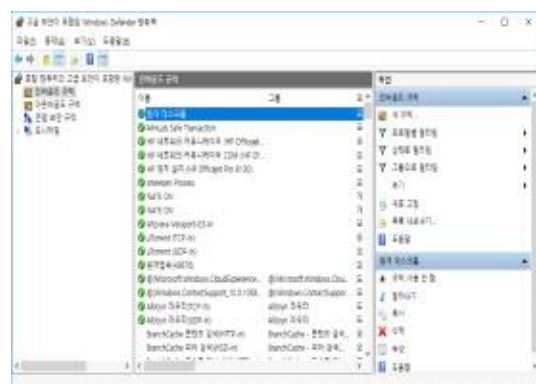
허용/차단 여부 선택



프로필 적용 대상 지정 (네트워크 속성에서 확인 가능)



이름과 설명 입력 후 마침 클릭하여 설정 완료



목록에서 추가된 규칙 항목 확인 가능



## Windows Defender - Windows 에 내장된 바이러스 백신



시작메뉴 - Windows Defender 보안  
센터 실행 후  
[바이러스 및 위협 방지] 클릭



바이러스 및 랜섬웨어 등에 대한  
검사 및 치료 가능





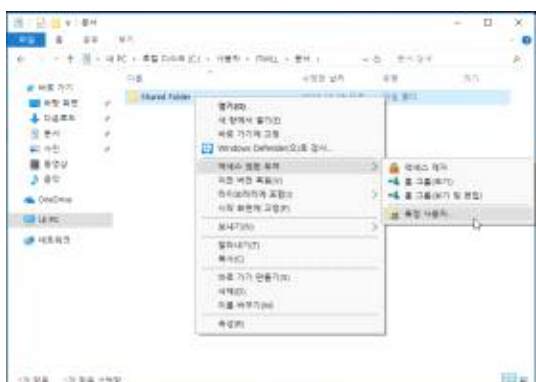
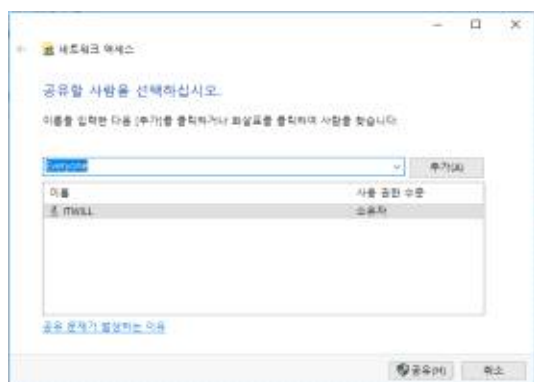


[지금 검사] 클릭 시 기본 검사 실행됨



[고급 검사 실행] 클릭 시  
상세 검사 옵션 선택 가능

## 8) 기본 파일 및 폴더 공유

같은 네트워크 상에서 복잡한 설정 없이 파일 공유

 <p>[네트워크 및 공유 센터] - [고급 공유 설정 변경] 클릭</p>	 <p>현재 프로필에서 [네트워크 검색 켜기] 체크는 옵션 [파일 및 프린터 공유 켜기] 체크는 필수</p>
 <p>공유할 폴더에서 마우스 우클릭 [액세스 권한 부여] - [특정 사용자] 선택</p>	 <p>현재 공유중인 사용자가 표시되며, Everyone 추가 시 모든 사용자에게 공유됨</p>
 <p>사용자 추가 후 [공유] 버튼 클릭 - [완료]</p>	 <p>[탐색기] 주소표시줄에서 '\\IP주소' 형식으로 입력 시 해당 PC의 공유폴더가 표시됨</p>



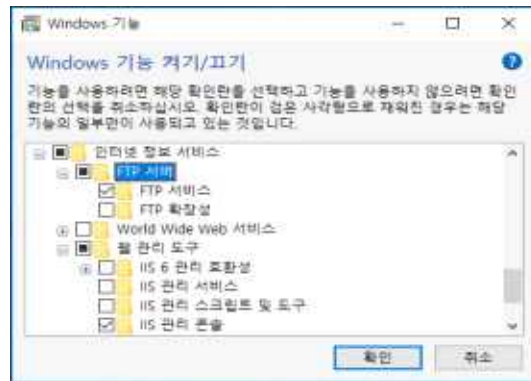
## 9) FTP 서버를 통한 파일 및 폴더 공유

### FTP(File Transfer Protocol, 파일 전송 프로토콜) 설정

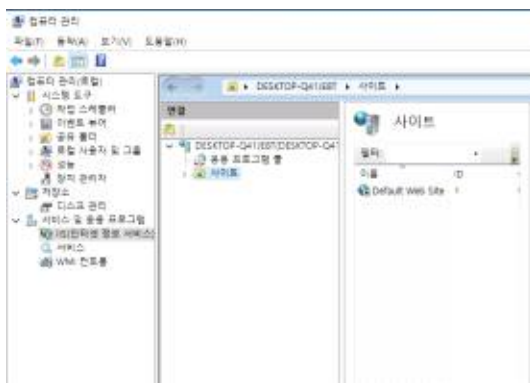
Windows 에 내장된 IIS(Internet Information Services)를 사용하여 FTP 서버 및 Web 서버를 쉽게 구축할 수 있으나 보안상 위험하므로 간단한 공유 용도로만 사용하도록 한다.



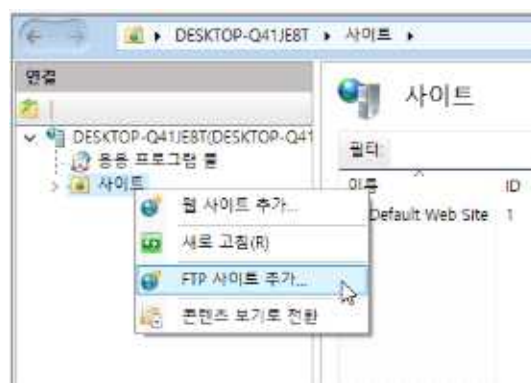
[제어판] - [프로그램] - [프로그램 및 기능]  
[Windows 기능 켜기/끄기] 클릭



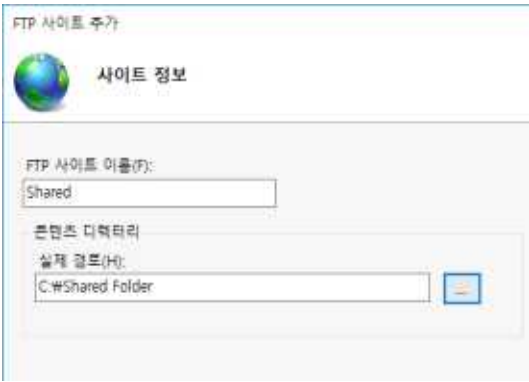


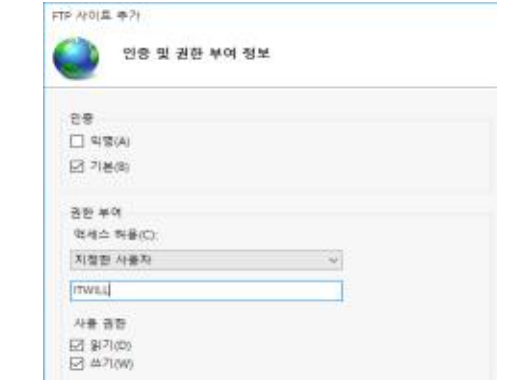
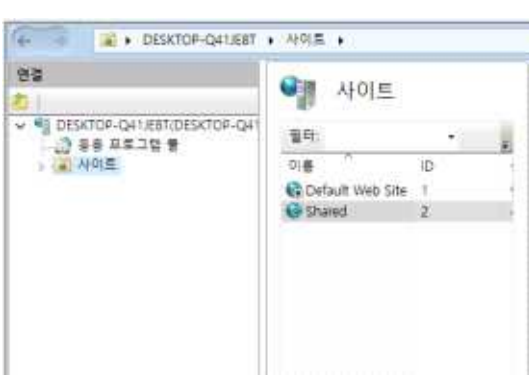
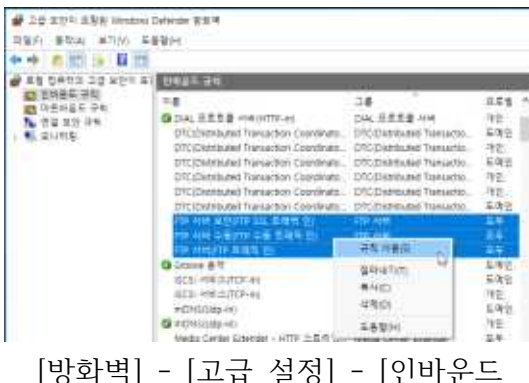
[인터넷 정보 서비스] 항목 중  
[FTP 서버] - [FTP 서비스] 체크,  
[웹 관리 도구] - [IIS 관리 콘솔] 체크 후  
확인 시  
관련 프로그램 추가 및 적용됨



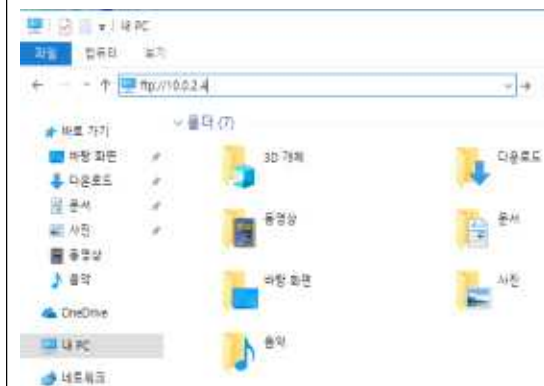
[서비스 및 응용 프로그램] - IIS(인터넷  
정보 서비스) 클릭 후 [연결] 창에서  
PC명 확장



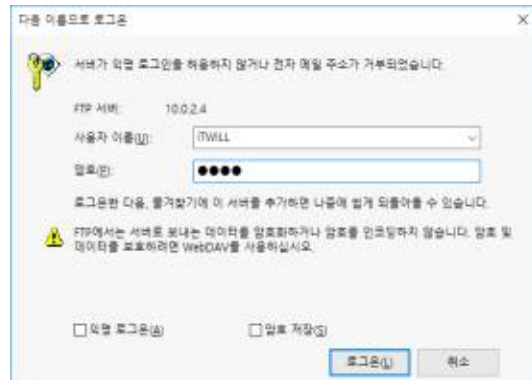
[사이트] 항목에서 우클릭  
[FTP 사이트 추가] 선택

 <p>FTP 사이트 이름 : 클라이언트에게 표시할 폴더 이름 실제 경로 : 클라이언트에게 공유할 폴더 경로</p>	 <p>IP 주소 : 공유에 사용할 자신의 IP 주소 포트 : FTP 기본 포트는 21번이나 변경 가능 SSL : SSL 사용 안 함 선택 (실제로는 보안상의 위험 때문에 인증 사용하는 것이 좋음)</p>
 <p>[인증] 항목은 [기본] 에만 체크 [권한 부여] 항목은 [모든 사용자] 선택 시 로그인 가능한 모든 계정에 접속 권한을 부여한다.</p>	 <p>만약 [지정한 사용자] 선택 후 계정명 입력 시 해당 계정에만 FTP 접속 권한을 부여한다. (단, 지정된 계정 생성이 필요함)</p>
 <p>[마침] 클릭 시 [사이트] 항목에 공유 폴더가 표시됨</p>	 <p>[방화벽] - [고급 설정] - [인바운드 규칙]에서 FTP 서버 항목 선택 - 우클릭 - [규칙 사용] 선택</p>

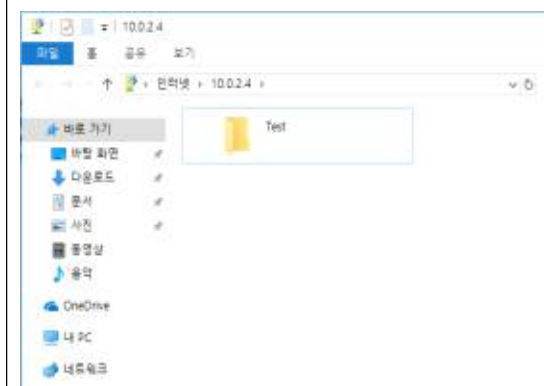
## FTP 접속(윈도우 탐색기 활용)



탐색기 주소창에 'ftp://IP주소' 입력





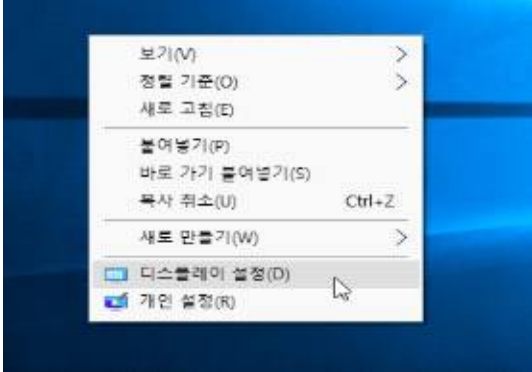

로그인 계정 입력 후 [로그인] 클릭



로그인 성공 시 공유된 폴더의  
내용이 표시됨

윈도우 탐색기 외에도 Filezilla 등의 FTP Client 프로그램을 사용하여 FTP 서버에 접속할 수 있으며, 탐색기보다 더욱 안정적이고 다양한 기능을 활용할 수 있다.

## 10) 기타 설정(전원 관리, 바탕화면 해상도)

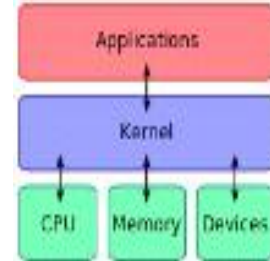
전원 관리	
	
<p>시작 메뉴 우클릭 - [전원 옵션] 실행 후 [화면]의 시간 조절을 통해 모니터 절전 시간 설정</p>	<p>[관련 설정]의 [추가 전원 설정] 클릭 [균형 조정] 항목의 설정 변경 클릭 시 상세 설정 가능</p>
바탕화면 해상도 설정	
<p>해상도(Resolution) : 모니터의 가로 * 세로 점의 갯수를 나타내는 단위. 해상도가 높을수록 그래픽이 선명해지는 대신 점 동일한 모니터에서 표시하는 점의 크기가 작아지므로 전체적인 그래픽 사이즈가 작아짐 (Full HD&lt;1K&gt; : 1920 * 1080, QHD&lt;2K&gt; : 2560 * 1440, UHD&lt;4K&gt; : 3840 * 2160)</p>	
	
<p>바탕화면 우클릭 - 디스플레이 설정</p>	<p>[해상도] 항목에서 원하는 해상도 선택 (그래픽카드에 따라 지원되는 해상도가 다르며, 가장 흔히 사용하는 해상도는 1920 * 1080 이다)</p>

### 3. Linux 기초

#### (1) Linux 특징 및 종류

##### 1) Linux 의 특징

- 핀란드의 리누스 토발즈(Linus B. Torvalds) 에 의해 처음 커널(Kernel)이 개발되었고, 이후 1992년 오픈 소스(Open Source)로써 인터넷 상에 소스코드를 공개했으며, 이후 여러 개발자들에 의해 현재 리눅스의 모습으로 변경되어 왔다.
- 커널<Kernel>이란 운영체제의 핵심으로써 하드웨어와 직접 맞닿아 자원을 효율적으로 사용할 수 있게 해준다.
- 소프트웨어 산업의 상업화를 반대하며 자유 소프트웨어 재단(FSF)을 설립한 리처드 스톨만이 저작권을 의미하는 Copyright의 반대말인 Copyleft 를 모토로 한 GNU 프로젝트에서 일반 공중 사용 허가서<GPL(General Public License)>를 발표했다. 리눅스는 이 GNU 프로젝트와 통합되어 완전히 공개된 운영체제가 되었으며, 누구나 소스코드를 변형할 수 있으며, 변형된 소프트웨어 코드는 누구나 볼 수 있도록 공개되어야 한다는 원칙이 적용되게 되었다.
- 리눅스 커널을 포함한 패키지 대부분이 무료로 제공되며, 사용자가 누구냐에 따라 다음과 같은 특징을 지닌다.




< 리눅스 커널의 역할 >

개인	<ol style="list-style-type: none"> <li>1. 무료</li> <li>2. 오픈 소스 - 커스터마이징 가능</li> <li>3. 일부 게임 제외 대부분의 SW가 윈도우용을 대체</li> <li>4. CUI 방식을 통해 기본적인 프로그래밍 개념 습득</li> <li>5. 네트워크에 대한 개념 등 전반적 이해도 상승</li> <li>6. 다양한 커뮤니티를 통해 대부분의 문제점 해결 가능</li> </ol>
개발자	<ol style="list-style-type: none"> <li>1. 운영체제 구성에 대한 기본적 이해도 상승</li> <li>2. 오픈소스 SW를 활용한 개발 능력 배양</li> <li>3. IoT 등 다양한 플랫폼에 대한 환경 변화 적응에 용이</li> <li>4. SW 배포 전 사전 테스트 환경 구축</li> <li>5. 실제 운영 환경에서의 위험 요소 예측</li> <li>6. 클라우드 환경으로의 진화에 따른 관련 사전 지식 습득으로 활용도 증가</li> </ol>
기업체	<ol style="list-style-type: none"> <li>1. 대부분 무료(사무용 SW 비용 절감) =&gt; 초기 진입 비용이 비교적 저렴</li> <li>2. 오픈소스로 구성된 운영체제를 기업 환경에 최적화 가능</li> <li>3. 대부분의 플랫폼에 마이그레이션 용이</li> <li>4. 타 OS 대비 적은 비용으로 운영 가능 =&gt; 유지비용 감소</li> <li>5. 향후 변화하는 환경에 지속적, 능동적 대처 가능(안정성)</li> <li>6. 최신 기술에 대한 접근성 및 활용도 증대</li> <li>7. 윈도우 대비 악성코드 등으로 인한 기업 생산성 저하 가능성 감소</li> </ol>

## 2) Linux 종류

- 일반 사용자들은 리눅스 커널만으로는 리눅스를 사용할 수 없으며, 커널에 다양한 응용프로그램을 추가해 쉽게 설치할 수 있도록 만든 것을 ‘리눅스 배포판’ 이라고 한다.
- 수백 가지에 해당하는 배포판이 존재하며, 우리나라에서 주로 사용하는 유명 배포판도 10여가지나 된다. 전 세계적으로 가장 유명한 배포판 중 하나가 RedHat(레드햇)사에서 제작한 레드햇 리눅스이다.
- 기업체에서 가장 많이 쓰는 리눅스는 레드햇 계열이다.
- 다음은 대표적인 리눅스 배포판에 대한 비교 목록이다.

RedHat Linux	<b>FEDORA Linux</b> 	1. 레드햇 지원 커뮤니티 프로젝트 2. 6개월 단위의 패키지 업데이트 3. 실험적 기능 및 프로그램 포함 = 베타판 SW 4. 무료 5. 가장 앞서가는 버전 제공
	<b>RedHat Linux</b> 	1. 레드햇 자체 개발 및 페도라 프로젝트 기반 2. 현재 레드햇 리눅스는 RHEL<RedHat Enterprise Linux> 로써 상용소프트웨어를 의미하므로, 유료 구독하지 않을 경우 업데이트 및 기술 지원 불가 3. 최소 1년 이상의 장기 단위의 패키지 업데이트 4. 실험적 기능 보다는 안정적 기능 및 프로그램 위주(검증된 프로그램)로 제공되며, 대부분의 하드웨어와 호환성 뛰어남(배포판 중에서 하드웨어 호환성 아주 좋음)
	<b>CentOS Linux</b> 	1. 레드햇 지원 커뮤니티 프로젝트 2. 레드햇 릴리즈 패키지를 재컴파일하여 배포하므로, 레드햇에서 공개 불가능한 소스를 제외한 나머지는 레드햇과 동일 3. 레드햇 릴리즈 후 일정 기간 (보름~한 달) 후 패키지 업데이트 4. 무료
<b>Debian Linux</b> 		1. 데비안 프로젝트 2. 패키지 설치 및 업그레이드가 단순(apt를 사용한 자동화) 3. 데스크탑, 서버, 모바일기기 등 다양한 하드웨어에서 사용 가능 4. 안정성과 보안성에 중점
<b>Ubuntu Linux</b> 		1. 데비안 리눅스의 안정화 버전 2. 개인용 PC에 최적화 되어 있으며, 서버 버전도 배포 3. CentOS 와 시스템 관련 명령어가 거의 동일 4. GUI 환경이 윈도우와 가장 유사하므로, 초보자 입장에서 가장 사용하기 쉬움



## (2) Linux 설치

### 1) CentOS 다운로드

www.centos.org 사이트에서 버전별 다운로드 가능



① DVD ISO : 일반적으로 배포되는 CentOS 이미지 파일

② Everything ISO : 커널 개발도구, 소스 파일 등 개발환경까지 포함된 이미지 파일

③ Minimal ISO : 최소 설치를 위한 이미지 파일

④ More download choices : 위의 세 가지를 포함한 더욱 다양한 이미지 파일

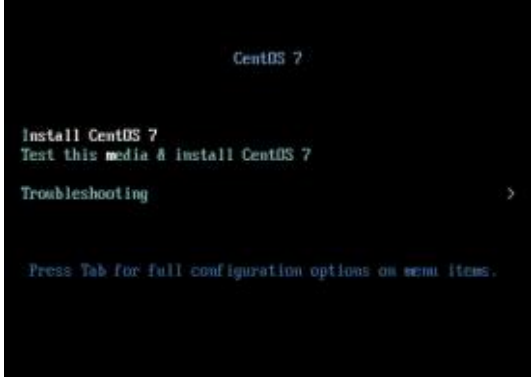
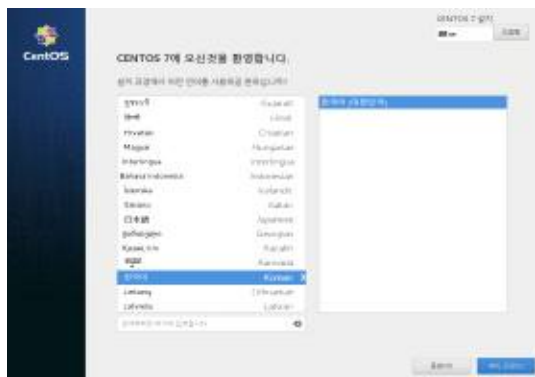
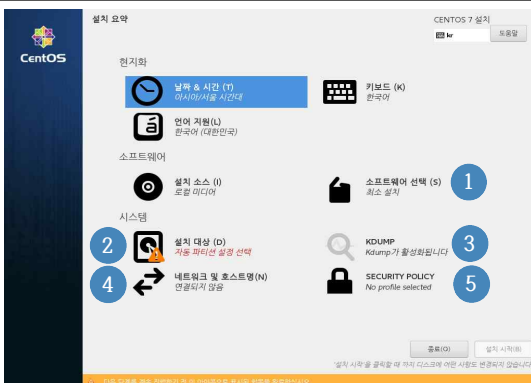



⑤ list of current mirrors : 미러(Mirror) 사이트를 통해 다운로드 가능

※ 미러 사이트란? 각 나라마다 CentOS 설치 이미지를 실제 CentOS 홈페이지 내용과 똑같이 하여 배포하는 사이트로써, 동시 다운로드로 인한 트래픽 과부하를 방지하며, 메인 사이트 접속 장애 시에도 다운로드를 가능하도록 함.

※ Asia - South Korea 목록에서 국내 미러 사이트를 찾을 수 있다.

대표적인 미러 사이트로는 KAIST, Kakao Corp, NAVER 등이 있다.

## 2) CentOS 설치

 <p>첫 화면에서 방향키 사용하여 첫 번째 항목 Install CentOS 7 선택 후 엔터</p>	 <p>한국어 - 한국어 선택</p>
 <p>설치 요약 화면 1번부터 5번까지의 항목 위주로 설정하며, 처음 설치 시 3, 5번의 경우 비활성화한 상태로 설치 (실제로는 5번이 매우 중요함)</p>	 <p>최소설치 모드 : 기본 환경만 설치하고, 나머지는 직접 다운로드 및 설치 (실제 서버 설치에 주로 사용)  GNOME(그놈) 데스크탑 모드 : 윈도우 환경과 가장 유사하므로 초보자에게 가장 익숙한 설치방법</p>
 <p>리눅스를 설치할 디스크 선택 파티션 분할이 가능하나 차후에 설정도 가능</p>	 <p>KDUMP 기능의 경우 비활성화 상태로 설치</p>





네트워크 & 호스트 이름의 이더넷 항목 우측의 버튼을 클릭하여 ‘컴’ 으로 변경 (설정 버튼 클릭 시 수동 IP 주소 설정도 가능하다. 차후 설정도 가능)



리눅스 보안 정책 적용 메뉴로 매우 중요한 항목이나 처음 설치 및 연습 시 ‘Apply security policy’ 항목 우측 버튼을 클릭하여 ‘끔’ 으로 변경하도록 한다.



모든 항목이 선택된 후의 화면이며 [설치 시작] 버튼을 클릭하면 설치가 시작됨



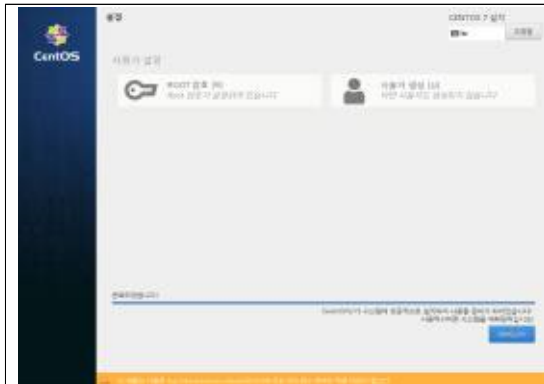
ROOT 암호 : 관리자 암호 설정  
사용자 생성 : 사용자 계정 생성



관리자 계정(root)에 대한 암호 설정 화면  
완료 버튼을 두 번 클릭해야 완료됨



사용자 계정을 생성할 수 있음  
필수 항목은 아니므로 생략 (생성 시 권한 문제 발생)



재부팅 버튼 클릭 시 설치 완료됨



설치 완료 후 부팅 화면

설치 완료 후 리눅스 설치 마무리 단계(초기 설정)








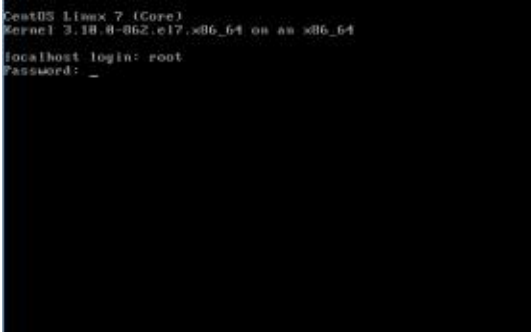


LICENSE INFORMATION 진입 후  
라이센스 동의 체크 하고 [완료] 클릭  
→ [설정 완료] 클릭



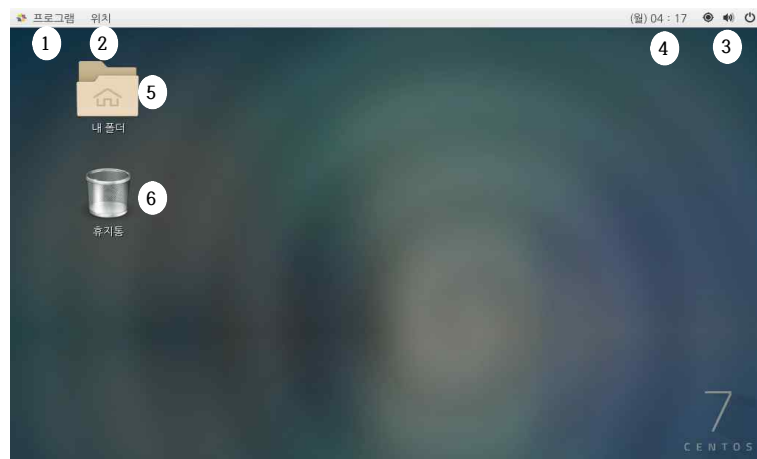
언어 : 한국어



 <p>일반 사용자 등록 (필수사항 - 보안 문제) (절대 root 등록 금지. root 계정이 일반 사용자가 됨)</p>	 <p>root 계정과 별개의 패스워드 설정</p>
 <p>설치 마무리 완료</p>	 <p>기본 사용법 설명 페이지</p>
 <p>로그인 화면 (root 계정은 '목록에 없습니까?' 클릭)</p>	 <p>로그인 아이디 클릭 시 패스워드 입력 가능</p>
 <p>리눅스 GUI 모드(X Window) 초기 화면</p>	 <p>※ 최소 설치 시 CUI 모드 로그인 화면</p>

### (3) Linux 기본 사용법 및 추가 설정

## 1) 기본 화면 구성 및 메뉴



## < CentOS 7 GUI 모드 기본 화면 >



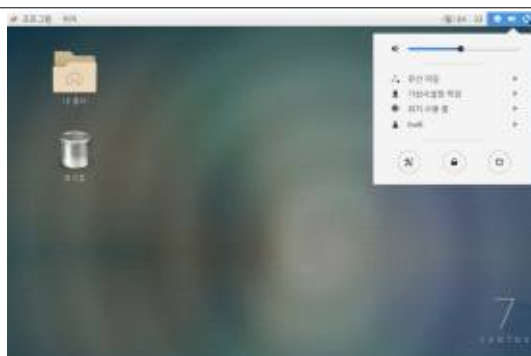
< 1. 프로그램 메뉴 >

- Windows 의 시작메뉴 기능과 동일
- 각종 프로그램을 실행할 수 있는 메뉴



< 2. 위치 메뉴 >

- Windows 탐색기 좌측 트리의 바로 가기 기능처럼 특정 디렉토리(폴더)에 바로 접근 가능한 메뉴






### < 3. 제어 메뉴 >


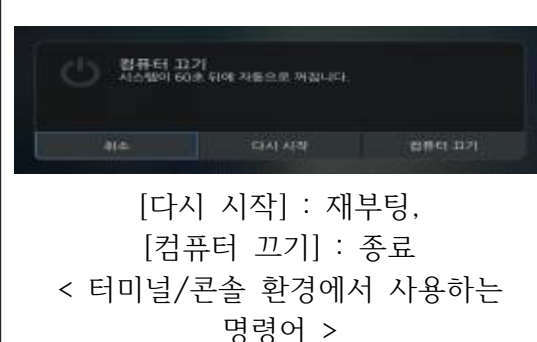
- 볼륨 조정, 네트워크 설정, 로그아웃
- 제어판, 화면 잠금, 전원


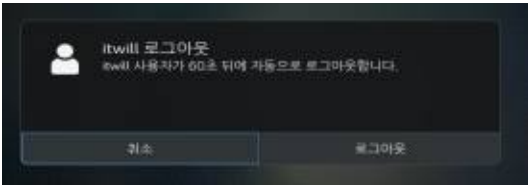
< 4. 시계 및 달력, 5. 내 폴더,  
6. 휴지통 >  
Windows 의 기능과 거의 동일함

## 2) 터미널 - 윈도우의 명령 프롬프트처럼 CUI(TUI) 모드에서 명령을 내릴 수 있는 화면

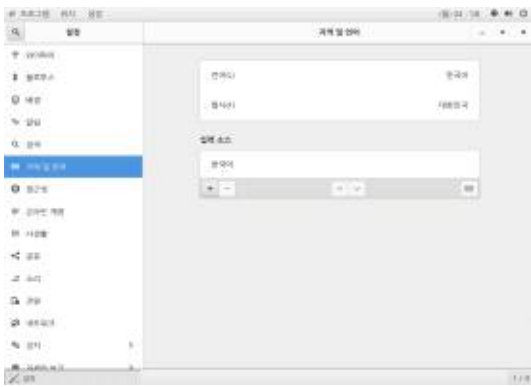



	
<p>프로그램 - 즐겨찾기 - 터미널 또는 프로그램 - 시스템 도구 - 터미널 선택하여 실행</p>	<p>바탕화면에서 마우스 우클릭 - 터미널 열기 선택하여 실행 가능</p>
 <p>터미널 실행 화면</p>	<p>&lt; [ itwill@localhost ~ ] \$ 프롬프트 의미 &gt; [ 사용자@호스트명 ~ ] \$</p> <ul style="list-style-type: none"> <li>- itwill : 현재 로그인 된 사용자</li> <li>- localhost : 컴퓨터 이름. 미 설정 시 기본 값이 localhost 로 표시됨</li> <li>- ~ : 현재 사용자의 Home 디렉터리를 의미</li> <li>- \$ : 셸 표시 기호. 일반 사용자 로그인일 때</li> </ul> <p>(관리자 계정으로 로그인 할 경우 #표시됨)</p> <ul style="list-style-type: none"> <li>- 로컬호스트 컴퓨터에 있는 itwill 계정이 접속하였으며, 현재 사용자의 홈 디렉토리에 위치해 있다.</li> </ul>


## 3) 리눅스 종료 및 재시작, 로그아웃

	
<p>우측 상단 아이콘 - 전원 버튼 클릭</p>	<p>[다시 시작] : 재부팅, [컴퓨터 끄기] : 종료</p> <p>&lt; 터미널/콘솔 환경에서 사용하는 명령어 &gt;</p> <ul style="list-style-type: none"> <li>- 종료 : poweroff, shutdown -P now, halt -p, init 0</li> <li>- 재시작 : reboot, shutdown -r now, init 6</li> </ul>

 <p>우측 상단 아이콘 - 계정명 클릭 - 로그아웃 클릭</p>	 <p>현재 사용자의 시스템 접속을 끝내기만 하고, 시스템을 종료하지는 않음 &lt; 터미널/콘솔 환경에서 사용하는 명령어 &gt; logout, exit</p>
---	--

#### 4) 한글 입력을 위한 한글 입력 소스 추가

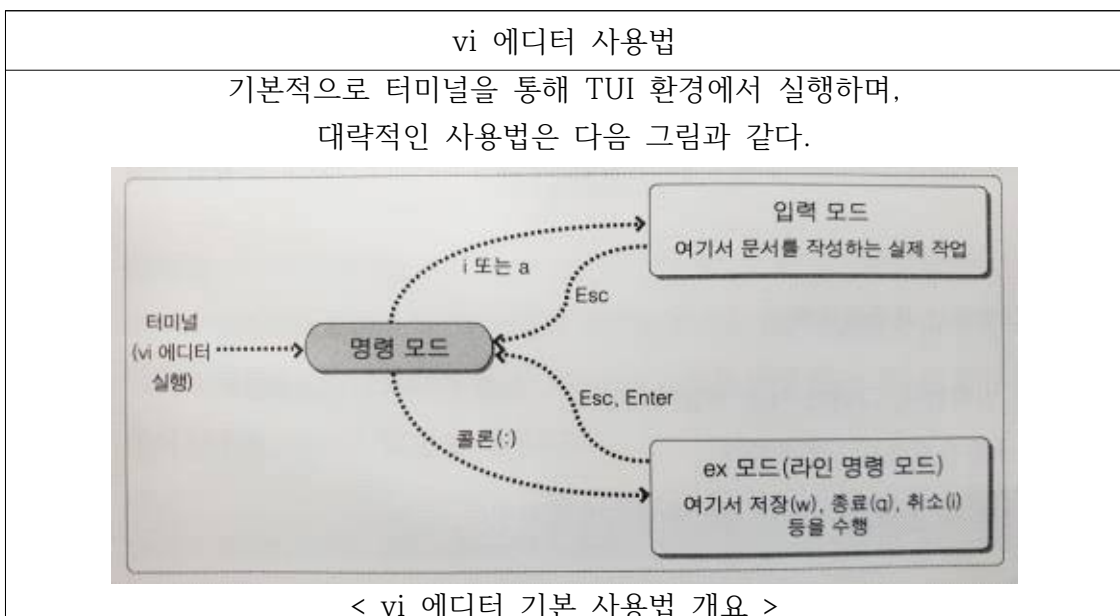
 <p>우측 상단 메뉴 - 도구(제어판) - 지역 및 언어</p>	 <p>+ 버튼 클릭 - 한국어 - 한국어(Hangul) 선택 - 추가</p>
 <p>기존 한국어 클릭 후 - 버튼 클릭하여 삭제</p>	 <p>동일한 방식으로 영어(미국) 추가</p>

	<p>우측 하단 톱니모양 아이콘 클릭 Hangul toggle key 항목에서 단축키 변경</p> <p>[ 한글 ]</p> <ol style="list-style-type: none"> <li>1. 한/영 키 사용하여 전환하려면 Hangul 만 남김</li> </ol> <p>[ 한자 ]</p> <ol style="list-style-type: none"> <li>2. 한자 키 사용하여 전환하려면 Hangul_Hanja 만 남김</li> </ol>
---	---

##### 5) 에디터(Editor)

- 기본적인 사용법을 익히지 않으면 리눅스 운영이 매우 힘들어진다.
- 리눅스에서 제공하는 기본 에디터는 두 가지(gedit, vi)이다.

gedit	vi 에디터
<ul style="list-style-type: none"> <li>- GUI 환경에서 사용하는 에디터</li> <li>- 윈도우 메모장과 유사</li> <li>- 프로그램 → 보조프로그램 → 텍스트 편집기</li> </ul>	<ul style="list-style-type: none"> <li>- TUI 환경에서 사용하는 에디터</li> <li>- 가장 많이 사용하는 에디터이며, '비' 또는 '브이아이' 로 읽음</li> <li>- Windows 사용자가 초반에 가장 어려움</li> <li>- 편한 에디터는 아니나, 리소스를 적게 차지하며 다른 운영체제에서도 사용됨</li> </ul>





vi 에디터 실행 : vi [경로명]/[파일명]

(\* 실제 운영 시 최악의 상황에 대비해 한글 경로, 한글 파일명 등은 사용하지 않는다)

```
root@localhost:~#
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@localhost ~# ls
anaconda-ks.cfg  initial-setup-ks.cfg
root@localhost ~# vi anaconda-ks.cfg
```

< vi 에디터 실행 방법 >

```
root@localhost:~#
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
#version=DEVEL
# System authorization information
auth --enableshadow --passalgo=sha512
# Use CDROM installation media
cdrom
# Use graphical install
graphical
# Run the Setup Agent on first boot
firstboot --enable
ignoredisk --only-use=sda
# Keyboard layouts
keyboard --vckeymap=kr --xlayouts='kr'
# System language
lang ko_KR.UTF-8

# Network information
network --bootproto=dhcp --device=enp0s3 --onboot=off --ipv6=auto --activate
network --hostname=localhost.localdomain

# Root password
rootpw --iscrypted $6$Kt12gy1UB5tn00Q0$X1/29oFfJ5ErEXeClaobaBxtmxsnXW4xGf4Q2hcGc
Jf/XAZVrDs2BEqnFbZ6oJGswB6xv3f8AfPffkyzsK70l0
# System services
"anaconda-ks.cfg" 65L 1542C
```

< vi 에디터 실행 화면 >

맨 아래 파일명과 현재 커서 위치가 표시되거나,  
아무것도 표시되지 않으면 명령모드이다.

< 입력 모드로 전환(명령 모드에서 사용) >

키	설명	키	설명
i	현재 커서의 위치부터 입력(주로 사용)	I	현재 커서 줄의 맨 앞에서부터 입력
a	현재 커서의 위치 다음 칸부터 입력(주로 사용)	A	현재 커서 줄의 맨 마지막부터 입력
o	현재 커서의 다음 줄에 입력	O	현재 커서의 이전 줄에 입력
s	현재 커서 위치의 한 글자를 지우고 입력	S	현재 커서의 한 줄을 지우고 입력

- 입력 모드로 전환 시 편집 화면 좌측 하단부가 INSERT 로 표시됨  
(※ 입력모드에서 esc 키를 누르면 명령모드로 전환된다.)

```
# Root password
rootpw --iscrypted $6$Kt1
Jf/XAZVrDs2BEqnFbZ6oJGswB
# System services
-- INSERT --
```

< 입력 모드 >



## &lt; 저장 및 종료 &gt;

- 저장 및 종료를 위해서는 명령 모드에서 라인 명령 모드(실행 모드, ex 모드)로 전환해야 한다. (명령 모드에서 콜론(:) 입력 시 실행 모드로 전환됨)

```
# Root password
rootpw --iscrypted $6$Kt1
Jf/XAZVrDs2BEqnFbZ6oJGsWb
# System services
:
```

## &lt; 실행 모드 &gt;

명령어	설명
:w	현재 작업 내용을 저장
:wq	현재 작업 내용을 저장하고 종료
:q	종료(작업 내용을 저장하지 않으면 종료 안 됨)
:q!	현재 작업 내용을 저장하지 않고 종료

## &lt; 저장 및 종료 명령어(실행 모드에서 사용) &gt;

## &lt; 불러오기 &gt;

- 실행모드에서 :e [불러올 파일명] 을 입력
- 파일명에 /로 시작하는 절대경로 입력 시 현재 디렉터리 외에 다른 위치의 파일도 불러올 수 있다.
- 만약, 불러올 파일명을 기존에 있는 파일명이 아닌 새로운 파일명을 입력하면 새 문서가 만들어지면서 커서가 깜빡인다. 이 후 w 명령어를 통해 저장하면 새 파일을 만들 수 있다.

## &lt; 편집(명령 모드에서 사용) &gt;

키	설명	키	설명
x	현재 커서가 위치한 글자 삭제	X	현재 커서가 위치한 앞 글자 삭제
dd	현재 커서의 행 삭제	숫자dd	현재 커서부터 숫자만큼의 행 삭제
yy	현재 커서가 있는 행을 복사	숫자yy	현재 커서부터 숫자만큼의 행을 복사
p	복사한 내용을 행 이후에 붙여 넣기	P	복사한 내용을 현재 행 이전에 붙여넣기

## &lt; 찾기(명령 모드에서 사용) &gt;

- 명령모드에서 '/' 기호 뒤에 찾을 단어를 입력한다.
- 계속 찾으려면 'n' 키를 입력한다.

## &lt; 되돌리기(Undo) &gt;

- 작업을 할 때 오타나 잘못된 입력에 의해서 앞으로 되돌아가고 싶을 때 사용한다.
- 단축키 : 'u' (명령모드에서 사용)
- 작업한 순서의 역순으로 Undo가 작동된다.
- 저장된 작업에 대한 취소가 모두 완료 되면 "Already at oldest change" 라는 메시지가 나온다.

## &lt; 재실행(Redo) &gt;

- 현재 작업의 앞 시점으로 갈 때 실행(Undo 작업을 되돌릴 때)
- 단축키 : Ctrl + r (Control Redo 라고 기억하면 편하다)
- 가장 앞 시점으로 이동하면 "Already at newest change" 라는 메시지가 나온다.

#### < 멀티 에디터 >

- 편집 화면 분할(Split) 기능으로써, 두 파일을 비교하며 수정해야 할 때 사용한다.  
(개발자에게 매우 유용한 기능)
- 화면 분할 : Ctrl + w -> n (창 분할 시에도 맨 아래 상태표시줄은 공유함)
- 파일 불러오기 : 불러올 창에서 :e [불러올파일명]
- 창 사이의 이동 : Ctrl + ww
- 편집 완료 후 종료는 일반 종료와 동일함

#### < 유용한 팁 >

- vi 에디터에서 셸 명령어(터미널에서 사용하는 명령어)를 사용해야 할 경우가 있다.  
(ex. 파일목록 조회 등)  
이럴 때, 실행모드에서 :! [실행할 명령어] 입력 시 셸 명령어 실행됨  
ex) :! ls 명령어 입력 시 에디터에서 ls 결과 출력됨. 엔터키 누르면 다시 에디터 화면으로 복귀.
- vi 에디터 사용 시 라인 번호를 출력해야 할 경우 :set number(줄여서 :set nu)  
입력 시 라인번호가 표시됨  
해제는 :set nonumber(줄여서 :set nonu) 입력

### 6) 리눅스 기본 명령어

#### < ls >

- list 의 줄임말로, 파일 목록을 조회하는 명령어(리눅스 운영체제에서 가장 많이 쓰이는 명령어 중 하나)
- 사용법 : ls [파라메타] [디렉터리명] 또는 ls [디렉터리명] [파라메타]
  - 1) ls : 뒤에 나오는 디렉터리 명을 비워두면 현재 디렉터리의 파일 목록을 보여준다.
  - 2) ls /etc/sysconfig : /etc/sysconfig 디렉터리의 목록을 보여준다.
  - 3) 파라메타 사용법 : '-'기호 + 알파벳의 형태를 옵션 또는 파라메타라고 하며 명령어의 부가적인 기능을 더할 수 있다.  
→ 대표적인 파라메타
    - a : 현재 디렉터리의 숨김 파일을 포함해서 보여준다.
    - l : 현재 디렉터리의 목록을 자세히 보여준다. (줄임 명령어 ll 로 대체 가능)
    - al : -a 옵션과 -l 옵션을 조합한 결과를 보여준다.

#### < cd >

- change directory의 줄임말로 디렉터를 이동할 때 사용한다.
- 사용법 : cd [디렉터리명]
  - 1) cd . : 현재 디렉터리로 이동
  - 2) cd .. : 상위 디렉터리로 이동
  - 3) cd /home : 홈 디렉터리로 이동  
(cd home 과 cd /home 은 이동되는 곳이 다르다!)
  - 4) cd : 접속한 사용자의 루트 디렉터리로 이동

<p style="text-align: center;">&lt; pwd &gt;</p> <ul style="list-style-type: none"> <li>- Print Working Directory의 줄임말로 현재 디렉터리의 전체 경로(절대 경로)를 화면에 보여준다.</li> <li>- 사용법 : pwd</li> </ul>
<p style="text-align: center;">&lt; cp &gt;</p> <ul style="list-style-type: none"> <li>- copy 의 줄임말로 파일이나 디렉터리를 복사</li> <li>- 사용법 : cp [원본 파일] [새로 복사될 파일]</li> <li>- 새롭게 복사될 파일이 이미 존재한다면 기존 파일을 덮어 쓰고 만약 없다면 새롭게 생성 된다.</li> <li>- 새로 복사한 파일은 복사한 사용자의 소유가 된다.</li> <li>- 만약, [새로 복사될 파일]의 형식을 절대 경로로 적으면 그 경로에 복사가 된다.</li> </ul>
<p style="text-align: center;">&lt; rm &gt;</p> <ul style="list-style-type: none"> <li>- remove 의 줄임말로 파일이나 디렉터리를 삭제</li> <li>- 사용법 : rm [지우고자 하는 파일 또는 디렉터리명]</li> <li>- 파일 삭제는 삭제를 할 수 있는 권한의 계정만 삭제를 진행할 수 있다.</li> <li>- 또한 터미널 환경에서는 휴지통 개념이 없으므로 삭제한 파일이나 폴더를 복구하기가 어렵다. (GUI 환경에서는 ‘휴지통에 버리기’ 메뉴가 있으므로 차후에 복구가 가능하다)</li> </ul>
<p style="text-align: center;">&lt; mv &gt;</p> <ul style="list-style-type: none"> <li>- move 의 줄임말로 파일이나 디렉터리의 이름을 변경하거나 다른 디렉터리로 옮길 때 사용</li> <li>- 사용법 : mv [원본파일] [새파일]</li> <li>- 기본적으로 Windows 에서의 잘라내기(Ctrl + X) + 붙여넣기(Ctrl + V) 기능과 동일</li> <li>- 원본파일의 위치와 새 파일의 위치가 같다면 파일의 이름을 변경할 수 있다.</li> </ul>
<p style="text-align: center;">&lt; mkdir &gt;</p> <ul style="list-style-type: none"> <li>- make directory 의 줄임말로 디렉터리(폴더)를 생성할 때 사용</li> <li>- 사용법 : mkdir [디렉터리명]</li> <li>만약, 현재 디렉터리 아래가 아닌 다른 디렉터리에 생성하려면 ‘/’가 붙은 절대경로 사용</li> <li>- -p 옵션을 사용할 경우 경로 상에 디렉토리가 없다면 상위 디렉터리까지 생성할 수 있다. ex) mkdir -p /d/e</li> </ul>
<p style="text-align: center;">&lt; touch &gt;</p> <ul style="list-style-type: none"> <li>- 빈 파일(파일의 크기가 0byte) 을 생성한다.</li> <li>- 사용법 : touch [파일명]</li> <li>- 만약 파일이 이미 존재한다면 파일의 최종 수정 시간을 변경하게 된다.</li> </ul>

<p style="text-align: center;">&lt; cat &gt;</p> <ul style="list-style-type: none"> <li>- conCATenate(연쇄된, 이어진) 의 줄임말로 파일의 내용을 화면에 출력해주는 명령어이다.</li> <li>- 사용법 : cat [파일명]</li> <li>- 만약, cat 명령어 뒤에 파일명을 여러 개 나열하면 파일의 내용을 연결해서 보여준다.</li> </ul>
<p style="text-align: center;">&lt; more &amp; less &gt;</p> <ul style="list-style-type: none"> <li>- 한 화면에 표시되지 않는 대량의 내용을 페이지 단위로 보여주기 위해 사용하는 옵션</li> <li>- 특정 명령어 뒤에 ' ' 기호와 함께 조합하여</li> <li>- 사용법 : [명령어]   more 또는 [명령어]   less</li> <li>ex) ls -al   more : 모든 숨김파일을 포함한 자세한 목록을 페이지 단위로 표시</li> <li>- more : 엔터키를 사용하여 한 줄씩 스크롤, 스페이스바를 사용하여 한 페이지씩 스크롤 가능</li> <li>less : more 옵션에서 사용하는 명령어는 물론 방향키를 사용할 수도 있다.</li> </ul>

#### 7) 리눅스 사용자 관리 명령어

- 리눅스는 다중 사용자 시스템(Multi-user System)으로써, 1대의 리눅스 서버에 여러 명의 사용자가 동시에 접속해서 사용한다.
- 여러 사용자들 중 리눅스를 관리하는 사용자를 관리자라고 하며, 기본 관리자 계정을 root 라고 한다.
- 사용자 관리에는 유저 생성, 유저 삭제 이외에도 유저 속성 변경, 패스워드 지정 등 여러 가지가 있다.
- 사용자 관리 명령어를 수행하는 계정은 반드시 관리자 계정이어야 한다.

<p style="text-align: center;">&lt; useradd &gt;</p> <ul style="list-style-type: none"> <li>- 새로운 사용자를 추가할 때 사용하는 명령어</li> <li>- 사용법 : useradd [새로운 사용자명]</li> <li>ex) useradd itwillbs : itwillbs 사용자를 추가한다.</li> <li>- 새로운 사용자를 추가하면 /home 디렉터리 아래에 사용자 명의 디렉터리가 생성된다.</li> <li>- '/home/새로운사용자명'이 새 사용자의 홈디렉터리이다.</li> </ul>
<p style="text-align: center;">&lt; passwd &gt;</p> <ul style="list-style-type: none"> <li>- 특정 사용자의 패스워드를 설정(변경)하는 명령어</li> <li>- 사용법 : passwd [사용자명]</li> <li>ex) passwd itwillbs : itwillbs 사용자의 비밀 번호 변경한다.</li> <li>itwillbs 사용자의 비밀 번호 변경 중</li> <li>새 암호 :</li> <li>새 암호 재입력 :</li> </ul>
<p style="text-align: center;">&lt; userdel &gt;</p> <ul style="list-style-type: none"> <li>- 생성된 사용자를 삭제하는 명령어</li> <li>- 사용법 : userdel [삭제할 사용자]</li> </ul>

## 8) 네트워크 설정

### ① 리눅스에서 네트워크 장치

- Windows 환경과 동일하게 랜카드를 인식하고 설정이 가능하다.
- 만약, 가상 머신을 사용하는 경우에도 실제 컴퓨터와 마찬가지로 가상의 랜카드를 갖게 되며, 자동으로 장치의 이름을 정하는데 보통 en 으로 시작하는 숫자와 문자의 조합으로 인식을 한다.
- ifconfig 명령어 또는 ip add show 명령어를 사용하여 네트워크 정보를 조회한다.

※ 기존 CentOS 7 버전 이전까지의 사용자들은 ifconfig 명령어가 더 친숙할 것이다. 그러나 앞으로 ifconfig 명령어를 지원하지 않게 될 수 있으므로 ip add show 명령어도 같이 기억해 두자.

### ② 네트워크 관련 설정

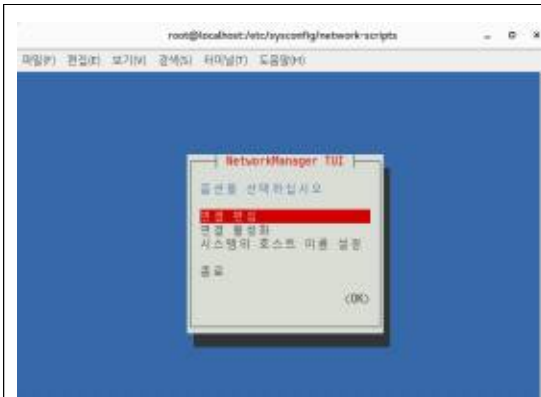
- CentOS 6 버전까지는 vi 편집기를 이용, TUI방식으로 설정을 했다.
- 랜카드 설정 경로는 /etc/sysconfig/network-scripts/ifcfg-enp0s3 이다.  
ex) vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
- 그러나, CentOS7 버전부터는 nmtui 명령어를 사용하여 보다 편리하게 네트워크 관련 설정을 할 수 있다.

#### < nmtui >

- Network Manager Text User Interface의 약자
- 텍스트 에디터를 사용하지 않고 좀 더 보기 편하게 네트워크 설정이 가능함.
- 로컬 컴퓨터에서는 한글로 표기가 되며, 만약 원격 실행 시 영어로 나오게 된다.
- nmtui 명령어를 통해 다음 항목을 변경할 수 있다.

- 1) 자동 IP 주소 또는 고정 IP 주소 사용 여부 결정
- 2) IP주소, 서브넷 마스크, 게이트웨이 정보, DNS 정보
- 4) 네트워크 카드 드라이버 설정
- 5) 네트워크 장치(enp0s3)의 설정

※ 터미널에서 gnome-control-center network 명령어를 입력하거나, 우측 상단의 설정 - 네트워크 메뉴를 클릭하면 GUI 환경에서의 네트워크 설정이 가능하다. 그러나, 텍스트 모드에서는 이 방법을 사용할 수 없으므로 되도록이면 nmtui 사용법을 익히도록 하는 것이 좋다.



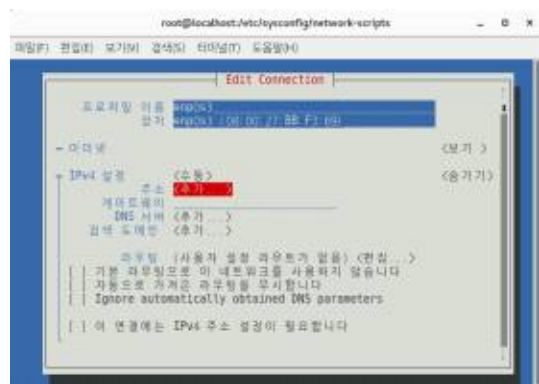
터미널에서 nmtui 실행 화면



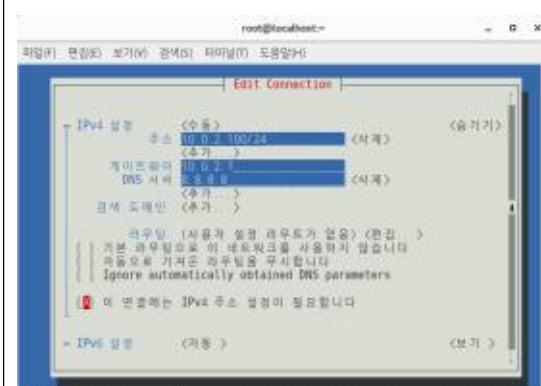
이더넷 - 네트워크 장치 선택 후 엔터



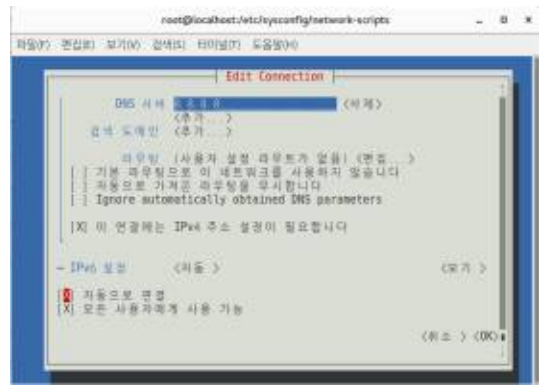
IPv4 설정 우측 <자동> 클릭 시  
DHCP 사용  
<수동> 클릭 시 IPv4 주소 직접 설정 가능



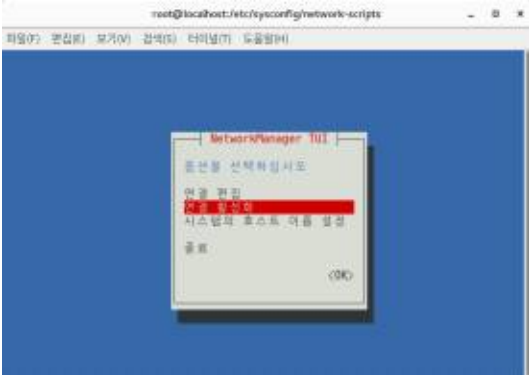


각 항목의 <추가...> 버튼 클릭하여  
주소 정보 입력  
(주소, 게이트웨이, DNS 서버는 필수로 입력)



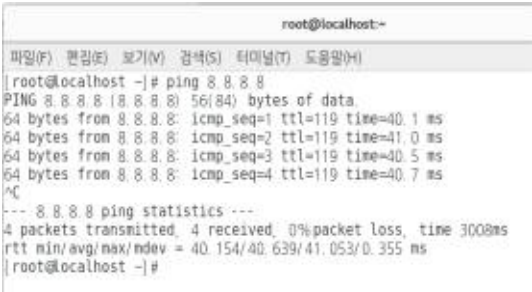

[이 연결에는 IPv4 주소...] 체크(스페이스바 사용)



[자동으로 연결] 체크 후 [OK] 선택 엔터

 <p>[연결 활성화] 에서 활성화/비활성화 변경 가능</p>	 <p>랜카드 선택 시 비활성화 표시되면 현재 활성화 상태</p>
	<p>종료 후 설정 내역 확인 시 vi 에디터 사용</p> <p>vi</p> <p>/etc/sysconfig/network-scripts/ifcfg-enp0s3</p> <p>BOOTPROTO=none : DHCP / none 선택</p> <p>IPADDR=10.0.2.100 : 자신의 IP 주소</p> <p>PREFIX=24 : 자신의 서브넷마스크 길이 (= NETMASK=255.255.255.0 과 동일)</p> <p>GATEWAY=10.0.2.1 : 게이트웨이 주소</p> <p>DNS1=8.8.8.8 : DNS 서버 주소</p> <p>: wq 명령어로 빠져나온 뒤 네트워크 재시작 명령 실행</p> <p>systemctl restart network</p>



 <pre> root@localhost:~# ping 8.8.8.8 PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data: 64 bytes from 8.8.8.8: icmp_seq=1 ttl=119 time=40.1 ms 64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=41.0 ms 64 bytes from 8.8.8.8: icmp_seq=3 ttl=119 time=40.5 ms 64 bytes from 8.8.8.8: icmp_seq=4 ttl=119 time=40.7 ms ^C --- 8.8.8.8 ping statistics --- 4 packets transmitted, 4 received, 0% packet loss, time 3008ms rtt min/avg/max/mdev = 40.154/40.639/41.053/0.355 ms root@localhost:~# </pre> <p>ping 명령어를 사용하여 통신 확인          ※ ping 대신 prettytyping 프로그램을 사용해도 된다.</p> <ul style="list-style-type: none"> <li>- prettytyping 은 ping에 비해 그래픽적인 요소를 통해 보다 직관적으로 연결 상태를 확인하기가 쉽다.</li> <li>- yum install -y prettytyping 명령어를 사용하여 설치</li> </ul>	 <p>Firefox 웹 브라우저를 사용하여 웹 접속 확인 가능</p>
<p style="text-align: center;">&lt; 네트워크 보안을 위한 SELinux &gt;</p> <ul style="list-style-type: none"> <li>- Security Enhanced Linux 의 약자로 보안에 취약한 리눅스를 보호하기 위해 탄생 됨.</li> <li>- 시스템에서 보안에 영향을 미치는 서비스, 권한 등을 제어할 수 있다.</li> <li>- SELinux 환경에서는 해커가 어떤 경로로 시스템 침입에 성공하든, 침입한 경로의 애플리케이션(프로그램) 사용 이상의 권한을 얻지 못한다. 가령, FTP 서버의 경로로 리눅스에 침입한다면 FTP와 관련된 디렉터리나 파일 외에는 다른 서버에 접근할 수 없어 해킹으로 인한 피해를 FTP 서버만으로 제한할 수 있다.</li> <li>- SELinux의 사용여부는 강제(Enforcing), 허용(Permissive), 비활성(Disabled)의 3레벨을 지원하며 기본값은 강제이다.</li> <li>- 강제는 시스템 보안에 영향을 미치는 기능이 감지되면, 그 기능이 작동되지 않도록 시스템에서 막아주는 반면, 허용의 경우 해당 기능을 허용은 하되 사용 내용이 로그(기록)에 남고, 화면상에도 출력된다.</li> <li>- 비활성 : SELinux를 사용하지 않도록 하므로 보안상 매우 취약해진다.</li> <li>- SELinux를 설정하기 위해서는 /etc/sysconfig/selinux 파일을 수정한다.              ex) echo "SELINUX=disabled" 입력 시 SELinux 비활성화</li> </ul>	

### 9) 리눅스 프로그램 설치

- 초창기의 리눅스는 새로운 프로그램을 설치하는 것이 어려워 초보자가 프로그램을 하나 설치하는 것도 힘든 일이었다. 따라서, 이와 같은 문제를 해결하기 위해 레드햇(RedHat)사에서 Windows의 setup.exe와 비슷한 프로그램 설치용 파일을 만들게 되었는데 확장자가 .rpm 이며, 이를 패키지(package)라고 한다. 이렇게 .rpm 확장자로 된 패키지를 설치하는 것을 RPM(RedHat Package Manager) 설치라고 한다.

< Source 설치와 RPM 설치의 비교 >	
<p>&lt; Source 설치 &gt;</p> <ul style="list-style-type: none"> <li>- 수동으로 설치</li> <li>- 수동설치 환경은 “.tar.gz” 또는 “.tgz” 확장자 형태가 많다.</li> <li>- 관리자는 원하는 위치에 설치가 가능</li> <li>- 패키지 매니저를 사용하지 않으므로 설치가 어렵다.</li> <li>- 형식 패키지이름-버전-릴리스번호. (아키텍처).tar.gz</li> </ul>	<p>&lt; RPM(Redhat Package Manager) &gt;</p> <ul style="list-style-type: none"> <li>- 수동이나 매니저를 사용하여 설치(반수동)</li> <li>- 자동 설치 환경은 “.rpm” 확장자로 끝난다.</li> <li>- RPM 설치를 하면 지정된 위치에 설치가 된다. (위치 변경 불가능!)</li> <li>- 소스 설치에 비해서 상대적으로 설치가 쉽다.</li> <li>- 형식 패키지명-버전-릴리스번호.CentOS버전. 아키텍처.rpm</li> </ul>
<p>&lt; RPM 명령어 &gt;</p> <ol style="list-style-type: none"> <li>설치 <ul style="list-style-type: none"> <li>- RPM 설치는 우선 해당 프로그램(패키지)의 .rpm파일을 다운로드 받는 것이다.</li> <li>- 다운로드 된 .rpm 확장자 파일이 있는 곳으로 이동한다.</li> <li>- 사용법 : rpm -Uvh 패키지이름.rpm</li> <li>- 옵션 U : 설치가 되어 있다면? 업그레이드, 미설치라면? 일반적인 설치 (대문자 필수) v : 설치과정 확인 h : 설치 과정을 # 기호로 화면에 출력</li> </ul> </li> <li>삭제 <ul style="list-style-type: none"> <li>- 사용법 : rpm -e 패키지이름 (erase)</li> <li>- 삭제 시에는 .rpm까지 다 적는 것이 아니라 프로그램명까지만 적으면 된다.</li> </ul> </li> <li>이미 설치된 패키지 조회 <ul style="list-style-type: none"> <li>- 시스템에 패키지가 설치되어있는지 확인</li> <li>- 사용법 : rpm -qa 패키지이름 (특정 패키지이름을 쓰지 않으면 설치된 모든 패키지의 목록을 보여준다.)</li> </ul> </li> <li>아직 설치되지 않은 rpm 파일 조회 <ul style="list-style-type: none"> <li>- rpm -qip 패키지이름.rpm</li> <li>- 어떤 기능 설치 전 rpm 파일 안에 해당 기능이 포함되었는지 미리 확인하는데 유용하게 사용될 수 있다.</li> </ul> </li> </ol>	

#### < RPM 의 단점 >

- 사용자의 편의성을 위해 만든 RPM은 몇 가지 단점을 가진다.
- 우선 설치가 단계별로 진행되기 때문에 특정 단계를 넘어가지 못한다면 설치를 완료할 수가 없다.
- 또한, 원하는 프로그램을 설치하고 실행을 하려고 하면 어떤 특정한 프로그램이 설치가 되지 않아 진행을 할 수 없다고 메시지를 보여준다. 그 이유는 프로그램을 설치하고 구동하는데 있어 해당 프로그램만이 아닌 다른 요소들의 설치도 필요하기 때문이다. (이를 파일의 “의존성” 이라고 한다.)
- ※ 메모를 하려고 종이를 샀는데 정작 펜을 구입하지 않은 상황에 비유할 수 있겠다!
- ※ 좀 더 리눅스에 빗대어보면 리눅스에서 기본적으로 사용하는 웹 브라우저인 Firefox를 X 윈도우가 설치되지 않은 상태에서 설치를 하려고 한다면 웹브라우저인 Firefox는 화면상에 Graphic 적으로 보여주는 X 윈도우에 대한 의존성이 있으므로 설치가 되지 않을 것이다.
- 이런 RPM의 단점을 보완한 개념이 바로 ‘YUM’ 이다.
- YUM 설치는 어떤 프로그램을 설치하고자 할 때 사용자가 미처 챙기지 못한 외적인 부분까지 함께 설치를 해주며 이로 인해 “의존성을 확보했다”고 한다.

#### < YUM >

- Yellow dog Updater, Modified 의 약자이며, 명령어의 이름이다.
- ‘yum’ 명령어는 ‘rpm’ 명령어의 패키지 의존성 문제를 완전하게 해결해준다.
- 특정 패키지를 설치하고자 할 때 필요한 다른 패키지도 같이 설치됨으로써 의존성을 해결한다.
- RPM 설치는 .rpm 파일이 DVD에 있거나 인터넷에서 미리 다운로드한 후 설치해야 하나, YUM 설치는 CentOS 프로젝트가 제공하는 **rpm 파일 저장소(Repository)** 에서 rpm 파일과 의존성이 있는 다른 rpm 파일까지 인터넷을 통해 모두 알아서 다운로드한 후 자동으로 설치된다.
- ※ 사용자는 더 이상 rpm 패키지를 설치할 때 의존성 문제를 신경 쓰지 않아도 된다!

#### < YUM Repository(저장소) >

- YUM 명령어를 이해하는데 있어 저장소(Repository)의 개념이 중요하다.
- 저장소는 로컬 컴퓨터가 아닌 외부에 존재한다.
- 저장소(Repository)의 URL은 /etc/yum.repos.d/ 디렉터리의 파일에 저장되어 있다.

## < YUM 기본 사용법 >

### 1. 기본 설치

- 사용법 : yum -y install [패키지이름]  
ex) yum -y install java-1.8.0-openjdk 또는 yum -y install java-1.8.0-open\* (파일명 다 모를 경우)
- '-y' 옵션은 중간이나 뒤쪽에 적어도 되며 install 명령을 실행하면 설치할 것 인지를 물어보는 yes/no 선택 사항에서 무조건 yes를 한 것으로 간주되어, 자동으로 설치가 된다.
- 파일명을 완벽히 알지 못할 경우 만능문자(\*)를 사용할 수 있다. (관련 패키지 모두를 대상으로 함)

### 2. yum 명령어를 이용한 rpm 파일 설치

- 사용법 : yum localinstall [rpm파일이름.rpm]
- rpm 파일이 로컬 호스트에 있다면 yum 명령어를 통해 rpm 파일을 설치할 수 있다.
- 이 때, 'rpm -Uvh' 명령어 대신 'yum localinstall' 명령어를 사용한다.
- rpm 파일에 의존성 문제가 있을 때 해결할 수 있는 파일을 인터넷에서 다운 로드해서 설치해 준다는 점이다.

### 3. 업데이트 가능한 목록 보기

- 사용법 : yum check-update  
ex) yum check-update java\*
- 시스템에 설치된 패키지 중에서 업데이트가 가능한 패키지의 목록을 출력한다.

### 4. 업데이트

- 사용법 : yum update [패키지이름]  
ex) yum update java\*
- 기존에 설치되지 않은 패키지는 새로 설치하고, 이미 설치되어 있다면 최신으로 업데이트 한다.
- yum update 뒤에 패키지이름을 따로 명시하지 않으면 모두 업데이트 하므로 시간이 오래 걸린다.

### 5. 삭제

- 사용법 : yum remove [패키지이름]  
ex) yum remove java-1.8.0-openjdk
- 기존 설치된 패키지를 제거한다.

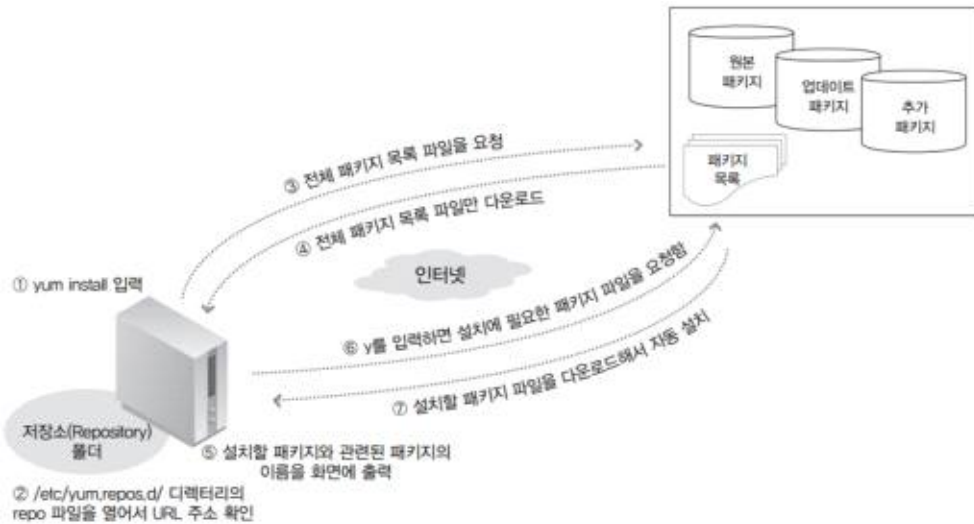
### 6. 패키지 정보 확인

- 사용법 : yum info [패키지이름]  
ex) yum info java-1.8.0-openjdk
- 패키지의 요약된 정보를 보여준다.

### 7. 패키지 목록확인

- 사용법 : yum list 패키지이름  
ex) yum list java\*

### < YUM의 작동 흐름도 >



① yum install release 명령어를 실행한다.

② /etc/yum.repos.d/ 디렉터리의 repo 파일을 열어서 URL 주소를 확인한다.

③,④ 패키지에 관련 목록을 요청하고 다운 받는다.

※ 주의할 점은 실제 패키지 파일을 다운로드 받는 건 아니고 이름만 들어있는 목록을 받아온다는 것이다.

⑤,⑥ 패키지 관련 설치 화면을 확인하고 y를 누르면 설치를 위해 파일을 요청한다.

※ 이 과정에서 '-y' 옵션을 사용하면 설치가 자동으로 진행이 된다.

⑦ 저장소에서 설치에 필요한 패키지 파일을 다운로드해서 설치가 진행된다.

### < YUM 설치 실습 >

#### 1. Repository 종류 중 하나인 epel 추가하기

```
yum install -y epel-release
```

#### 2. JDK 및 Tomcat(WAS) 설치

(JDK 설치 시 Oracle JDK 가 아닌 OpenJDK 설치)

```
# yum install -y java-1.8.0-openjdk
```

```
# yum install -y tomcat
```

위의 두 문장을 하나로 결합하여 다음과 같이 사용할 수도 있다.

```
# yum install -y java-1.8.0-openjdk && yum install -y tomcat
```

(앞 패키지 설치 작업이 종료되면 뒤의 패키지 설치 작업이 이어서 진행된다.)

[ Tomcat 설치 후 서비스 실행 ]

(1) 설정 파일이 있는 Tomcat 디렉토리로 이동 후 필요한 xml 파일 편집

```
# cd /etc/tomcat
```

```
# vi [파일명]
```

(2) Tomcat 서비스 시작 및 상태 확인

```
# systemctl start tomcat
```

```
# systemctl status tomcat
```

```
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/usr/lib/systemd/system/tomcat.service; disabled; vendor preset: disabled)
   Active: active (running) since 금 2018-10-12 12:50:54 KST; 3h 6min ago
     Main PID: 2778 (java)
       Tasks: 41
      CGroup: /system.slice/tomcat.ser
              └─2778 /usr/lib/jvm/jre/bin/java -classpath /usr/share/tomcat/bin/...
```

Active(running)  
상태 확인

(3) 서비스 디렉토리 확인(war 파일 업로드 위치)

```
# cd /usr/share/tomcat
```

```
# ll
```

```
drwxr-xr-x. 2 root root 76 10월 12 12:35 bin
lrwxrwxrwx. 1 root tomcat 11 10월 12 12:35 conf -> /etc/tomcat
lrwxrwxrwx. 1 root tomcat 22 10월 12 12:35 lib -> /usr/share/java/tomcat
lrwxrwxrwx. 1 root tomcat 15 10월 12 12:35 logs -> /var/log/tomcat
lrwxrwxrwx. 1 root tomcat 22 10월 12 12:35 temp -> /var/cache/tomcat/temp
lrwxrwxrwx. 1 root tomcat 23 10월 12 12:35 webapps -> /var/lib/tomcat/webapps
lrwxrwxrwx. 1 root tomcat 22 10월 12 12:35 work -> /var/cache/tomcat/work
```

서비스 중인  
디렉토리

(4) 접속 테스트

webapps 디렉토리에 프로젝트 디렉토리(ex. ROOT, Sample 등) 생성한 후 index.html 파일이나 index.jsp 파일 생성 및 웹브라우저로 확인

ex) Sample 프로젝트 내의 index.html 파일 확인

```
# mkdir /usr/share/tomcat/webapps/Sample
```

- 폴더 생성하는 명령어(webapps 폴더 내에 war 파일을 업로드 했을 경우 자동으로 생성됨)

```
# echo "TOMCAT" > /usr/share/tomcat/webapps/Sample/index.html
```

- 해당 디렉토리에 index.html 파일을 생성하면서 TOMCAT 텍스트 포함시켜 저장

- 웹브라우저에서 http://localhost:8080/Sample 입력 후 “TOMCAT” 텍스트 출력 확인

### 3. MySQL DB(Maria DB) 설치

- MySQL 과 Maria DB 의 차이점 : MySQL 의 소유권이 Sun Microsystems 로 부터 Oracle에 넘겨지면서 License의 불확실함을 해결하고자 만든 오픈소스 DB 가 Maria DB 이다. MySQL 5.6 = Maria DB 10.0, MySQL 5.7 = Maria DB 10.1 버전으로 매칭이 되며, Maria DB 10.2 부터는 MySQL 의 개발 속도를 앞서 나가기 시작했다.

#### - Maria DB 설치

# yum install -y mariadb\* 명령어를 사용하여 mariadb 관련 패키지를 모두 설치하거나

# yum install mariadb.x86\_64 mariadb-server.x86\_64 -q 명령어로 필수 패키지만 설치할 수도 있다.

#### [ Maria DB 설치 후 설정 및 실행 ]

- (1) Maria DB 설정(설정 파일 my.cnf 의 경우 새로 만들거나 샘플파일 복사해서 사용할 수 있음)

# cp /usr/share/mysql/my-small.cnf /etc/my.cnf

- 샘플 설정 파일 복사(덮어쓰기 여부 질문에 y 입력)

# vi /etc/my.cnf

- 설정 파일 편집

[client] 항목 : 클라이언트용 설정

- 마지막 라인에 다음 항목 추가하여 한글 인코딩 셋(Encoding set) 변경  
default-character-set=utf8

[mysqld] 항목 : 서버용 설정

- 첫 라인에 다음 항목 추가하여 한글 인코딩 셋 변경  
character-set-server=utf8
- 49번 라인 #log-bin=mysql-bin 주석 해제(# 삭제)하여 로그 기능 수행  
(롤백 가능하도록 설정)

[mysql] 항목 : 기본 설정

- 80번 라인 no-auto-rehash 주석 확인(주석처리 되어 있을 경우 주석 제거)

- (2) 서비스 실행 및 확인

# system start mariadb (mariadb 또는 mysql 또는 maria 로 대상 지정)

# system status mariadb



(3) 기본 보안 설정

1) # mysql\_secure\_installation

- 보안 설정 스크립트 실행(만약 스크립트 설정 도중 실패 시 Ctrl + C 로 취소 후 재 입력 가능)

2) Enter current password for root (enter for none) : 엔터(현재 root 계정 패스워드)

3) Change the root password? [Y/n] : y 입력 (root 계정 패스워드 변경 여부)

4) New password : 1234(root 계정 패스워드 입력)

5) Re-enter new password : 1234(패스워드 재입력)

6) Remove anonymous users? [Y/n] : y (익명 유저 삭제 여부)

7) Disallow root login remotely? [Y/n] : y (원격 접속 시 root 계정 로그인 차단 여부)

8) Remove test database and access to it? [Y/n] : y (테스트 데이터베이스 삭제 여부)

9) Reload privilege tables now? [Y/n] : y

(설정 파일을 DB에 업데이트<flush, reloading> 시킴 = my.cnf 파일을 systemdb 에 업데이트)

(4) 접속 확인

# mysql -uroot -p

Enter password : 1234

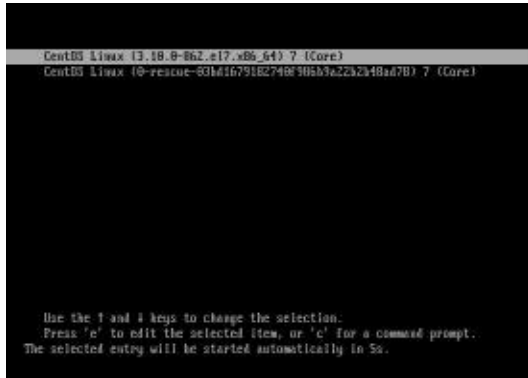
MariaDB > 프롬프트 나타나면 성공!

## 10) CentOS 7 패스워드 초기화

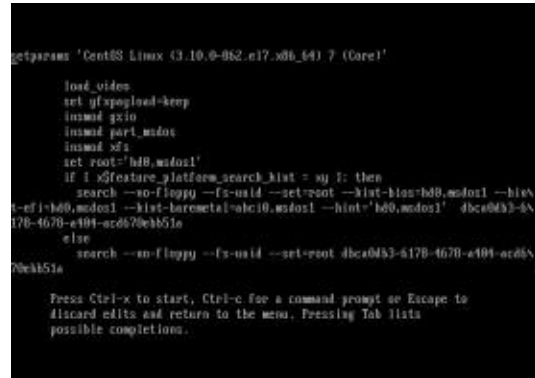
- CentOS 7에서 패스워드 복구 시 명령어를 통한 복구를 수행해야 하므로 이전 버전까지의 방법보다 어려워졌다.

### < CentOS 7 패스워드 초기화 과정 >

1. 처음 부팅 시 메뉴 선택 화면에서 아무키나 입력 후 'e' 키 누르면 부트로더 실행



< 부팅 메뉴 선택 화면 >



< e 눌렀을 때 실행되는 부트로더 화면 >

2. 부트로더 화면에서 방향키를 사용하여 맨 아래쪽으로 스크롤하여 다음 문장 찾기

```
linux16 /vmlinuz-3.10.0-862.el7.x86_64 root=/dev/mapper/centos-root ro\
rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet LANG=ko_KR.UTF-8
```

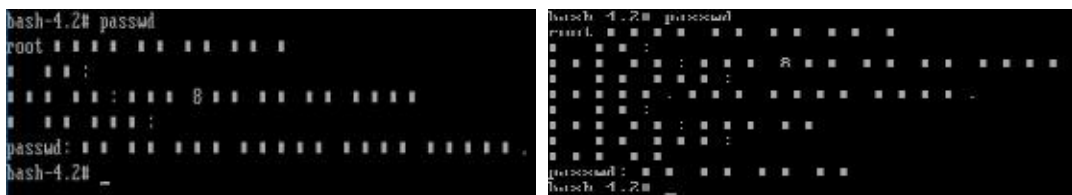
3. 위의 표시된 문장을 다음과 같이 편집 (오타 주의! 한 문장이므로 좌우 이동 필요)

- 1) 첫 번째 라인 마지막의 'ro' 를 'rw' 로 변경
- 2) 두 번째 라인 가운데 rhgb quiet 삭제 후 init=/bin/bash 문장을 추가

4. Ctrl + X 누르면 부팅 진행 후 로그인 과정 없이 다음과 같은 프롬프트가 나타남

```
[ OK ] Started Cleanup udevd DB.
[ OK ] Reached target Switch Root.
bash-4.2#
```

5. passwd 명령어 사용하여 패스워드 변경 진행(한글이 깨져 보임)



< 패스워드 변경 성공 시(메세지가 길다) > < 패스워드 변경 실패 시(메세지가 짧다) >

6. SELinux에서 카운트하는 파일을 수동으로 생성

touch /.autorelabel

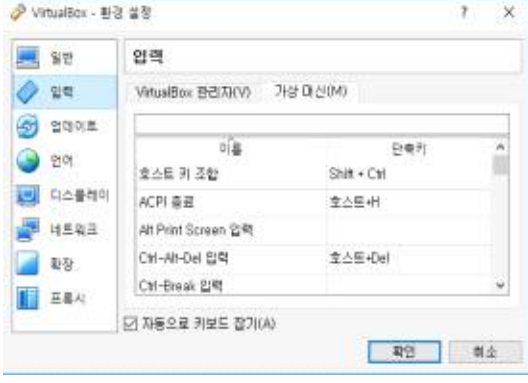
7. 재부팅 시 reboot 명령어 사용이 불가능하므로 초기화 프로세스 호출

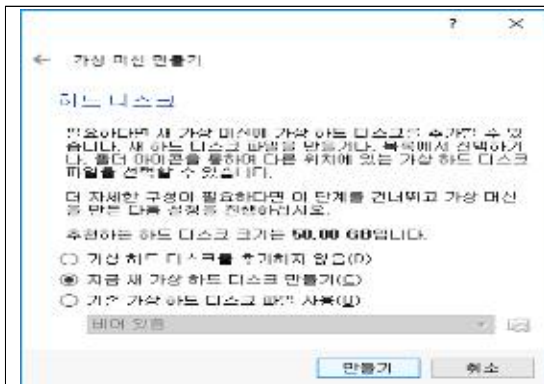
exec /sbin/init

(init 과정을 거치지 않으면 halt 상태가 되어 시스템 멈춤)

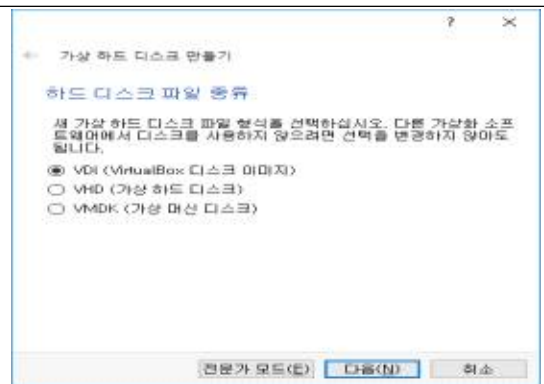
8. 부팅 완료 후 root 계정 로그인

[ 부록 - VirtualBox 설정 및 사용법 ]

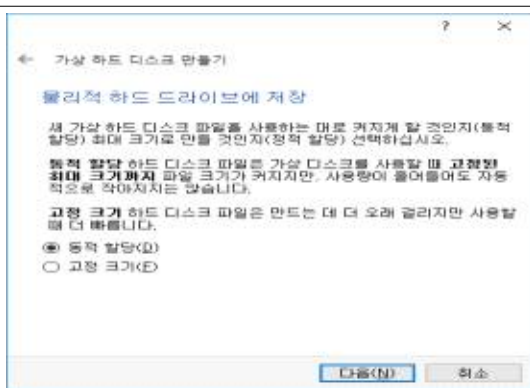
 <p>VirtualBox 실행화면</p>	 <p>파일 - 설정 실행 시 환경설정 화면</p>
 <p>입력 - 가상머신 항목에서 단축키 변경 가능 주로 호스트 키 조합을 변경할 때 사용</p>	 <p>네트워크 - NAT 네트워크 항목 우측 '+' 버튼 클릭 NAT 설정할 네트워크 주소 및 DHCP 사용 여부 설정</p>
 <p>첫 화면에서 새로 만들기 클릭 시 운영체제 선택</p>	 <p>가상머신에 할당할 메모리 크기 지정 초록색 범위까지가 안정범위</p>



가상머신에서 사용할 디스크 추가



VDI 형식을 가장 많이 사용함  
(속도 차이 때문)



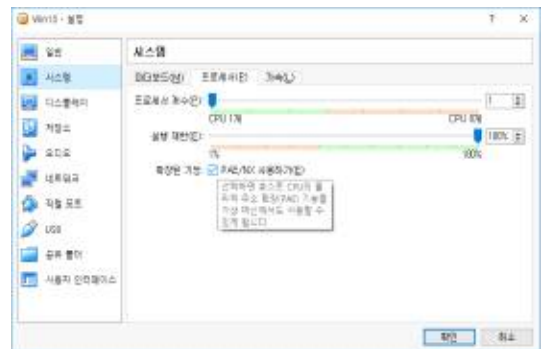
동적 할당 : 고정된 최대 크기까지 파일이 커지지만, 사용량 줄어도 자동적으로 작아지진 않음  
고정 크기 : 가상 머신용으로 새로 할당. 처음에 만들 때 오래 걸리지만, 사용 시 빠름. 디스크 공간 낭비 큼.



디스크 크기 조정  
Windows 의 경우 10GB 이내로도 사용이 가능하지만  
Linux 는 배포판 마다 다름.  
대략 8 ~ 10GB 정도가 기본 크기이나  
GUI 사용 시 10GB 도 부족하므로  
여유있게 20 ~ 30GB 이상 지정



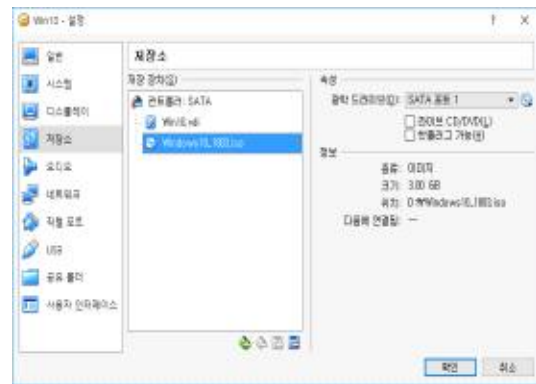
가상머신 상세 설정  
(주로 시스템, 저장소, 네트워크 정도만 설정)



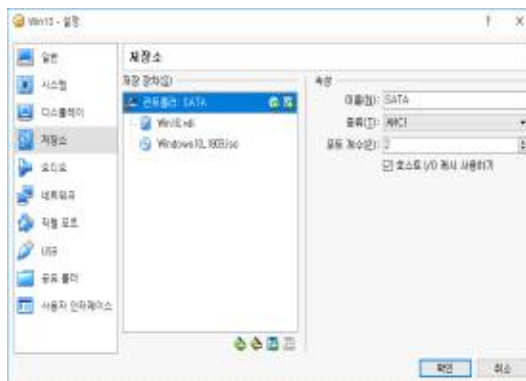
시스템 설정  
마더보드 탭 - [하드웨어 시각 UTC 보고] 체크 해제  
프로세서 탭 - CPU 개수 초록색 이내에서 지정



저장소 설정(설치미디어 삽입용)  
컨트롤러 항목 '비어 있음' 클릭 후 CD  
버튼 클릭



[가상 광 디스크 파일 선택] 클릭하여  
이미지 선택  
이미지 삽입 완료된 화면



컨트롤러 항목 클릭 후  
'호스트 I/O 캐시 사용' 체크



설정 완료



## 제2장

# 데이터베이스 기초 활용





## 1. 데이터베이스 개념

### (1) 데이터베이스 정의

데이터베이스란 **조직화된** 정보들의 모임으로 데이터베이스의 개념은 다음과 같이 네 가지로 설명할 수 있다.

통합된 데이터(Integrated data)	여러 곳에서 사용하던 데이터를 통합하여 하나로 저장한 데이터
저장된 데이터(Stored data)	문서가 아닌 디스크, 테이프 같은 컴퓨터 저장장치에 저장된 데이터
운영 데이터(Operational data)	조직의 목적을 위해 사용되는 데이터
공용 데이터(Shared data)	공동으로 사용되는 데이터

### (2) 데이터베이스 특징

실시간 접근성 (Real time accessibility)	데이터베이스는 실시간으로 서비스된다. 실시간이란 사용자가 요청하는 순간에 실제 데이터를 서비스하는 것을 말한다.
계속적인 변화 (Continuous change)	데이터베이스에 저장된 내용은 시간에 따라 항상 바뀐다. 데이터베이스는 삽입(insert), 수정(update), 삭제(delete) 등의 작업으로 최신의 데이터를 유지한다.
동시 공유 (Concurrent sharing)	데이터베이스는 여러 사용자에게 동시에 공유된다.
<b>내용에 의한 참조</b> (Content Reference)	데이터베이스의 데이터는 물리적인 위치가 아닌 데이터 값에 따라 참조된다. 사용자가 원하는 데이터를 얻기 위해 조건을 제시하면 데이터베이스는 이에 해당하는 데이터를 검색해 준다.

### (3) 데이터베이스 구성 요소

데이터베이스를 구성하는 요소는 사용자의 관점에서의 논리적 구성요소와 시스템의 관점에서의 물리적 구성요소로 나누어 생각할 수 있다.

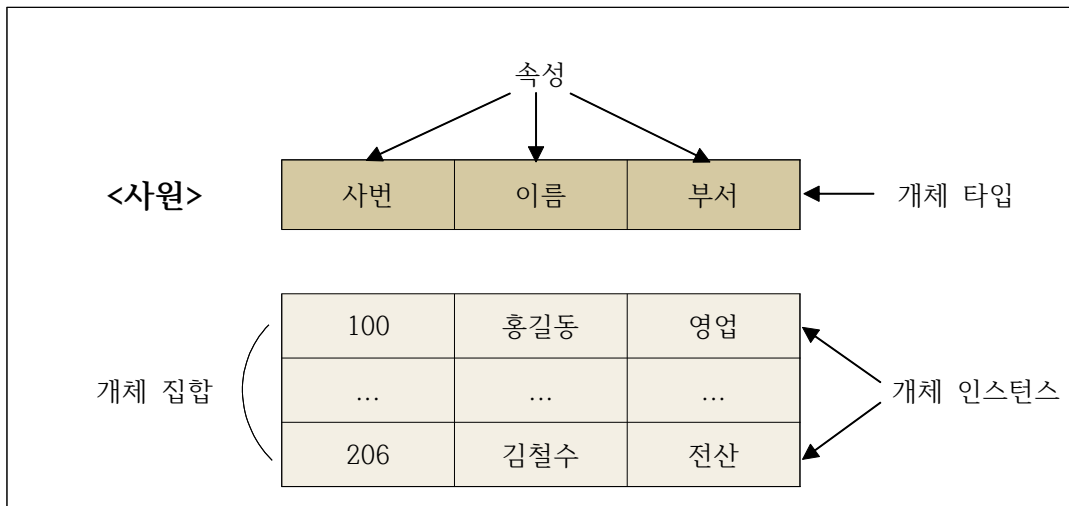
관점	구성요소
사용자 관점	논리적 구성요소 - 개체(entity), 관계(relationship)
시스템 관점	물리적 구성요소 - 비트, 바이트, 블록, 실린더 등

물리적 구성요소는 사용자에게는 큰 의미가 없으므로 사용자 측면의 논리적 구성 요소에 대해 알아보도록 한다.

### 1) 개체

데이터베이스 분야에서 **개체** 또는 **엔티티(entity)**라고 하는 것은 데이터베이스에 표현하려고 하는 유형, 무형의 객체로서 서로 구별되는 것을 뜻한다.

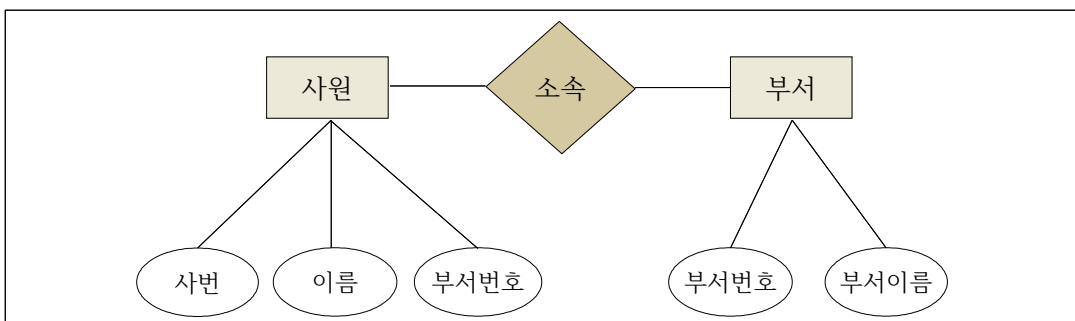
개체는 단독으로 존재할 수 있으며, 정보로서의 역할을 한다. 하나의 개체는 하나 이상의 속성, 즉 애트리뷰트(attribute)로 구성되고 각 속성은 그 개체의 특성이나 상태를 기술해 준다.



예를 들어, 사원이라는 개체는 사번, 이름, 부서라는 3개의 속성들로 구성되어 있다. 속성, 즉 애트리뷰트(attribute)라고 하는 것은 이름을 가진 데이터의 가장 작은 논리적 단위가 된다.

각 속성들, 즉 사번, 이름, 부서는 개별적으로는 우리에게 어떤 정보를 제공해 주지 못하지만 이것들이 모여 사원이라는 개체를 구성해서 표현할 때는 큰 의미를 제공하고 있다. 물론 각 속성이 갖는 값은 시간에 따라 변할 수 있다.

### 2) 관계



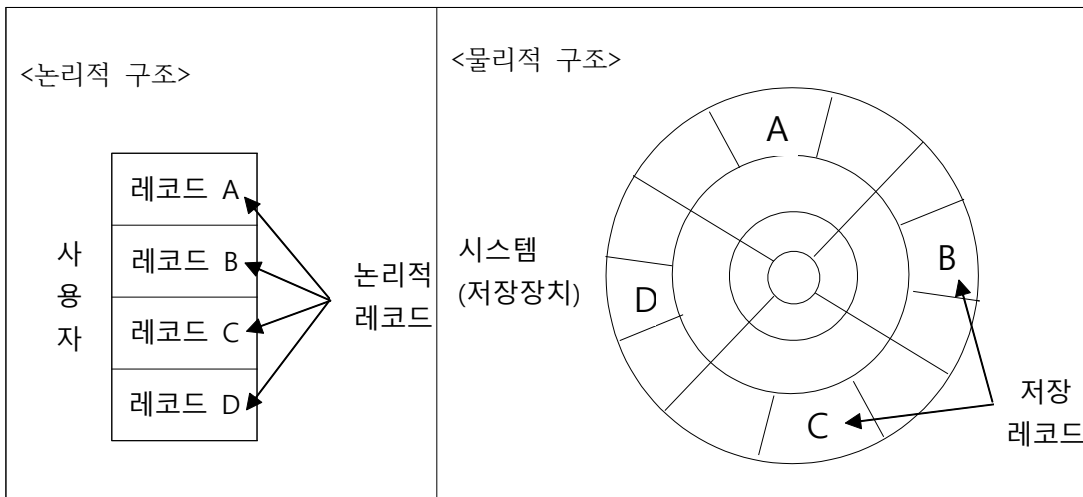
예를 들어, 사원 개체는 사번, 이름, 부서번호라는 속성들로 구성되어 있으며 이런 속성들이 사원 개체와 속성 관계이다. 또한 사원 개체와 부서 개체 사이의 “소속”이라는 관계는 개체 관계가 된다. 이와 같은 다이어그램을 E-R 다이어그램이라

한다. E-R 다이어그램에 대해서는 3장 데이터 모델링에서 다시 설명된다.  
관계(Relationship)는 실제로 정보를 추출해 내는데 중요한 역할을 한다. “사번이 100번인 사원의 이름을 검색하시오.”와 같은 질의문을 처리하기 위해서는 사번과 이름 사이에 존재하는 속성 관계를 통해서 정보를 추출해 낼 수 있는 반면 “사번이 100번인 사원의 부서 이름을 검색하시오.”라는 질의문은 사원과 부서 개체 사이에 존재하는 “소속”이라는 개체 관계를 이용해서만 정보 추출이 가능하다. 그러므로 관계(Relationship)는 데이터베이스를 접근하는데 중요한 도구가 된다.

#### (4) 데이터베이스 구조

데이터베이스는 값들로 구성되어 있으며 결국 물리적 저장 장치 위에 저장되어 있다. 이 저장 구조를 사용자 입장에서 보느냐, 시스템(저장장치)의 입장에서 보느냐에 따라 논리적 구조와 물리적 구조로 구별하고 있다.

논리적 구조 (Logical organization)	데이터베이스의 논리적 구조는 사용자가 생각하는 데이터의 논리적 표현이다. 즉, 데이터를 이용하는 응용 프로그램이나 일반 사용자의 입장에서 본 구조로서 데이터의 논리적 배치를 말한다.
물리적 구조 (Physical organization)	데이터베이스의 물리적 구조라고 하는 것은 디스크나 테이프와 같은 저장 장치 위에 물리적으로 저장되어 있는 데이터의 실제 구조를 말한다. 이것은 저장 장치의 입장에서 본 데이터베이스의 구조로서 저장 데이터의 물리적 배치를 표현한 것이다.



#### (5) 데이터베이스 시스템의 구성

- 스키마
- 데이터베이스 언어 **SQL**
- 사용자
- 데이터베이스 관리 시스템(DBMS)

- 데이터베이스 관리자(DBA)
- 하드웨어

## 1) 스키마

### ① 스카마의 정의

- 데이터베이스의 논리적 정의, 다시 말해 데이터 구조와 제약조건에 대한 명세를 기술한 것을 스키마(Schema)라 한다.
- 스키마는 데이터베이스를 구성하는 데이터 개체(Entity), 속성(Attribute), 관계(Relationship) 및 데이터 조작 시 데이터 값들이 갖는 제약조건 등에 관해 전반적으로 정의한다.
- 스키마는 사용자의 관점에 따라 외부 스키마, 개념 스키마, 내부 스키마로 나뉜다.

### ② 스키마의 3계층

외부 스키마 (External Schema)	<ul style="list-style-type: none"> <li>· 서브 스키마 또는 사용자 뷰라고도 한다.</li> <li>· 외부 스키마는 사용자나 응용 프로그래머가 각 개인의 입장에서 필요로 하는 데이터베이스의 논리적 구조를 정의한 것이다.</li> <li>· 외부 스키마는 전체 데이터베이스의 한 논리적인 부분으로 볼 수 있으므로 서브 스키마(Sub Schema)라고도 한다.</li> <li>· 하나의 데이터베이스 시스템에는 여러 개의 외부 스키마가 존재할 수 있으며, 하나의 외부 스키마를 여러 사용자가 함께 사용할 수 있다.</li> <li>· 일반 사용자는 질의어(SQL)를 이용하여 DB를 사용한다.</li> <li>· 응용 프로그래머는 C 등의 언어를 사용하여 DB에 접근한다.</li> </ul>
개념 스키마 (Conceptual Schema)	<ul style="list-style-type: none"> <li>· 개념 스키마는 데이터베이스의 전체적인 논리적 구조로서, 모든 응용 프로그램이나 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터베이스로 하나만 존재한다.</li> <li>· 개념 스키마는 개체 간의 관계와 제약조건을 나타내고 데이터베이스의 접근 권한, 보안 및 무결성 규칙에 관한 명세를 정의한다.</li> <li>· 단순히 스키마(Schema)라고 하면 개념 스키마를 의미한다.</li> <li>· 기관이나 조직체의 관점에서 데이터베이스를 정의한 것이다.</li> <li>· 데이터베이스 관리자(DBA)에 의해서 구성된다.</li> </ul>
내부 스키마 (Internal Schema)	<ul style="list-style-type: none"> <li>· 내부 스키마는 데이터베이스의 물리적 구조를 정의한다.</li> <li>· 내부 스키마는 데이터의 실제 저장 방법을 기술한다.</li> <li>· 물리적 저장장치와 밀접한 계층이다.</li> <li>· 시스템 프로그래머나 시스템 설계자가 보는 관점의 스키마이다.</li> </ul>

## 2) 데이터베이스 언어

데이터베이스 시스템은 전용 언어인 SQL을 사용한다. SQL은 데이터베이스를 구축하고 이용하기 위한 데이터베이스 시스템과 통신하는 수단이다.

SQL의 자세한 내용한 4장 데이터베이스 기본 명령어에서 알아본다.

데이터 정의어 (Data Definition Language)	<ul style="list-style-type: none"> <li>· 데이터 정의어는 DB구조, 객체 생성/수정/삭제, 데이터 형식, 접근 방식 등 DB를 구축하거나 수정할 목적으로 사용하는 언어이다.</li> <li>· create, alter, drop, truncate 구문과 같이 DB에 저장된 객체 구조를 정의한다.</li> </ul>
데이터 조작어 (Data Manipulation Language)	<ul style="list-style-type: none"> <li>· 데이터 조작어는 사용자로 하여금 데이터를 처리할 수 있게 하는 언어이다.</li> <li>· select, insert, update, delete 구문과 같이 데이터를 검색, 삽입, 수정, 삭제하는데 사용하는 언어이다.</li> </ul>
데이터 제어어 (Data Control Language)	<ul style="list-style-type: none"> <li>· 데이터 제어어는 무결성 및 권한 제어 등을 하기 위한 언어이다.</li> <li>· grant, revoke 구문과 같이 데이터 사용 권한을 관리 하는 언어이다.</li> </ul>

## 3) 사용자

데이터베이스 시스템을 사용하는 사람은 크게 DBA, 응용 프로그래머, SQL 사용자, 일반 사용자로 구분할 수 있다.

데이터베이스 관리자	<ul style="list-style-type: none"> <li>· 데이터베이스 시스템의 모든 관리와 운영에 대한 책임을 지고 있는 사람이다.</li> <li>· DBA는 데이터 설계, 구현, 유지보수의 전 과정을 담당하고 있다.</li> </ul>
응용 프로그래머	<ul style="list-style-type: none"> <li>· 일반 사용자가 사용할 수 있도록 프로그램을 만드는 사람이다.</li> <li>· 응용 프로그래머는 C, Java, Jsp 등의 프로그래밍 언어와 SQL로 일반 사용자를 위한 사용자 인터페이스와 데이터를 관리하는 프로그램을 개발한다.</li> </ul>
SQL 사용자	<ul style="list-style-type: none"> <li>· SQL로 업무를 처리하는 IT 부서의 담당자이다.</li> <li>· 응용 프로그램으로 구현되어 있지 않은 업무를 SQL로 처리한다.</li> <li>· 주로 데이터 검색, 데이터 변경, 데이터를 모니터링하는 업무를 한다.</li> </ul>
일반 사용자	<ul style="list-style-type: none"> <li>· 은행의 창구직원, 관공서의 접수처 직원, 병원의 의사/간호사/접수처 등에서 데이터를 다루는 업무를 하는 사람이다.</li> <li>· 프로그래머가 개발한 프로그램을 이용하여 데이터베이스에 접근 한다.</li> <li>· 보통 특별한 지식 없이 데이터베이스에 접근하여 데이터의 검색, 삽입, 수정, 삭제 작업을 하는 사람이다.</li> </ul>

#### 4) 데이터베이스 관리 시스템(DBMS, Data Base Management System)

DBMS는 사용자와 데이터베이스를 연결시켜주는 소프트웨어로, 데이터베이스 사용자가 데이터베이스를 생성, 공유, 관리할 수 있도록 지원해주는 총체적인 역할을 한다.

DBMS에 대한 자세한 내용은 2장 데이터베이스 관리 시스템에서 알아본다.

#### 5) 데이터베이스 관리자(DBA, Database Administrator)

데이터베이스 관리자(DBA)는 데이터베이스 시스템의 관리 운영에 대한 모든 책임을 지고 있는 사람으로 데이터베이스 관리자가 책임지고 있는 업무는 다음과 같다.

- 데이터베이스 관련 소프트웨어 설치 및 생성
- 스키마 정의
- 저장 영역 관리
- 보안 및 권한 부여 정책, 데이터의 유효성 검사
- 백업, 복구 작업
- 데이터베이스 다중화
- 데이터베이스 성능 관리
- 데이터의 표현이나 시스템의 문서화에 표준 설정
- 사용자와 개발자의 요구와 불평을 청취 및 해결
- 시스템 자원의 이용도, 병목 현상, 장비 및 시스템 성능 감시
- 데이터 접근, 저장 공간, 데이터의 이용 추세, 각종 통계 정보 등을 수집 및 분석

#### 6) 하드웨어

데이터베이스 시스템은 기본적으로 컴퓨터 중심 시스템이기 때문에 컴퓨터 하드웨어 시스템의 성능은 곧 데이터베이스 시스템의 성능과 직결된다. 특히 실시간 접근, 계속적인 변화, 동시 공유, 내용식 참조와 같은 특성을 가지고 있는 대규모 데이터베이스를 관리하고 운용하기 위해서는 고성능의 주기억 장치, CPU, 입출력 연산, 그리고 대용량 저장 장치 등이 요구된다.



## 2. 데이터베이스 관리 시스템

### (1) DBMS 정의

- 데이터베이스 관리 시스템(DBMS, Database Management System)이란 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해 주고, 데이터베이스를 관리해 주는 소프트웨어이다.
- DBMS는 기존의 파일 시스템이 갖는 데이터의 종속성과 중복성의 문제를 해결하기 위해 제안된 시스템으로 모든 응용 프로그램들이 데이터베이스를 공유할 수 있도록 관리해 준다.

### (2) DBMS 발전 배경

#### 1) 파일 처리 방식에서의 데이터 구성

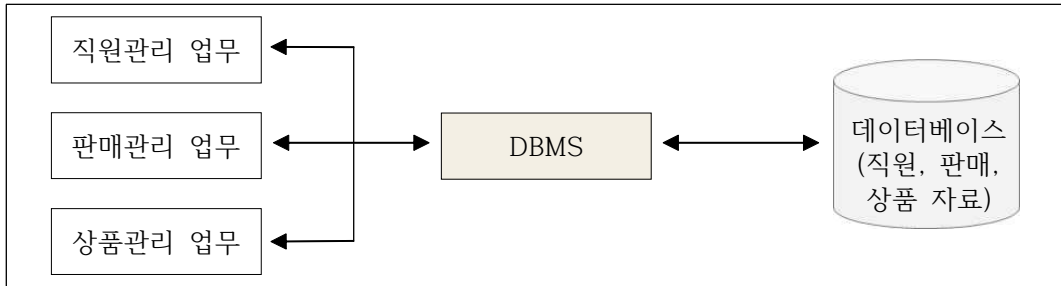
- 파일 처리 방식에서는 처리 업무 하나하나마다 데이터 파일을 독립적으로 구성함으로써, 같은 내용의 데이터가 서로 다른 업무의 파일에서 중복 저장되어 관리되는 데이터 중복성과 응용 프로그램과 데이터 간의 상호 의존 관계에 따른 데이터 종속성을 초래하는 단점이 있다.

처리 업무	파일	파일구성항목
직원관리 업무	↔ 직원 파일	직원번호, 직원이름, 주소, 전화번호
판매관리 업무	↔ 판매 파일	직원번호, 직원이름, 주소, 전화번호, 상품번호, 상품명, 가격, 판매수량
상품관리 업무	↔ 상품 파일	상품번호, 상품명, 가격, 입고수량, 입고 일자

- **종속성으로 인한 문제점** : 응용 프로그램과 데이터 파일이 상호 의존적인 관계에서는 데이터 파일이 보조기억장치에 저장되는 방법이나 저장된 데이터의 접근 방법을 변경할 때 응용 프로그램도 같이 변경하여야 한다.
- **중복성으로 인한 문제점**
  - 일관성 : 중복된 데이터 간에 내용이 일치하지 않는 상황이 발생함
  - 보안성 : 중복되어 있는 모든 데이터에 동등한 보안수준을 유지하기 힘들
  - 경제성 : 저장 공간의 낭비와 동일 데이터의 반복 작업으로 비용이 증가함
  - 무결성 : 제어의 분산으로 데이터의 정확성을 유지할 수 없음

#### 2) 데이터베이스 도입에서의 데이터 구성

- 데이터베이스를 이용하는 방식에서는 각 업무 처리에 필요한 자료들의 중복을 최소화하여 한 곳에 모아서 구성한다.



### (3) DBMS 필수 기능

<b>정의</b>	데이터베이스에 저장될 데이터의 형(type)과 구조에 대한 정의, 이용 방식, 제약조건 등을 명시하는 기능이다.
<b>조작</b>	데이터 검색, 삽입, 수정, 삭제 등을 체계적으로 처리하기 위해 사용자와 데이터베이스 사이의 인터페이스 수단을 제공하는 기능이다.
<b>제어</b>	데이터의 정확성과 안전성을 유지하기 위한 무결성, 보안 및 권한 검사, 병행 수행 제어 등의 기능을 정하는 기능이다.

### (4) DBMS 종류 및 특징

Oracle	<ul style="list-style-type: none"> <li>· 오라클에서 만들어 판매중인 상업용 RDBMS</li> <li>· 윈도우, 리눅스, 유닉스 등 다양한 운영체제에 설치 가능</li> <li>· MySQL, MSSQL보다 <b>대용량의 데이터 처리가 용이함</b></li> <li>· <b>대기업, 관공서</b>에서 주요 사용, 글로벌 DB 시장 점유율 1위</li> <li>· 가장 널리 사용되는 RDBMS, 비공개 소스, 폐쇄적인 운영</li> </ul>
MySQL	<ul style="list-style-type: none"> <li>· MySQL사에서 개발, 썬마이크로시스템즈를 거쳐 현재 오라클에 인수 합병됨</li> <li>· 윈도우, 리눅스, 유닉스 등 다양한 운영체제에서 설치 가능</li> <li>· 오픈소스로 이루어져있는 무료 프로그램(상업적 사용 시 비용 발생)</li> <li>· 가격 등의 장점을 앞세워 <b>다수의 중소기업</b>에서 사용 중인 RDBMS</li> </ul>
MSSQL	<ul style="list-style-type: none"> <li>· 마이크로소프트(MS)사에서 개발한 상업용 RDBMS</li> <li>· 다른 운영체제에서도 사용 가능은 하지만 윈도우에 특화됨</li> <li>· 비공개 소스로 폐쇄적인 운영(리눅스 버전은 오픈소스)</li> <li>· 중소기업에서 주로 사용 중임</li> </ul>
MariaDB	<ul style="list-style-type: none"> <li>· MySQL이 오라클에 인수 합병된 후 불확실한 라이선스 문제를 해결하려고 나온 오픈소스 RDBMS</li> <li>· 구현언어 : C++</li> <li>· MySQL과 동일한 소스 코드 기반, MySQL보다 애플리케이션 부분 속도가 빠름</li> </ul>

### 3. 데이터 모델링

#### (1) 데이터 모델링의 개념

- 데이터 모델은 현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형이다.
- 현실 세계를 데이터베이스에 표현하는 중간 과정, 즉 데이터베이스 설계 과정에서 데이터의 구조를 논리적으로 표현하기 위해 사용되는 도구이다.

#### (2) 데이터 모델링 과정

데이터 모델링 과정은 요구사항 수집 및 분석 -> 설계 -> 데이터베이스 구현 단계로 진행된다. 요구사항 수집 및 분석 단계에서는 현실 세계의 대상 및 사용자의 요구사항을 정리하고 분석한다. 설계 단계에서는 중요 개념을 구분(개념적 모델링)한 후 각 개념을 구체화(논리적 모델링)하고 최종적으로 데이터베이스에 테이블, 인덱스 등을 생성(물리적 모델링)한다.

##### 1) 요구사항 수집 및 분석

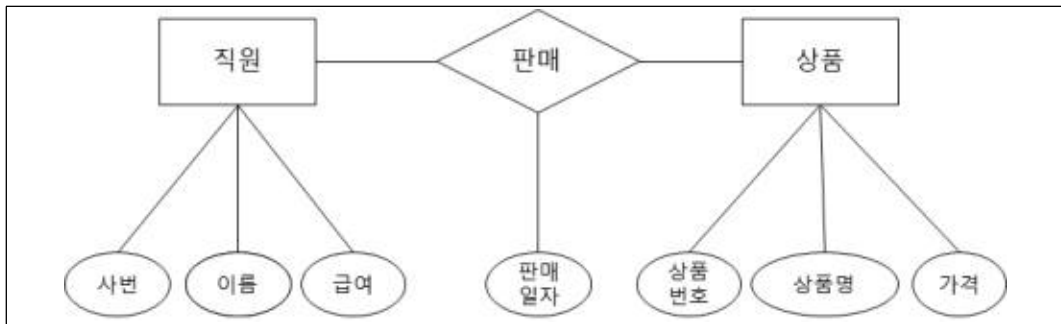
데이터베이스 구축을 위한 첫 단계로, 현실 세계를 파악하고 사용자의 요구사항을 수집 및 분석하는 단계이다. 앞으로의 진행 방향이 이 단계에서 결정되므로 매우 중요하다. 담당자는 구축할 데이터베이스와 관련된 전문적인 지식을 갖추고 있어야 한다. 일반적으로 요구사항 수집을 위해 다음과 같은 방법이 많이 사용된다.

- 실제 문서를 수집하고 분석한다.
- 담당자와의 인터뷰나 설문조사를 통해 요구사항을 수집한다.
- 비슷한 업무를 처리하는 기존의 데이터베이스를 분석한다.

사용자의 요구사항은 대부분 모호하며 사용자마다 쓰는 용어도 달라 개념을 명확하게 파악하기 어렵다. 이러한 모호성을 제거하고 최대한 구체적이고 명확하게 정리해야 한다. 또한 분석 과정을 거쳐 필요로 하는 정보와 업무 처리 절차를 도출하여 다음 단계에서 활용할 수 있도록 한다.

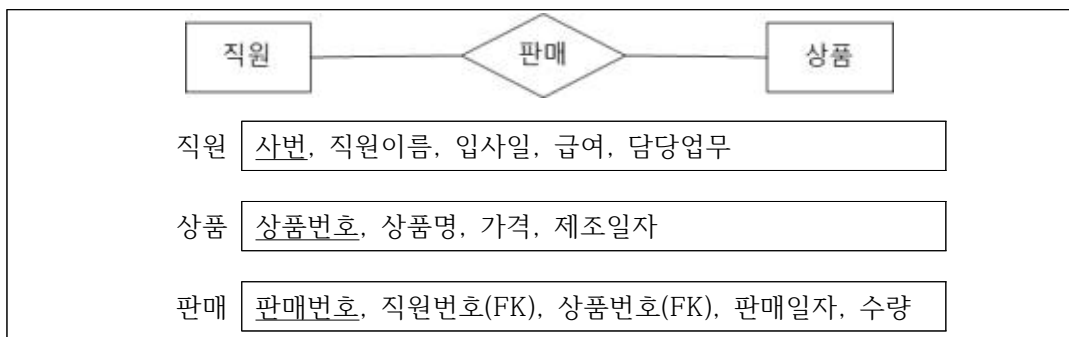
##### 2) 개념적 모델링

개념적 모델링은 요구사항을 수집하고 분석한 결과를 토대로 업무의 핵심적인 개념을 구분하고 전체적인 뼈대를 만드는 과정이다. 즉, 개체(entity)를 추출하고 각 개체들 간의 관계(relationship)를 정의하여 ER다이어그램(ERD, Entity Relationship Diagram)을 만드는 과정이다.



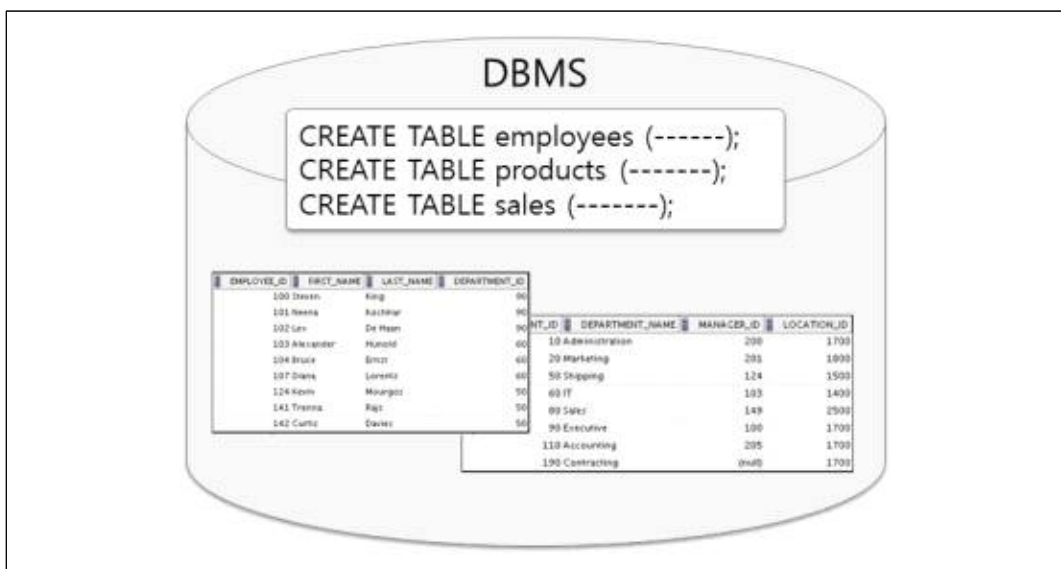
### 3) 논리적 모델링

논리적 모델링은 개념적 모델링에서 만든 ERD를 사용하고자 하는 DBMS에 맞게 매핑하여 실제 데이터베이스로 구현하기 위한 모델을 만드는 과정이다.



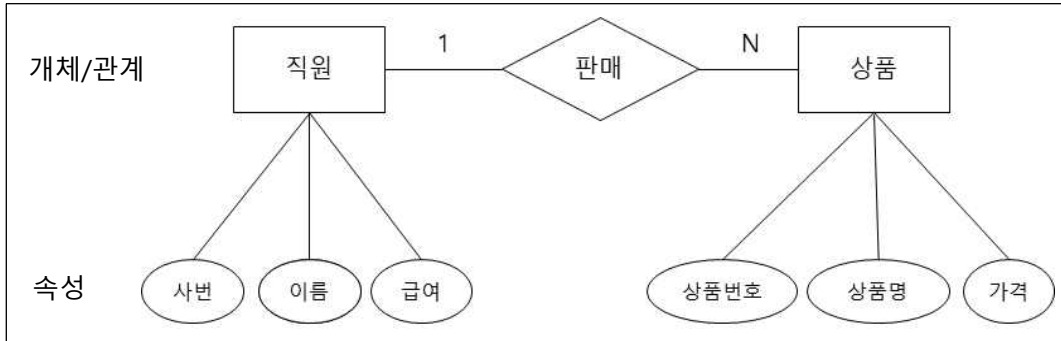
### 4) 물리적 모델링

물리적 모델링은 작성된 논리적 모델을 실제 컴퓨터의 저장 장치에 저장하기 위한 물리적 구조를 정의하고 구현하는 과정이다. 물리적 모델링을 할 때는 DBMS의 특성에 맞게 저장 구조를 정의하여야 데이터베이스가 최적의 성능을 낼 수 있다.



### (3) ER 다이어그램

ER(Entity Relationship) 모델은 데이터 모델링 과정 중 개념적 모델링에 사용하는 모델로 1976년 피터 첸(Peter Chen)이 제안하였다. 개체, 속성, 관계를 규명하는 것이 ER 모델의 기본이며, 개체와 개체 간의 관계를 표준화된 그림으로 표현한 것이 ER 다이어그램이다.



ER 모델은 개념적 모델링 단계에서 사용하기 때문에 DBMS와 무관하게 설계할 수 있다.

#### 1) 개체(Entity) = 객체=object=table

개체는 사람, 사물, 장소, 개념, 사건과 같은 유무형의 정보를 가지고 있는 독립적인 실체를 말한다.

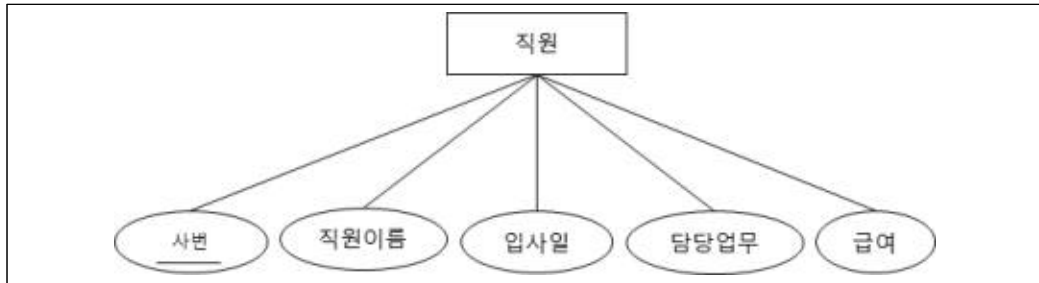
개체 타입 유형	기호	유형
강한 개체 타입	직원	다른 개체의 도움 없이 독자적으로 존재할 수 있는 개체 (보통 개체 타입이라고 하면 강한 개체 타입을 말한다.)
약한 개체 타입	직원	독자적 존재할 수 없고 반드시 상위 개체 타입을 가짐

#### 2) 속성(Attribute) =column(열)

속성은 개체가 가진 성질을 말한다. 예를 들어 '직원'의 경우 사번, 직원이름, 입사일, 담당업무, 급여 등의 속성을 가질 수 있다.

##### ① 속성의 ER 다이어그램 표현

속성은 기본적으로 타원으로 표현하며 개체 타입을 나타내는 직사각형과 실선으로 연결된다. 속성이 개체를 유일하게 식별할 수 있는 키일 경우 속성 이름에 밑줄을 긋는다. 직원 개체와 속성을 ER 다이어그램으로 나타내면 다음과 같다.



## ② 속성의 유형

속성은 값의 형태에 따라 다음과 같이 여러 유형으로 구분할 수 있다.


<ul style="list-style-type: none"> <li>· 단순 속성 : 더 이상 분해가 불가능한 속성</li> <li>· 복합 속성 : 독립적인 의미를 가진 속성으로 분해할 수 있는 속성</li> </ul>	
<ul style="list-style-type: none"> <li>· 단일값 속성 : 하나의 값만을 가지는 속성</li> <li>· 다중값 속성 : 여러 개의 값을 가지는 속성, 예를 들어 자격증은 한 사람이 여러 개의 취득할 수 있다.</li> </ul>	
<ul style="list-style-type: none"> <li>· 저장 속성 : 다른 속성의 영향 없이 단독으로 저장되는 속성</li> <li>· 유도 속성 : 다른 저장 속성으로부터 유도된(계산된) 속성</li> </ul>	

## 3) 관계(Relationship)

ER 모델은 개체와 개체 사이의 관계 또는 연관성을 말한다.

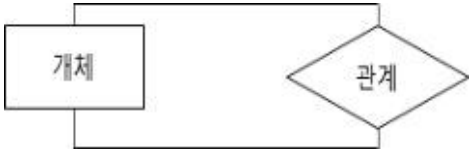


### ① 관계 타입의 ER 다이어그램 표현

관계 타입은 마름모로 표현한다.

기호	의미
	관계 타입

## ② 관계 타입의 유형

### - 차수에 따른 유형

기호	설명
	한 개의 개체가 자기 자신과 관계를 맺음
	두 개의 개체가 관계를 맺음
	세 개의 개체가 관계를 맺음

### - 관계 대응 수에 따른 유형

기호	의미	설명
	일대일 관계	하나의 개체가 하나의 개체에 대응
	일대다 관계	하나의 개체가 여러 개체에 대응
	다대일 관계	여러 개체가 하나의 개체에 대응
	다대다 관계	여러 개체가 여러 개체에 대응

### - 참여 제약 조건

관계의 특성 중 하나로 참여 제약 조건이 있다. 참여 제약 조건은 개체 집합 내 모든 개체가 관계에 참여하는지 유무에 따라 전체 참여와 부분 참여로 구분할 수 있다. 전체 참여는 개체 타입과 관계를 두 줄 실선으로 표현하고, 부분 참여는 일반적인 관계 표현과 동일하게 단일 실선으로 표현한다.



기호	의미
	전체 참여
	부분 참여



(전체 참여와 부분 참여의 예)



직원의 경우 휴가 또는 영업 부진 등의 이유로 판매를 하지 못한 직원이 있을 수 있으므로 판매 관계와 부분 참여를 맺는다. 그러나 상품의 경우 판매되지 않는 상품은 생산을 하지 않으므로 하나도 판매되지 않은 상품은 없다고 가정한다면 판매 관계와 전체 참여를 맺는다.

#### 4) IE 표기법 좀더 구체적인 내용 표기가능

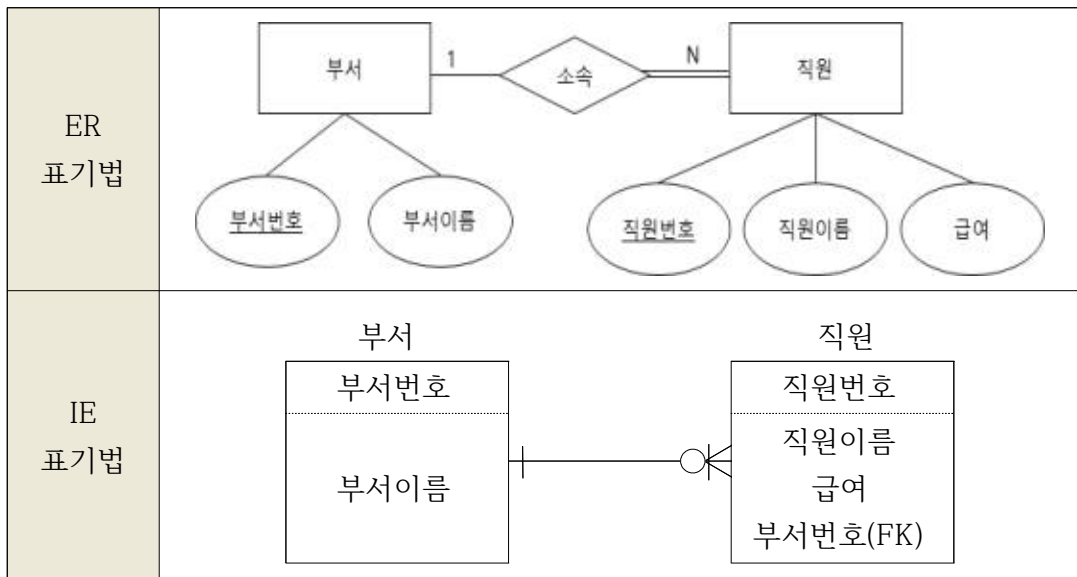
ER 모델 표기법은 여러 표기법이 존재한다. 지금까지의 ER 다이어그램 표기법은 피터 첸이 제안한 ER 모델의 기본적인 표기법이였다. 또 다른 대표적인 표기법으로 IE 표기법(Information Engineering Notation)이 있으며 새발 표기법이라고도 한다.

피터 첸 표기법	IE 표기법
	

IE 표기법은 개체 타입과 속성은 직사각형으로 표현한다. 관계는 마름모꼴 대신 개체 타입인 직사각형을 관계 실선으로 연결하고 다음과 같은 모양의 기호를 이용하여 관계 대응 수 등을 표현한다. 관계 실선에서 개체 쪽의 기호는 반대쪽 개체의 최대참여(1 혹은 N)를 나타내며, '○', '1' 기호는 최소참여로 선택참여(○) 혹은 필수참여(1)를 나타낸다.

기호	의미
-----	· 비식별자 관계 : 강한 개체 타입 · 부모 개체의 키가 일반 속성으로 포함되는 관계
_____	· 식별자 관계 : 약한 개체 타입 · 부모 개체의 키가 주식별자로 포함되는 관계
_____<	· 일대다(1:N)의 관계 : N쪽에 새발을 표시
_____○	· 0(선택 참여), 최소 참여가 0일 경우
_____+	· 1(필수 참여), 최소 참여가 1일 경우

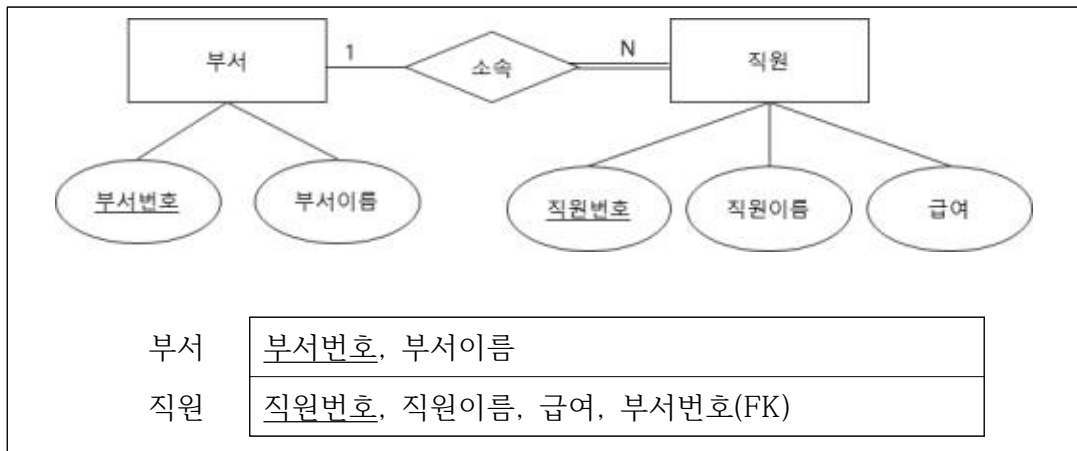
(IE 표기법 예)



#### (4) 관계 데이터 모델

ER 모델은 데이터베이스 생명주기의 개념적 모델링에서 사용하는 모델로, ER 다이어그램을 통해 완성된다. 완성된 ER 모델은 실제 데이터베이스로 구축하기 위해 논리적 모델링 단계를 거치는데, 이 단계에서 사상(Mapping)이 이루어진다. ER 모델을 관계 데이터 모델로 사상하는 방법은 다음과 같이 7단계 알고리즘을 거친다.

단계	사상할 대상	구분
1단계	개체 타입	강한 개체 타입
2단계		약한 개체 타입
3단계	관계 타입	이진 1:1 관계 타입
4단계		이진 1:N 관계 타입
5단계		이진 N:M 관계 타입
6단계		N진 관계 타입
7단계	속성	다중값 속성



[예제] 다음 내용을 토대로 데이터 모델링 작업을 수행하시오.

- WILL 쇼핑몰 가입 페이지는 다음과 같다.
- WILL 쇼핑몰에서 판매하고 있는 **상품**에 대해서는 고유한 번호가 있으며, 상품명, 제조일자, 수량 등의 정보를 포함하여야 한다.
- WILL 쇼핑몰에서는 회원가입 후 상품을 구매할 수 있다.

**회원가입**

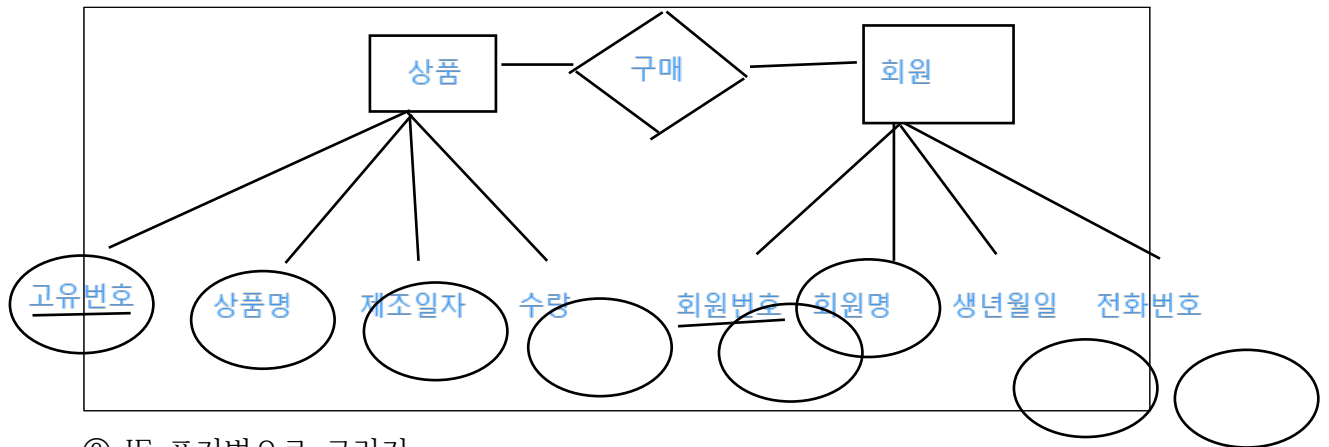
회원번호 : 자동부여

회원명 :

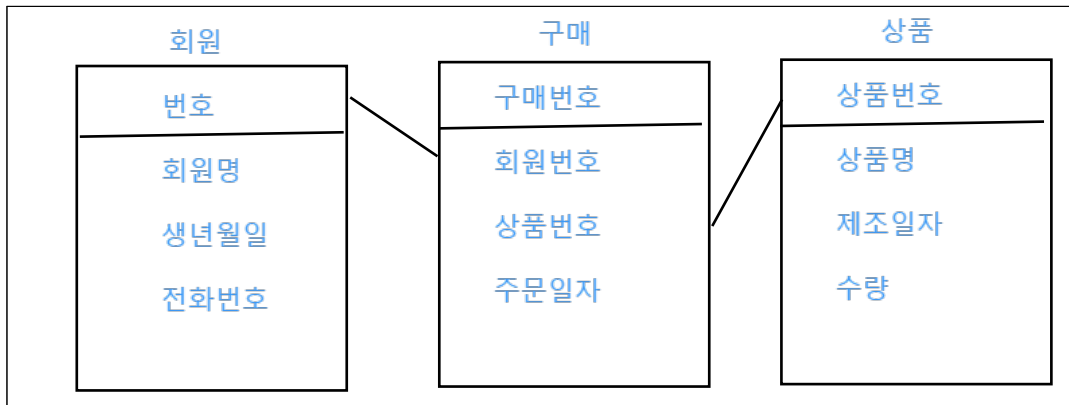
생년월일 :

전화번호 :

① ER 다이어그램 그리기



② IE 표기법으로 그리기

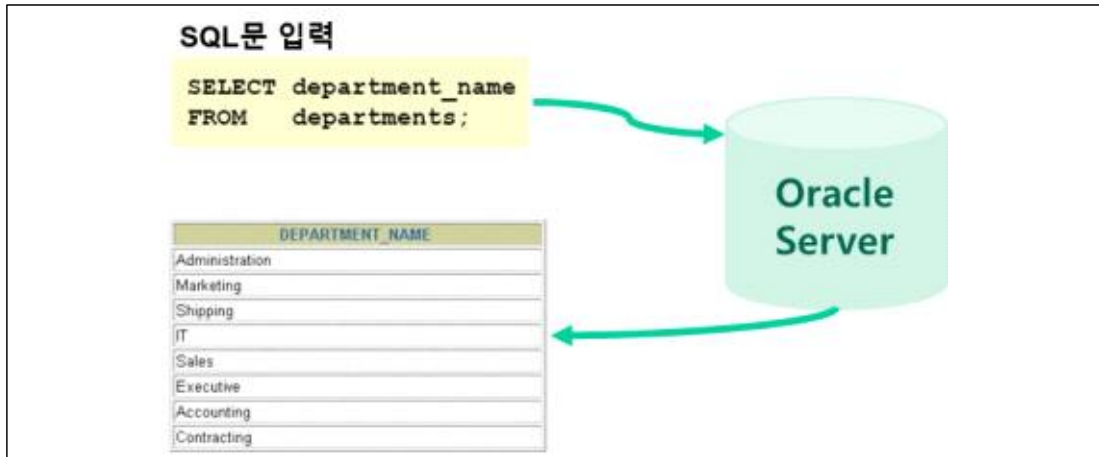


③ 관계 데이터 모델로 사상하기

## 4. 데이터베이스 기본 명령어

### (1) SQL 개념

SQL이란 관계형 데이터베이스 작동을 위한 ANSI(American National Standards Institute) 표준 언어이다. SQL은 모든 프로그램 및 사용자가 데이터베이스의 데이터를 액세스하기 위해 사용하는 일련의 명령문이다.



SQL에서는 다양한 작업에 대한 명령문을 제공한다.

명령문		설명
DDL	CREATE ALTER DROP RENAME TRUNCATE COMMENT	<ul style="list-style-type: none"> <li>- 객체를 생성, 수정, 삭제하는 명령문으로 이러한 명령문을 통틀어 DDL(데이터 <b>정의어</b>)이라고 한다.</li> <li>- 객체 종류 : 테이블, 뷰, 시퀀스, 인덱스 등</li> </ul>
DML	SELECT INSERT UPDATE DELETE MERGE	<ul style="list-style-type: none"> <li>- 데이터베이스에서 데이터를 검색하고, 행을 삽입/수정/삭제하는 명령문으로 이러한 명령문을 통틀어 DML(데이터 <b>조작어</b>)이라고 한다.</li> </ul>
DCL	GRANT REVOKE	<ul style="list-style-type: none"> <li>- 데이터베이스와 그 안의 구조에 대한 액세스 권한을 부여하거나 회수하는 명령문을 DCL(데이터 <b>제어어</b>)이라고 한다.</li> </ul>
트랜잭션 제어 명령어	COMMIT ROLLBACK SAVEPOINT	<ul style="list-style-type: none"> <li>- DML문으로 인한 데이터 변경 작업은 논리적인 작업 단위인 트랜잭션으로 묶일 수 있으며, 이러한 트랜잭션을 제어하는 명령문이다.</li> </ul>

(2) 데이터 정의어(DDL) **롤백 안됨**

## 1) 테이블 생성(create table)

- 테이블 생성 시 테이블명, 컬럼명, 데이터타입, 컬럼 사이즈 지정해야함
- 옵션 : Default값, 제약조건 추가 가능

```
CREATE TABLE dept
(
    deptno      NUMBER(2),
    dname       VARCHAR2(14),
    loc         VARCHAR2(13),
    create_date  DATE DEFAULT SYSDATE);
```

CREATE TABLE succeeded.

2) **제약조건** - Primary key, Foreign key, Unique, Not Null, Check  
[문법]

- Example of a column-level constraint:

```
CREATE TABLE employees(
    employee_id NUMBER(6)
    CONSTRAINT emp_emp_id_pk PRIMARY KEY,
    first_name  VARCHAR2(20),
    ...);
```

1

- Example of a table-level constraint:

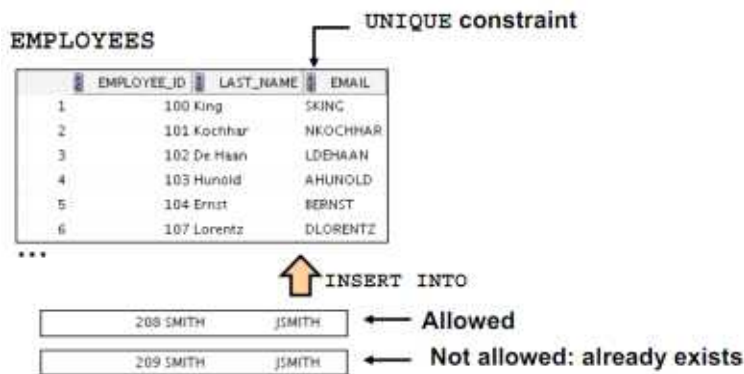
```
CREATE TABLE employees(
    employee_id NUMBER(6),
    first_name  VARCHAR2(20),
    ...
    job_id      VARCHAR2(10) NOT NULL,
    CONSTRAINT emp_emp_id_pk
    PRIMARY KEY (EMPLOYEE_ID));
```

2

① NOT NULL 제약조건 - 컬럼 레벨의 문법만 가능

```
SQL> create table test1
  ( id number(10) constraint t1_id_nn not null,
    name varchar2(30) constraint t1_name_nn not null,
    job varchar2(20),
    email varchar2(20),
    phone varchar2(20) constraint t1_ph_nn not null,
    start_date date );
```

② UNIQUE 제약조건



```
SQL> create table test2
  ( id number(10) constraint t2_id_nn not null
    constraint t2_id_uk unique,
    name varchar2(30) constraint t2_name_nn not null,
    job varchar2(20),
    email varchar2(20),
    phone varchar2(20) constraint t2_ph_nn not null
    constraint t2_ph_uk unique,
    start_date date,
    constraint t2_email_uk unique(email) );
```

③ PRIMARY KEY 제약조건

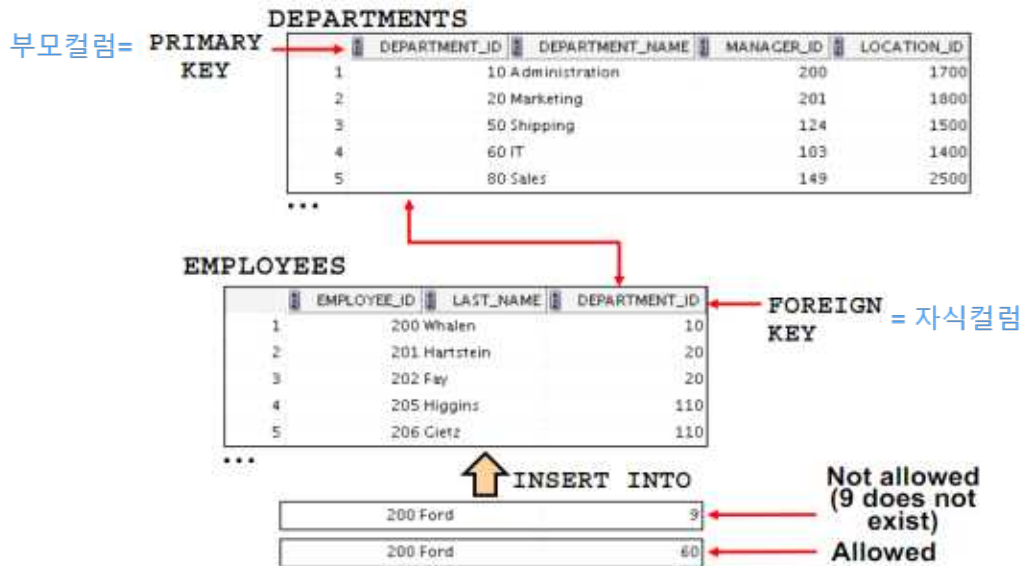
- 기본키 제약조건
- **NOT NULL + UNIQUE의 성격을 가짐** 테이블당 한번만 선언 가능함

```
SQL> create table test3
  ( id number(10) constraint t3_id_pk primary key,
    name varchar2(30) constraint t3_name_nn not null,
    job varchar2(20),
    email varchar2(20),
    phone varchar2(20) constraint t3_ph_nn not null
    constraint t3_ph_uk unique,
    start_date date,
    constraint t3_email_uk unique(email) );
```



④ FOREIGN KEY 제약조건 외래키 : 부모컬럼에 있는 값만 자식컬럼에 들어올 수 있음

- 정의 : 자기 자신 테이블이나 다른 테이블의 특정 컬럼(PK, UK)을 참조하는 제약조건



[FK - table level 문법]

```
CREATE TABLE employees (
  employee_id      NUMBER(6),
  last_name        VARCHAR2(25) NOT NULL,
  email            VARCHAR2(25),
  salary           NUMBER(8,2),
  commission_pct   NUMBER(2,2),
  hire_date        DATE NOT NULL,
  ...
  department_id    NUMBER(4),
  CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
    REFERENCES departments(department_id),
  CONSTRAINT emp_email_uk UNIQUE(email));
```

[FK - column level 문법]

```
CREATE TABLE employees
(
  ...
  department_id NUMBER(4) CONSTRAINT emp_deptid_fk
    REFERENCES departments(department_id),
  ...
)
```

⑤ CHECK 제약조건 해당 컬럼이 만족해야하는 조건문을 지정하는 제약조건

```
..., salary NUMBER(2)
  CONSTRAINT emp_salary_min
    CHECK (salary > 0),...
```

## 3) 테이블 생성 시 제약조건 정의하여 테이블 생성하기

```

CREATE TABLE employees
( employee_id      NUMBER(6)
  CONSTRAINT emp_employee_id PRIMARY KEY
, first_name       VARCHAR2(20)
, last_name        VARCHAR2(25)
  CONSTRAINT emp_last_name_nn NOT NULL
, email            VARCHAR2(25)
  CONSTRAINT emp_email_nn    NOT NULL
  CONSTRAINT emp_email_uk    UNIQUE
, phone_number     VARCHAR2(20)
, hire_date        DATE
  CONSTRAINT emp_hire_date_nn NOT NULL
, job_id           VARCHAR2(10)
  CONSTRAINT emp_job_nn      NOT NULL
, salary           NUMBER(8,2)
  CONSTRAINT emp_salary_ck   CHECK (salary>0)
, commission_pct   NUMBER(2,2)
, manager_id       NUMBER(6)
  CONSTRAINT emp_manager_fk REFERENCES
    employees (employee_id)
, department_id    NUMBER(4)
  CONSTRAINT emp_dept_fk    REFERENCES
    departments (department_id));

```

## 4) 테이블 수정(alter table)

## ① 테이블에 컬럼 추가

```

ALTER TABLE dept80
ADD      (job_id VARCHAR2(9));

```

## ② 테이블의 기존 컬럼 수정 - 데이터타입, 컬럼 사이즈, Default값 등...

```

ALTER TABLE dept80
MODIFY      (last_name VARCHAR2(30));

```

## ③ 테이블의 기존 컬럼 삭제

```

ALTER TABLE dept80
DROP COLUMN job_id;

```

## 5) 테이블 삭제(drop table)

- 데이터베이스로부터 테이블의 구조, 데이터, 제약조건을 삭제함
- DDL구문이므로 rollback 안됨

```

DROP TABLE dept80;
DROP TABLE dept80 succeeded.

```

6) 테이블 절단(truncate table)

- 테이블 구조는 남겨두고 테이블의 모든 행 제거함
- DDL구문이므로 rollback 안됨

```
TRUNCATE TABLE table_name;
```

(3) 데이터조작어(DML)

데이터조작어 (DML)	<ul style="list-style-type: none"> <li>- 테이블의 데이터 조회(select)</li> <li>- 테이블에 새로운 행 추가(insert)</li> <li>- 테이블의 기존 행 수정(update)</li> <li>- 테이블의 기존 행 삭제(delete)</li> </ul>
트랜잭션 (Transaction)	<ul style="list-style-type: none"> <li>- 논리적인 작업 단위</li> <li>- 여러 DML이 모여서 하나의 트랜잭션이 구성됨.</li> <li>- DML 작업 후 commit, rollback으로 트랜잭션 종료해야함.</li> </ul>

1) SELECT 구문

① 테이블 구조 조회

```
DESCRIBE employees
```

② 테이블로부터 데이터 검색

```
SELECT *
FROM departments;
```

```
SELECT department_id, location_id
FROM departments;
```

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

alias as는 생략가능

	NAME	COMM
1	Whalen	(null)
2	Hartstein	(null)
3	Fay	(null)

\*\*\*

## ③ WHERE절을 사용한 데이터 검색

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE department_id = 90 ;
```

```
SELECT last_name, job_id, department_id
FROM employees
WHERE last_name = 'Whalen' ;
```

```
SELECT last_name
FROM employees
WHERE hire_date = '17-FEB-96' ;
```

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000 ;
```

## ④ ORDER BY절을 사용한 데이터 검색

- ASC: Ascending order, default
- DESC: Descending order

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC ;
```

1

## ⑤ 조인을 사용한 데이터 검색

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id)
WHERE e.manager_id = 149 ;
```

⑥ 서브쿼리를 사용한 데이터 검색

```
SELECT last_name, salary
FROM employees
WHERE salary > 11000
      (SELECT salary
       FROM employees
       WHERE last_name = 'Abel');
```

2) INSERT 구문

[문법]

```
INSERT INTO table [(column [, column...])]
VALUES (value [, value...]);
```

[예제]

```
INSERT INTO departments(department_id,
                        department_name, manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);
```

1 rows inserted

3) UPDATE 구문

[문법]

```
UPDATE table
SET column = value [, column = value, ...]
[WHERE condition];
```

[예제]

```
UPDATE employees
SET department_id = 50
WHERE employee_id = 113;
```

1 rows updated

```
UPDATE copy_emp
SET department_id = 110;
```

22 rows updated



#### 4) DELETE 구문

[문법]

```
DELETE [FROM]   table
[WHERE          condition];
```

[예제]

```
DELETE FROM departments
WHERE department_name = 'Finance';
```

1 rows deleted

```
DELETE FROM copy_emp;
```

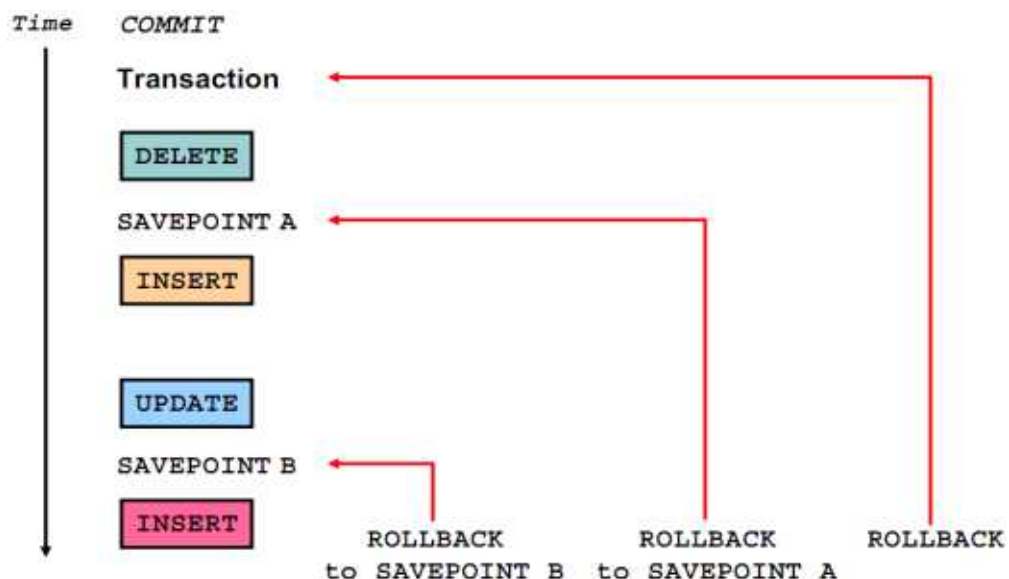
22 rows deleted

#### 5) 트랜잭션 제어 명령어 - COMMIT, ROLLBACK

- ① 트랜잭션 구성
- 여러 DML 구문이 하나의 트랜잭션
  - 하나의 DDL 구문이 하나의 트랜잭션
  - 하나의 DCL 구문이 하나의 트랜잭션

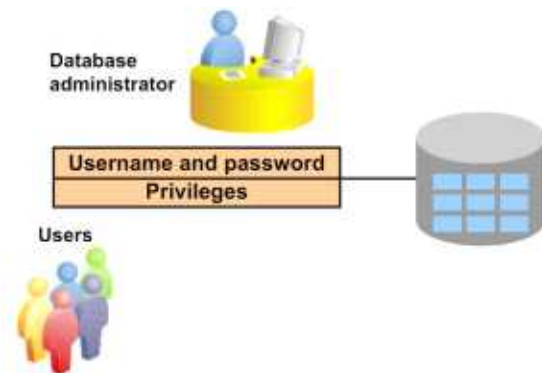
#### ② 트랜잭션 시작 및 종료

- 시작 : 첫 번째 DML 구문 실행 시
- 종료 : COMMIT 또는 ROLLBACK 실행 시  
DDL 또는 DCL 구문 실행 시(autocommit)  
SQL \*PLUS 정상 종료 시 (autocommit)  
시스템 장애 시(autorollback)



#### (4) 데이터 제어어(DCL)

##### 1) User 생성



```
SQL> conn sys/oracle as
sysdba
SQL> create user demo
        identified by demo;
SQL> conn demo/demo
=> error!(DB에 접속할 수 있는
권한 부족)
```

##### 2) 권한 유형

권한 유형	개념	소유자
System 권한	DB를 조작할 수 있는 권한	DBA
Object 권한	Object를 조작할 수 있는 권한	Object 소유자

##### 3) System 권한

- DB를 조작할 수 있는 권한, System 권한의 종류가 100개 정도 됨.
- 모든 System 권한은 DBA가 소유함.

###### ① System 권한 부여

```
GRANT create session, create table,
create sequence, create view
TO demo;
```

CONNECT demo/demo => demo 접속 가능 여부 확인해보기!

###### ② System 권한 회수

```
REVOKE create table
FROM demo;
```



#### 4) Object 권한

- Object를 조작할 수 있는 권한, Object 권한의 종류 다양함.
- 모든 Object 권한은 Object의 주인이 소유함.

##### ① Object 권한 부여

```
SQL> conn demo/demo
SQL> select employee_id, last_name, salary
      from hr.employees;                      => error! (권한부족)

SQL> conn hr/hr
SQL> GRANT select
      ON employees
      TO demo;

SQL> conn demo/demo
SQL> SQL> select employee_id, last_name, salary
      from hr.employees;
```

##### ② Object 권한 회수

```
SQL> conn hr/hr
SQL> REVOKE select
      ON employees
      FROM demo;
```

## **제3장**

# **네트워크 기초 활용**



## 1. 네트워크 들어가며

### (1) 네트워킹이란 ?

- 간단히 서로 연결.
- 정보를 공유하거나, 자원을 공유하기위해.
- 처음에는 비싼 장치에 연결 하다가 이제는 프린터, 복사기, 사물 인터넷까지 연결.
- 장비들이 대화 하는 것.
- 인터넷 : TCP/IP 라는 통신 프로토콜을 이용한 대화
- 인트라넷 : 인터넷 프로토콜을 이용한 폐쇄적 LAN

### 1) LAN

- 근거리 통신망, 로컬 영역 네트워크(영어: local area network, LAN), 구내 정보통신망은 네트워크 매체를 이용하여 집 사무실, 학교 등의 건물과 같은 가까운 지역을 한데 묶는 컴퓨터 네트워크

### 2) 이더넷

- 이더넷(영어: Ethernet)은 컴퓨터 네트워크 기술의 하나로, 전 세계의 사무실이나 가정에서 일반적으로 사용되는 LAN에서 가장 많이 활용되는 기술규격. '이더넷'이라는 명칭(이름)은 빛의 매질로 여겨졌던 에테르(ether)에서 유래되었다.

### 3) 인터넷

- 인터넷이란 이름은 1973년 TCP/IP를 정립한 빈튼 서프와 밥간이 네트워크의 '네트워크'를 구현하여 모든 컴퓨터를 하나의 통신망 안에 연결 (International Network)하고자 하는 의도에서 이를 줄여 인터넷(Internet)이라고 처음 명명하였는데 어원을 두고 있다. 이후 인터넷은 "정보의 바다"라고 불리면서 컴퓨터가 서버와 클라이언트로 연결되어 TCP/IP를 이용해 정보를 주고받게 되었다.

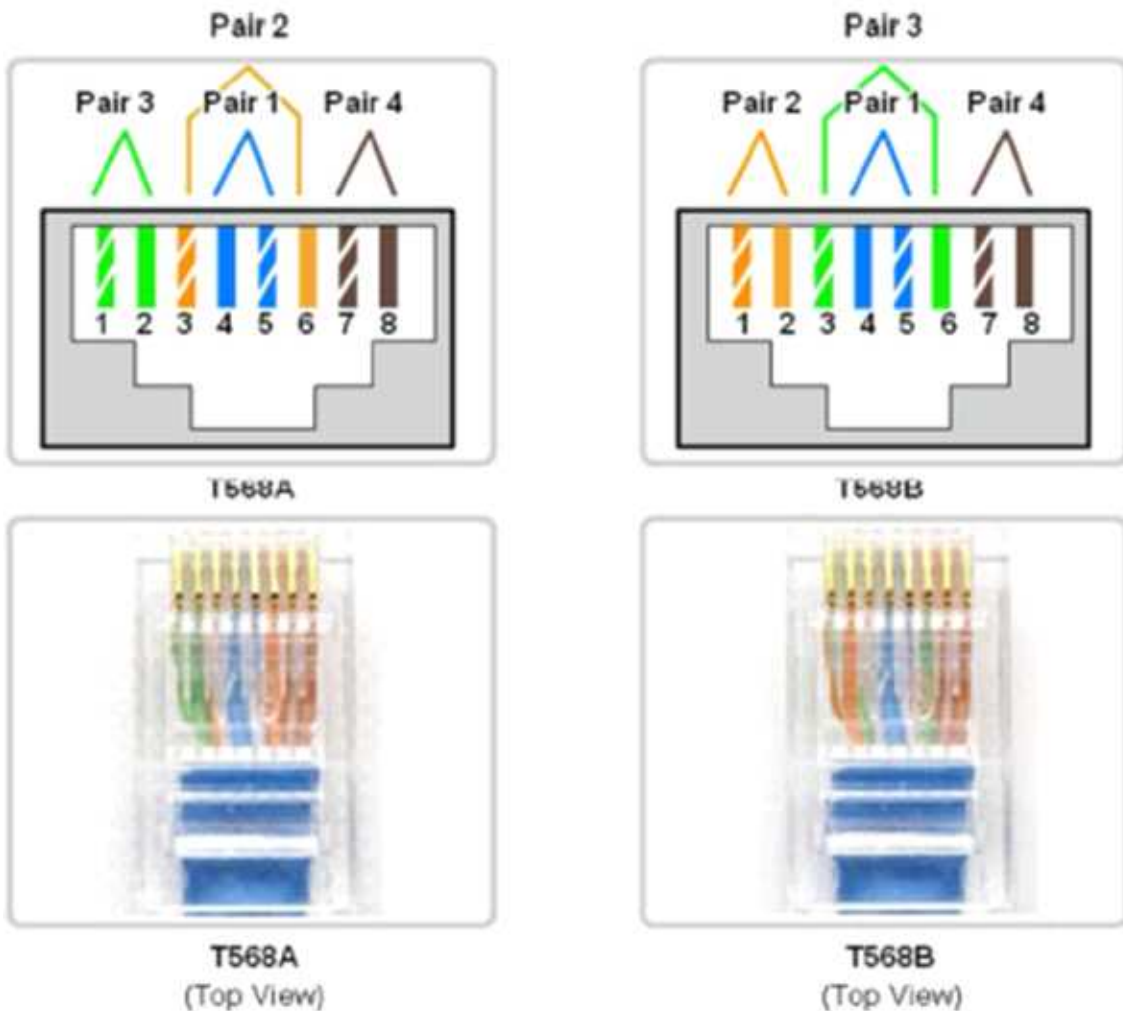
## 2. Cable

### (1) UTP Cable

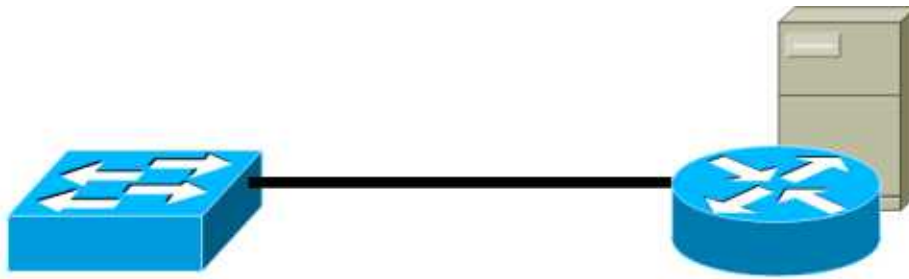
- 비차폐 연선은 절드가 없는 연선 UTP(Unshielded Twisted Pair)라고 한다. 전화선이나 이더넷 등에 많이 쓴다. 처리가 간단하고 값이 싸서 빠른 전송이 필요 없는 이더넷의 랜용도에 표준으로 쓰이고 있다.

황띠, 황, 초띠, 파, 파띠, 초, 갈띠, 갈  
1 2 3 4 5 6 7 8

### RJ45 T568A & T568B Termination

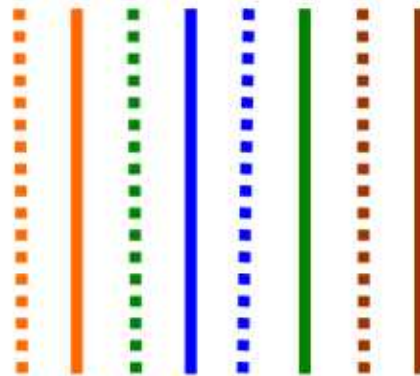
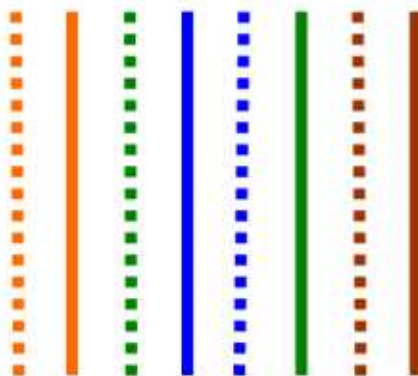
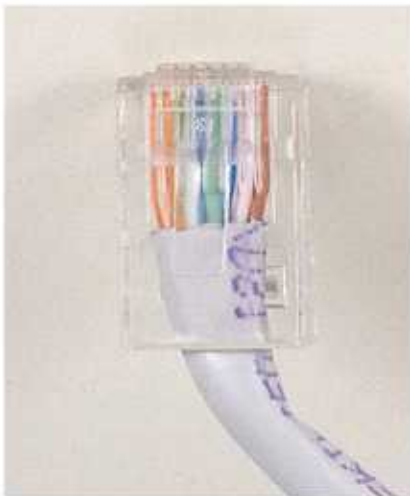


(2) Straight-Through Cable (Direct)



**Hub/Switch**

**Server/Router/PC**

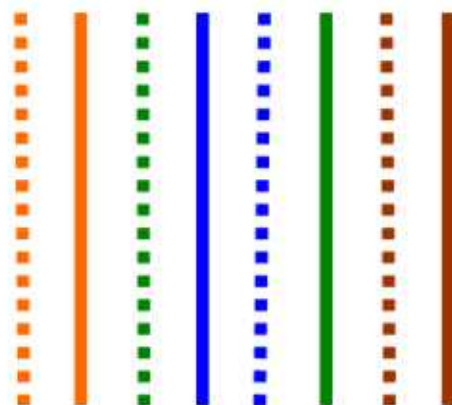
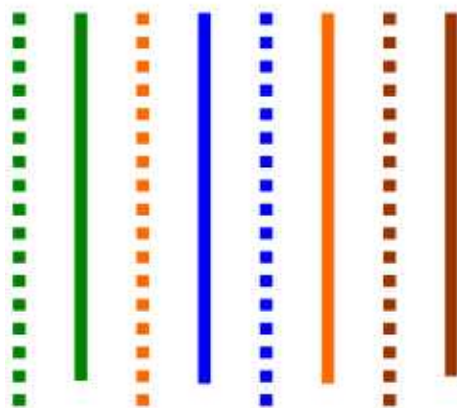
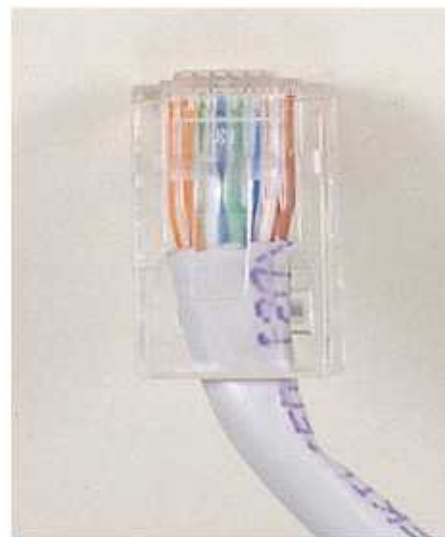


(3) Crossover Cable



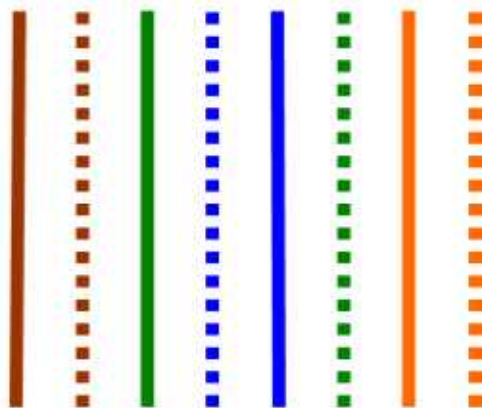
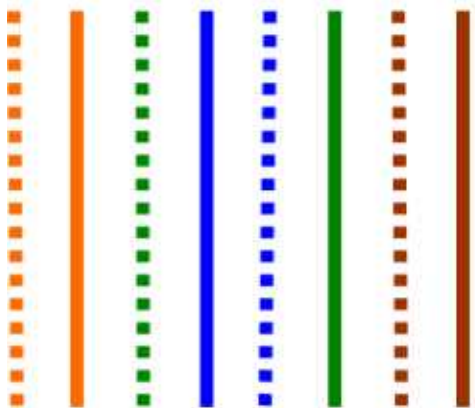
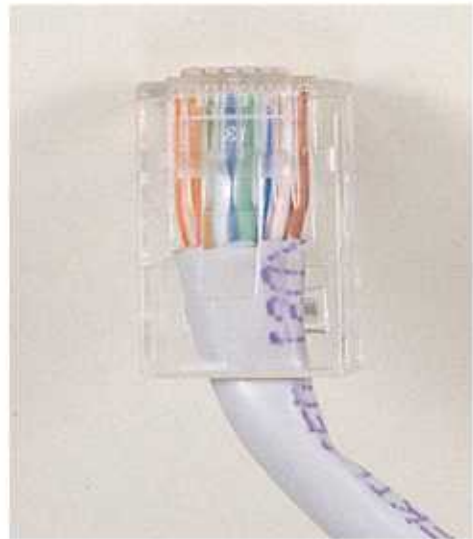
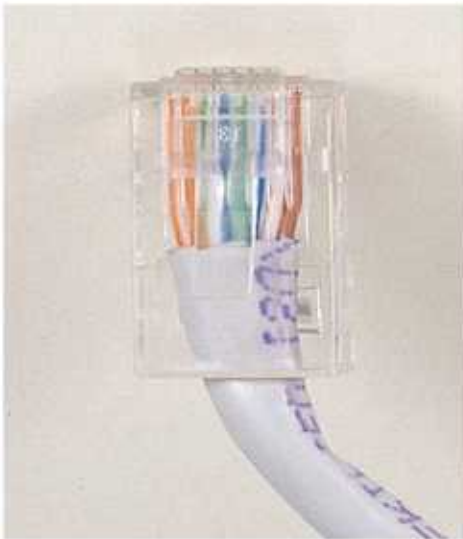
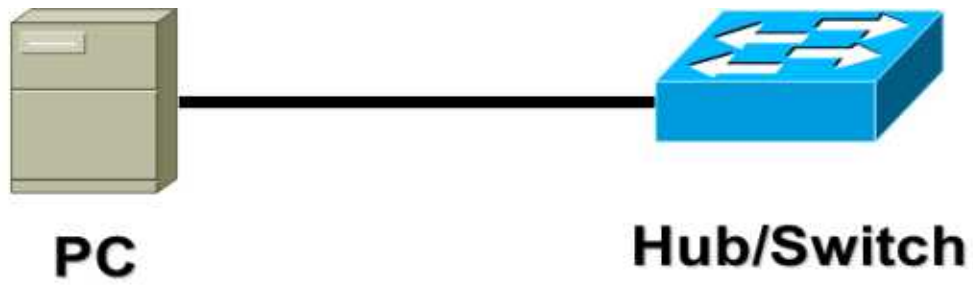
**Hub/Switch**

**Hub/Switch**





(4) Rollover Cable(console)



### 3. 네트워크 장비

#### (1) 네트워크 스위치

- 네트워크 스위치(network switch)는 처리 가능한 패킷의 숫자가 큰 것으로, 네트워크 단위 들을 연결하는 통신 장비로서 소규모 통신을 위한 허브보다 전송 속도가 개선된 것이다. 간단히 스위치라고 부르는 경우가 많으며, MAC 브리지, 스위칭 허브(switching hub), 포트 스위칭 허브(port switching hub)라고도 한다.



#### (2) 네트워크 라우터

- 라우터(router:경로기) 혹은 라우팅 기능을 갖는 공유기는 패킷의 위치를 추출하여 그 위치에 대한 최적의 경로를 지정하며, 이 경로를 따라 데이터 패킷을 다음 장치로 전향시키는 장치이다. 이때 최적의 경로는 일반적으로는 가장 빠르게 통신이 가능한 경로이므로, 이것이 최단 거리 일수도 있지만, 돌아가는 경로라도 고속의 전송로를 통하여 전달이 되는 경로가 될 수 있다. 간단히 말해 서로 다른 네트워크 간에 중계역할을 해준다.



## 4. OSI 7계층

(1) OSI 7 Layer (Open System Interconnection)

- 국제 표준 기관(ISO)으로써 개방형 통신을 할 수 있게 OSI 7 Layer를 만들었다.



(2) Encapsulation 및 Decapsulation 과정



(3) L1, L2

1) L1

- 데이터 전송을 위한 통신 케이블선입니다. 수많은 PC에 연결을 하려면 허브나 리피터를 이용한다.

2) L2

- 1계층과 연결을 통해 데이터를 전송하는 전송로 역할 LLC 부계층과 MAC 부계층 두가지로 나뉘는데 계층의 MAC부계층은 맥주소 (PC식별을 위한 고유 물리적 번호)가 있어 내부통신(케이블 선으로 연결되어 있는 PC들이) 가능합니다. LLC부계층은 3계층과 연결되는데 외부통신 (케이블 선을 통해 다른 장소의 PC)을 합니다. 결국 LLC 부계층은 여러 상이한AC(Media Access Control) 부계층 프로토콜을 사용할 수 있게 하여 망의 토폴로지에 관계없는 통신이 가능하도록 한다.(★)

(4) L3,L4,L5,L6,L7

1) L3

- 외부로 나가는 통신은 3계층을 통해 최적의 경로를 선택받는다. 라우터를 이용하여 서로 다른 네트워크를 중계해주는 역할을 합니다. 서로 다른 네트워크를 연결해주는데 라우터와 라우터간의 가는 여러 가지 길을 최적의 경로로 연결해 줍니다.

2) L4

- 전송 시 오류발생하면오류가난 부분만 재전송을 하는 데이터 전송을 보장하는 계층입니다.

3) L5

- 논리적인 연결 서비스 제공 full-duplex(동시 양방향), Half-duplex (한 번에 한 방향), Simplex(한쪽으로만 통신)

4) L6

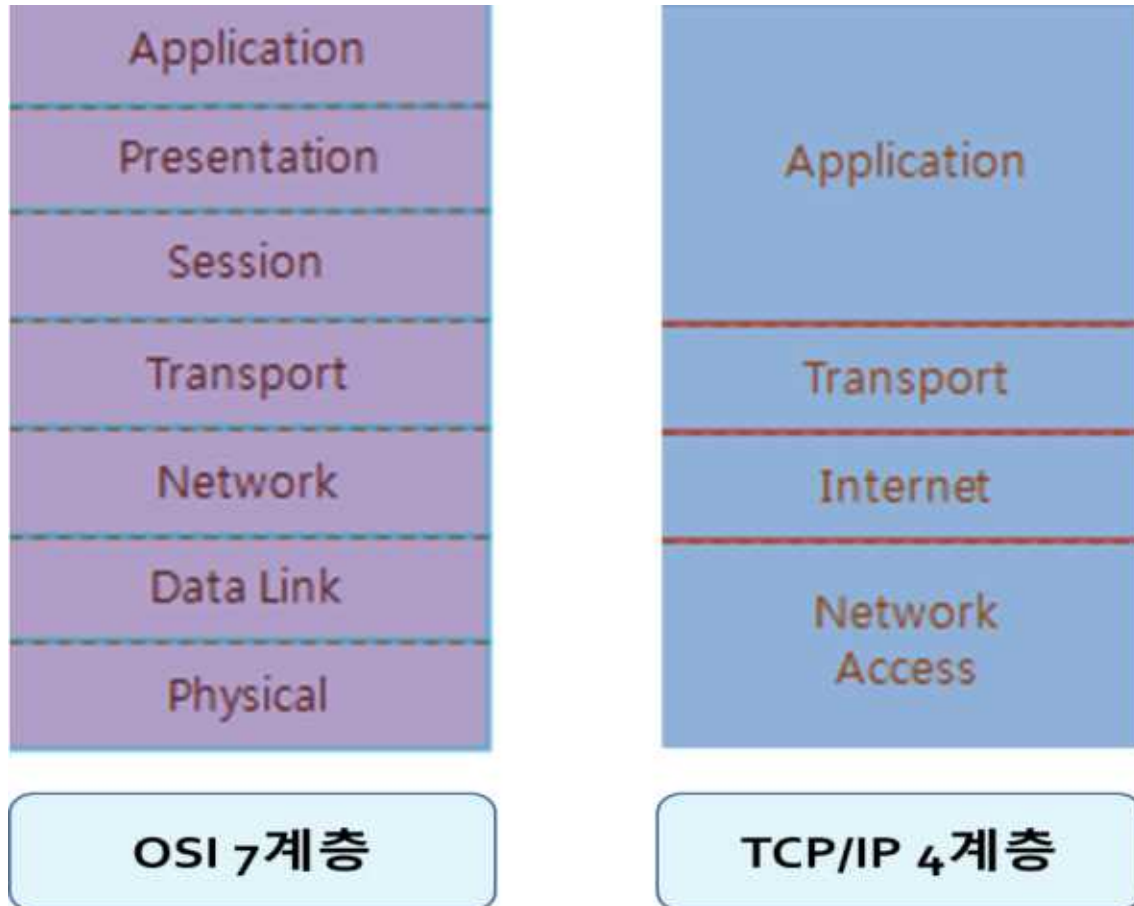
- 데이터를 암호화 혹은 압축하는 계층입니다.

5) L7

- 사용자 인터페이스 계층입니다.

## 5. TCP/IP

(1) TCP / IP 4계층



- L1 : 실제 물리적인 장치에 의해 전달
- L2 : 네트워크 디바이스 드라이버로 구성
- L3 L4 : 주로 OS 로 구성
- L5 L6 L7 : 응용 프로그램에 의해 구성되는 정보



## 6. IP 기초

### (1) IP 주소

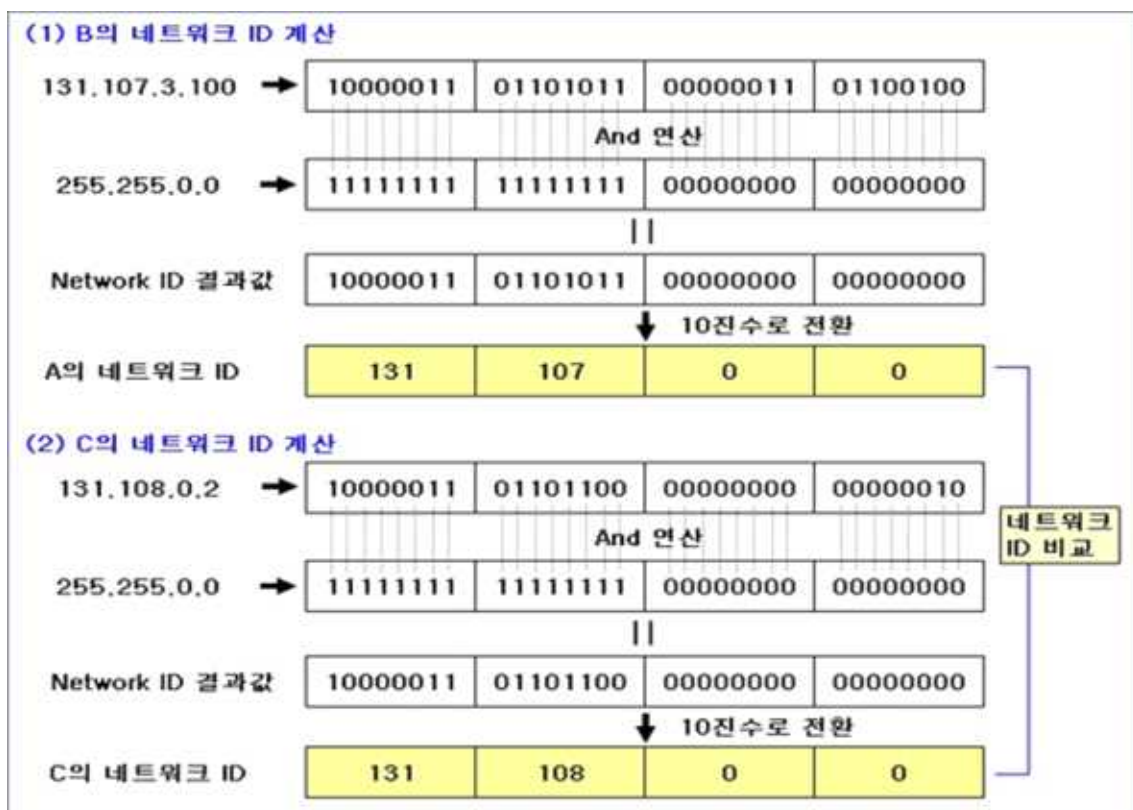
- IP 주소(Internet Protocol address, 인터넷규약주소)는 컴퓨터 네트워크에서 장치들이 서로를 인식하고 통신을 하기 위해서 사용하는 특수한 번호이다. 네트워크에 연결된 장치가 라우터이든 일반 서버이든, 모든 기계는 이 특수한 번호를 가지고 있어야 한다. 이 번호를 이용하여 발신자를 대신하여 메시지가 전송되고 수신자를 향하여 예정된 목적지로 전달된다. IP 주소를 줄여서 IP라고 부르기도 하나 IP는 인터넷 규약 자체를 가리키는 말이기 때문에 엄밀하게는 구별해야 한다.

### (2) 사설 IP 주소

- 인터넷에서 공인된 IP 주소를 사용하지 않고, 사적인 용도로 임의 사용하는 IP 주소
- Class A 규모 10.0.0.0~10.255.255.255 (10.0.0.0/8)
- Class B 규모 172.16.0.0~172.31.255.255 (172.16.0.0/12)
- Class C 규모 192.168.0.0~192.168.255.255 (192.168.0.0/16)

### (3) Subnet Mask

- 서브 네트워크를 만들기 위해 AND 비트 연산에 의해 씌우는 마스크  
TCP/IP 프로토콜에서 IP 주소체계로 네트워크를 나누는 (분할하는) 논리적인 수단



(4) Default Gateway

- 호스트가TCP/IP통신을 할 때 가장 먼저 목적지 호스트가 자신과 같은 로컬에 있는지 원격지에 있는지를 판단한다고 했다. 이때 원격지에 있는 결과가 나오면 컴퓨터는 Default Gateway를 이용해서 통신을 하게 된다.

(5) 십진수 이진수 변환

$$\text{Ex) } 127 = \underset{\text{백}}{1 \times 10^2} + \underset{\text{십}}{2 \times 10^1} + \underset{\text{일}}{7 \times 10^0}$$

$$2^7=128, 2^6=64, 2^5=32, 2^4=16, 2^3=8, 2^2=4, 2^1=2, 2^0=1$$

이진수로 나타내기 위한 재료.

127를 위의 재료로만 구성 해보자.

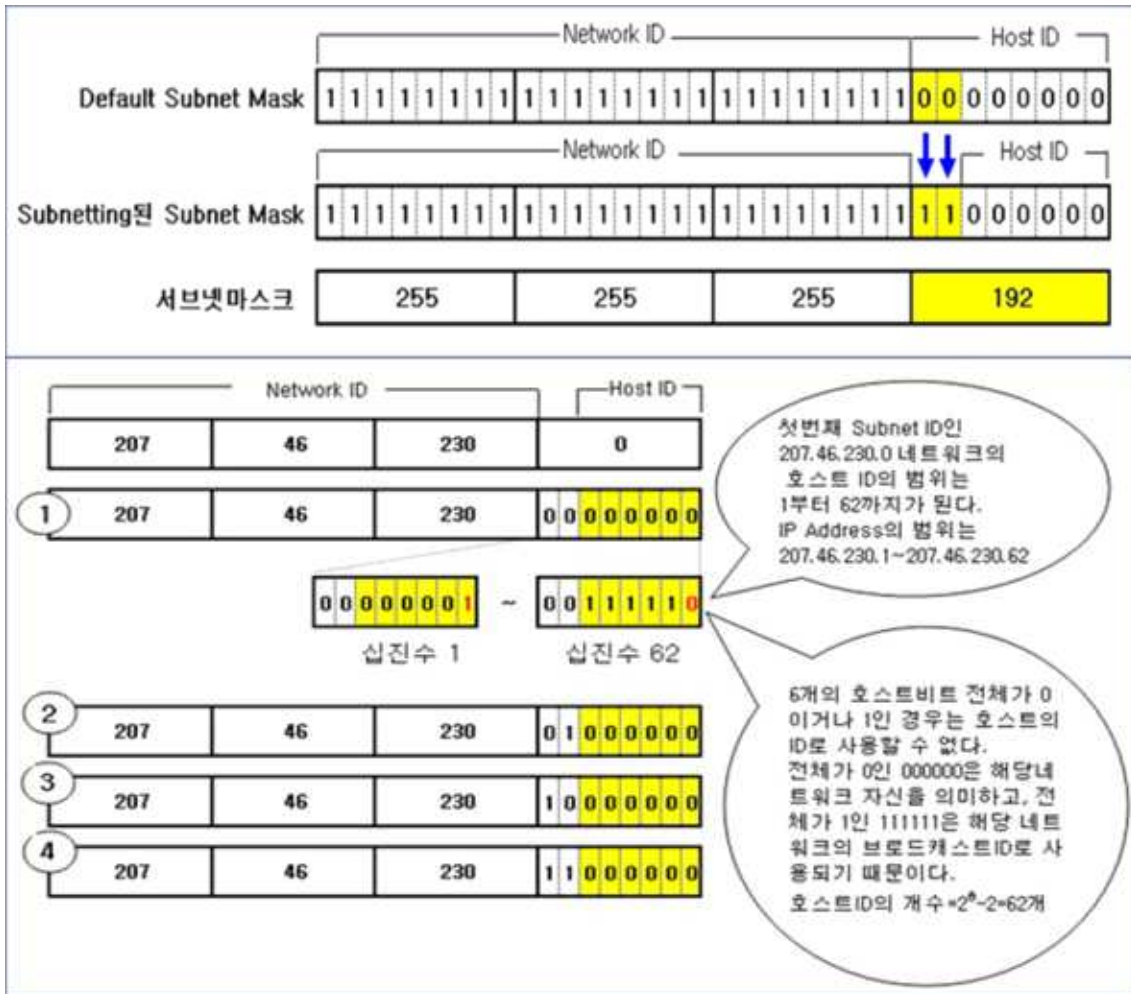
$$\begin{aligned} 127 &= 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 \\ &= 64 + 32 + 16 + 8 + 4 + 2 + 1 \end{aligned}$$

$$127 = 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$$



## (6) Subnetting

- IP 주소를 여러 개의 네트워크로 분리



- ① 회사에서 필요로 하는 네트워크의 수 결정
- ② 필요한 네트워크ID를 만들기 위해 전환할 bit수의 결정
- ③ 서브넷 마스크(User defined Subnet Mask) 계산
- ④ 서브넷 ID 계산
- ⑤ 서브넷별 호스트ID의 범위 계산

(7) CIDR(Classless Inter Domain Routing) = Subnetting + Supernetting

CIDR 표기법: 207.46.230.2/26

- 1) 사이더(Classless Inter-Domain Routing, CIDR)는 클래스 없는 도메인 간 라우팅 기법으로 1993년 도입되기 시작한, 최신의 IP 주소 할당 방법이다. 사이더는 기존의 IP 주소 할당 방식이었던 네트워크 클래스를 대체하였다. 사이더는 IP 주소의 영역을 여러 네트워크 영역으로 나눌 때 기존방식에 비해 유연성을 더해준다. IPv4 사이더 블록은 IPv4 주소와 비슷한 형태를 지니며, 점과 숫자로 이루어진 4부분의 주소와 '/'뒤의 0에서 32까지의 숫자로 이루어진다. 즉,  $A.B.C.D/N$ 과 같은 형태이다.

(8) 서브네팅 예제

- Ex) 192.168.1.0/24 4개의 네트워크 영역으로 분할. 1. 총무, 2 영업, 3. 기술, 4 경영  
전체 총 갯수 : 256개 4등분 -> 64개, 64개, 64개, 64개

풀이)

- 1)  $2^x \geq y$

y: 필요한 네트워크 영역 : 4

x: 가용 비트의 수: 2비트

- 2) 기존 서브네파스크에서

-> 서브네팅이 된 서브네파스크로 변환

호스트영역의 왼쪽에서부터 가용 비트만큼 1로 전환

255.255.255. 00000000 : 기존 C Class 서브네파스크

(255.255.255.: 네트워크 영역)

(00000000 : 호스트 영역)

255.255.255.11 000000

:255.255.255.192 :서브네팅이 된 서브네파스크

### 3) 주어진 아이피를 가용한 범위의 네트워크로 나누기

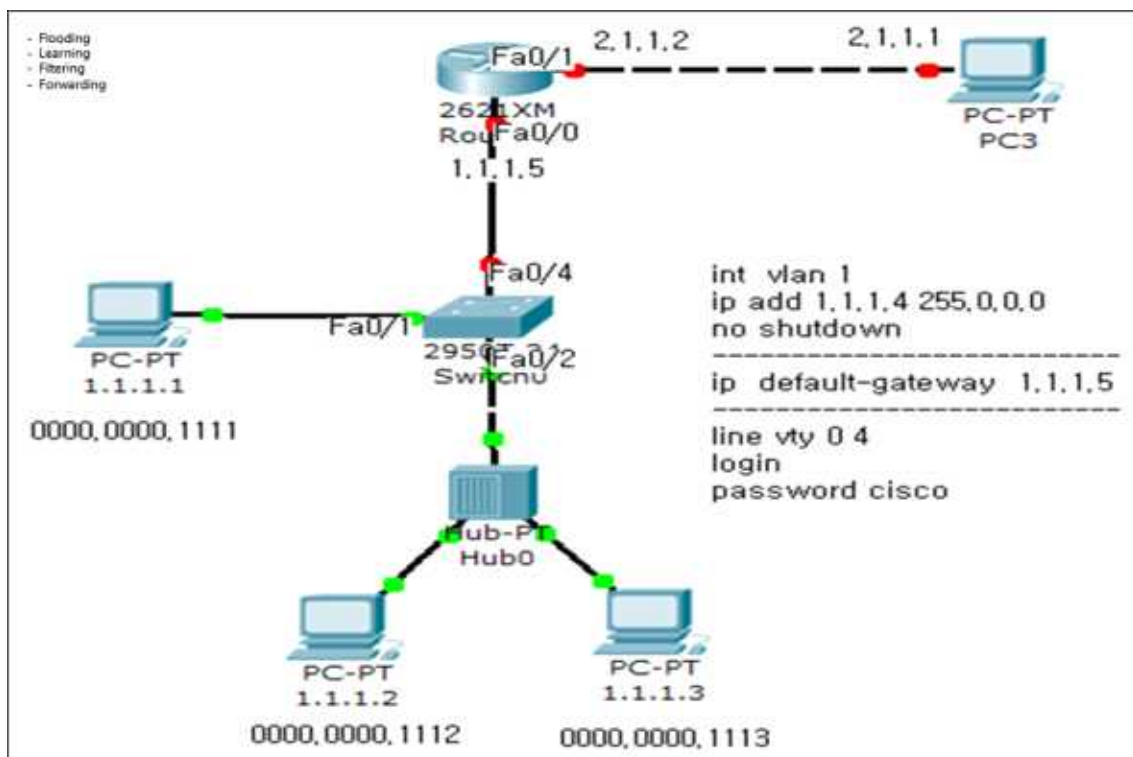
192.168.1.0/24		호스트영역	
192.168.1.	00	000000(0)	:네트워크 주소
192.168.1.	00	111111(63)	:브로드캐스트 주소
---		-----	
192.168.1.	01	000000(64)	:네트워크 주소
192.168.1.	01	111111(127)	:브로드캐스트 주소
192.168.1.	10	000000(128)	:네트워크 주소
192.168.1.	10	111111(191)	:브로드캐스트 주소
192.168.1.	11	000000(192)	:네트워크 주소
192.168.1.	11	111111(255)	:브로드캐스트 주소

네 개의 네트워크 주소를 구했습니다.

## 7. Packet Tracer 가상 환경 구성

### (1) 패킷 트레이서

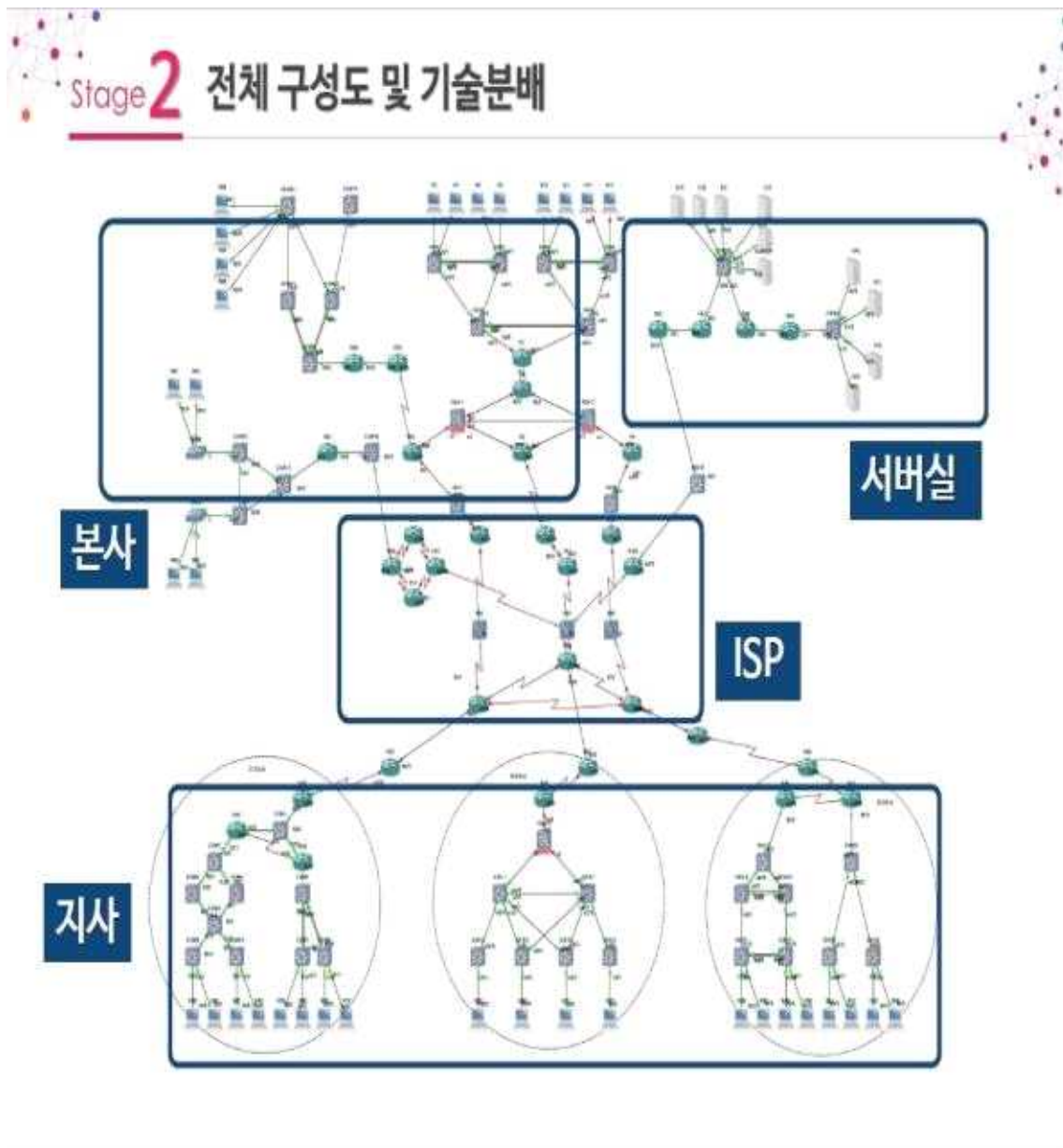
- 패킷 트레이서(Packet Tracer)는 교육과 실습, 그리고 단순 컴퓨터 네트워크 시뮬레이션을 위한 연구에 이용할 수 있는 시스코의 라우터 시뮬레이터이다. 이 도구는 시스코 시스템즈가 제작하였으며 시스코 네트워킹 아카데미에 참여하고 있거나 참여한 적이 있는 강사와 학생들에게 무료로 배포 제공된다. 패킷 트레이서의 목적은 학생과 교사들이 네트워킹의 원칙을 학습하고 시스코 기술의 특정 능력을 계발하는 도구를 제공하는 것이다.



## 8. Routing

### (1) 라우팅

- 라우팅(영어: routing)은 어떤 네트워크 안에서 통신 데이터를 보낼 경로를 선택하는 과정이다. 라우팅은 전화 통신망, 전자 정보 통신망(인터넷 같은) 그리고 교통망 등 여러 종류의 네트워크에서 사용된다. 주로 패킷 스위칭 기술을 이용한 컴퓨터 네트워크에서의 라우팅에 대해 주로 기술함.
- 패킷 스위칭 네트워크에서 라우팅은 패킷 전달 및 일반적으로 라우터, 브릿지, 게이트웨이, 방화벽 또는 스위치로 불리는 중간 노드를 거쳐 출발지부터 최종 목적지까지 논리적으로 주소가 부여된 패킷의 변환을 총괄



(2) 윈도우에서 라우팅 테이블 확인

 선택 C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.17134.228]  
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\itwill-cisco1205>netstat

활성 연결

프로토콜	로컬 주소	외부 주소	상태
TCP	1.252.62. :49714	52.230.3.194:https	ESTABLISHED
TCP	1.252.62. :49754	111.91.128.208:http	ESTABLISHED
TCP	1.252.62. :49755	17.248.161.167:https	CLOSE_WAIT
TCP	1.252.62. :49774	17.252.236.141:5223	ESTABLISHED
TCP	1.252.62. :49791	17.248.161.171:https	CLOSE_WAIT
TCP	1.252.62. :49792	a104-74-154-203:https	CLOSE_WAIT
TCP	1.252.62. :49793	17.248.161.47:https	CLOSE_WAIT
TCP	1.252.62. :49794	17.248.161.44:https	CLOSE_WAIT
TCP	1.252.62. :49797	17.248.161.171:https	CLOSE_WAIT
TCP	1.252.62. :49800	a104-74-154-203:https	CLOSE_WAIT
TCP	1.252.62. :49801	17.248.161.47:https	CLOSE_WAIT
TCP	1.252.62. :49803	17.248.161.44:https	CLOSE_WAIT
TCP	1.252.62. :49804	17.248.161.47:https	CLOSE_WAIT
TCP	1.252.62. :49806	120.50.133.195:5004	ESTABLISHED
TCP	1.252.62. :49977	ec2-54-88-142-229:https	ESTABLISHED
TCP	1.252.62. :49978	ec2-34-207-3-244:https	ESTABLISHED



## 9. DHCP

### (1) DHCP

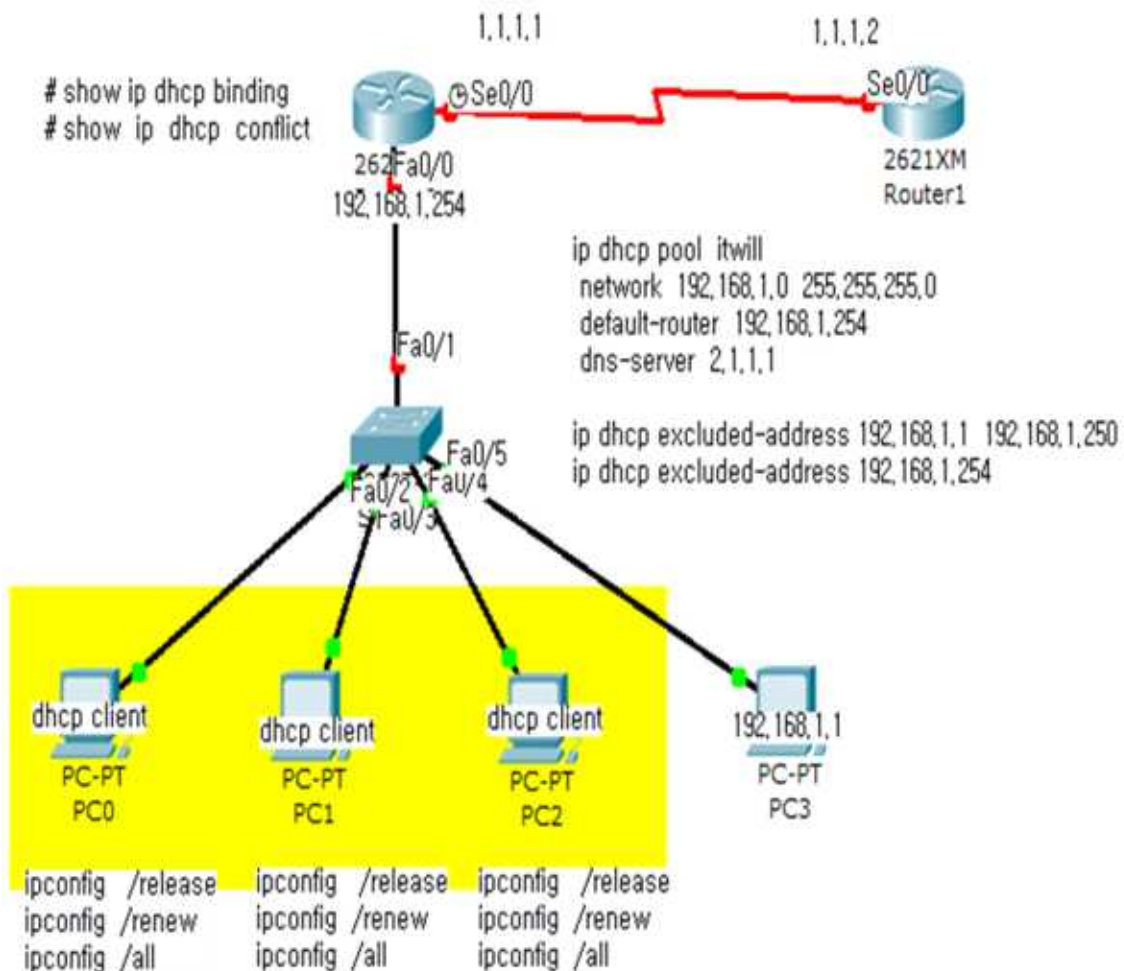
(Dynamic Host Configuration Protocol. )

(Discover, Offer, Request, Acknowledgement)

#### 1) DORA 프로세스

- Discover  
패킷 수신 대기할 DHCP서버 찾기 위해 전송.
- Offer  
패킷 서버가 클라이언트에게 서비스를 제공하기 위해 전송한다.
- Request  
패킷 클라이언트가 서버로부터 offer를 받으면 DHCP request 패킷을 받는다.
- Acknowledgement  
패킷 서버는 요청된 IP 주소를 확인 패킷으로 클라이언트에 전송하고 그 정보를 데이터베이스에 기록한다.

Discover -> Offer -> Request -> Ack  
= broadcast (255.255.255.255)



## 10. NAT

(1) NAT(네트워크 주소 변환) Network address translation

- IP패킷의 TCP/UDP 포트숫자와 소스 및 목적지의 IP 주소 등을 재기록하면서 라우터를 통해 네트워크 트래픽을 주고받는 기술을 말한다.
- 패킷에 변화가 생기기 때문에 IP나 TCP/UDP의 체크섬(checksum)도 다시 계산되어 재기록 해야 한다.
- NAT를 이용하는 이유는 대개 사설 네트워크에 속한 여러 개의 호스트가 하나의 공인 IP 주소를 사용하여 인터넷 접속하기 위함이다.
- NAT가 호스트 간의 통신에 있어서 복잡성을 증가시킬 수 있으므로 네트워크 성능에 영향을 줄 수 있는 것은 당연하다.

(2) Oracle VM VirtualBox

1) NAT 포트 포워딩 예제





## **제4장**

# **기본 개발환경 구축**



## 1. 기본 개발환경 구축하기

### (1) 운영체제 설치

- 윈도우 CD, USB 준비
- 부팅순서 CD, USB 설정
- 윈도우 파티션 설정 후 설치

### (2) 개발도구 설치

#### 1) JDK 설치

- <https://www.oracle.com/technetwork/java/index.html>
- Download
- 설치 : exe파일 실행
- 환경설정 : 제어판 - 시스템 - 고급시스템 설정- 환경변수
  - 시스템변수
  - JAVA\_HOME 설정
  - path - 편집 - 새로 만들기 - 경로 설정
  - CLASSPATH 경로 설정

```
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

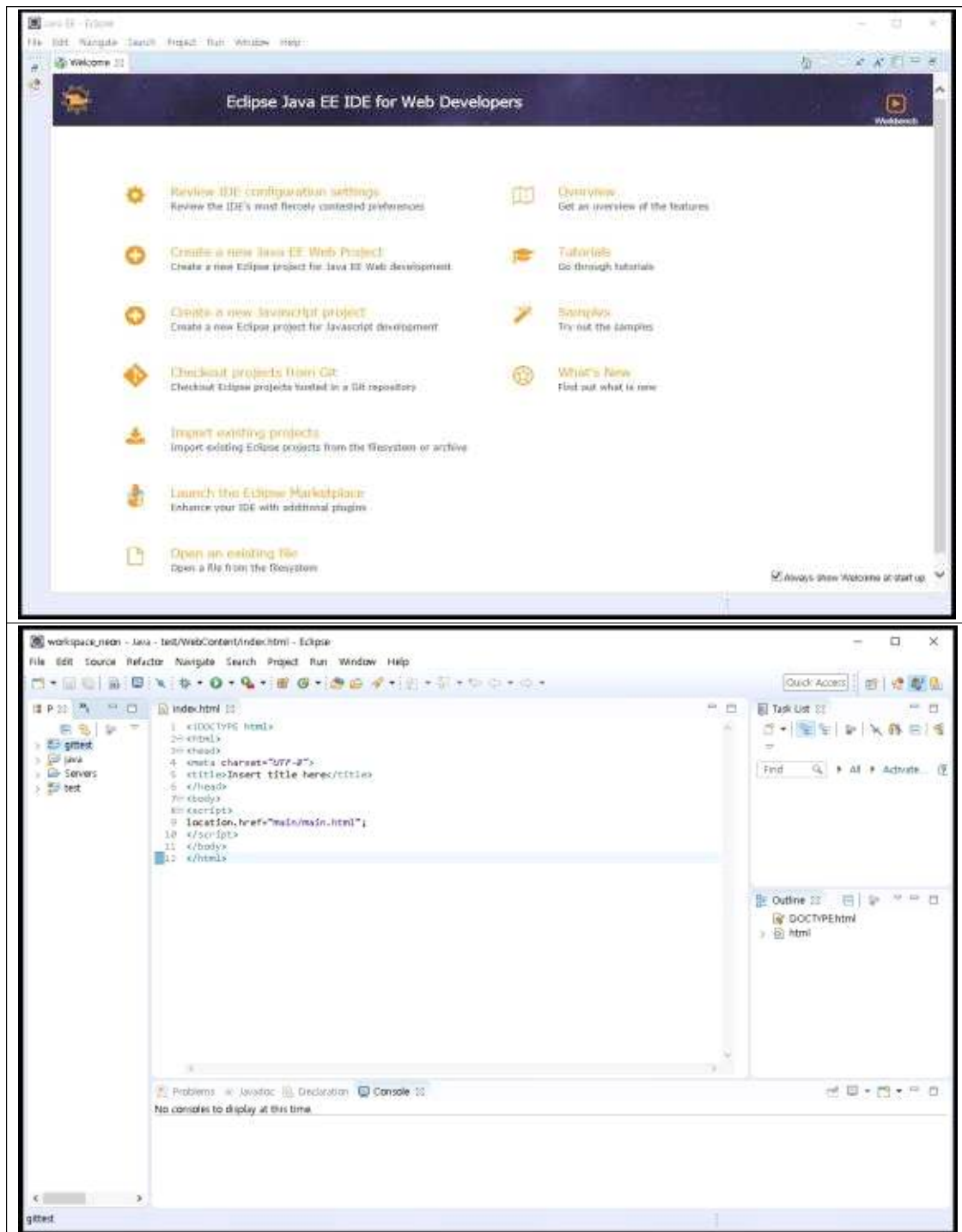
C:\Users\User>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)

C:\Users\User>javac -version
javac 1.8.0_101

C:\Users\User>
```

## 2) Eclipse 설치

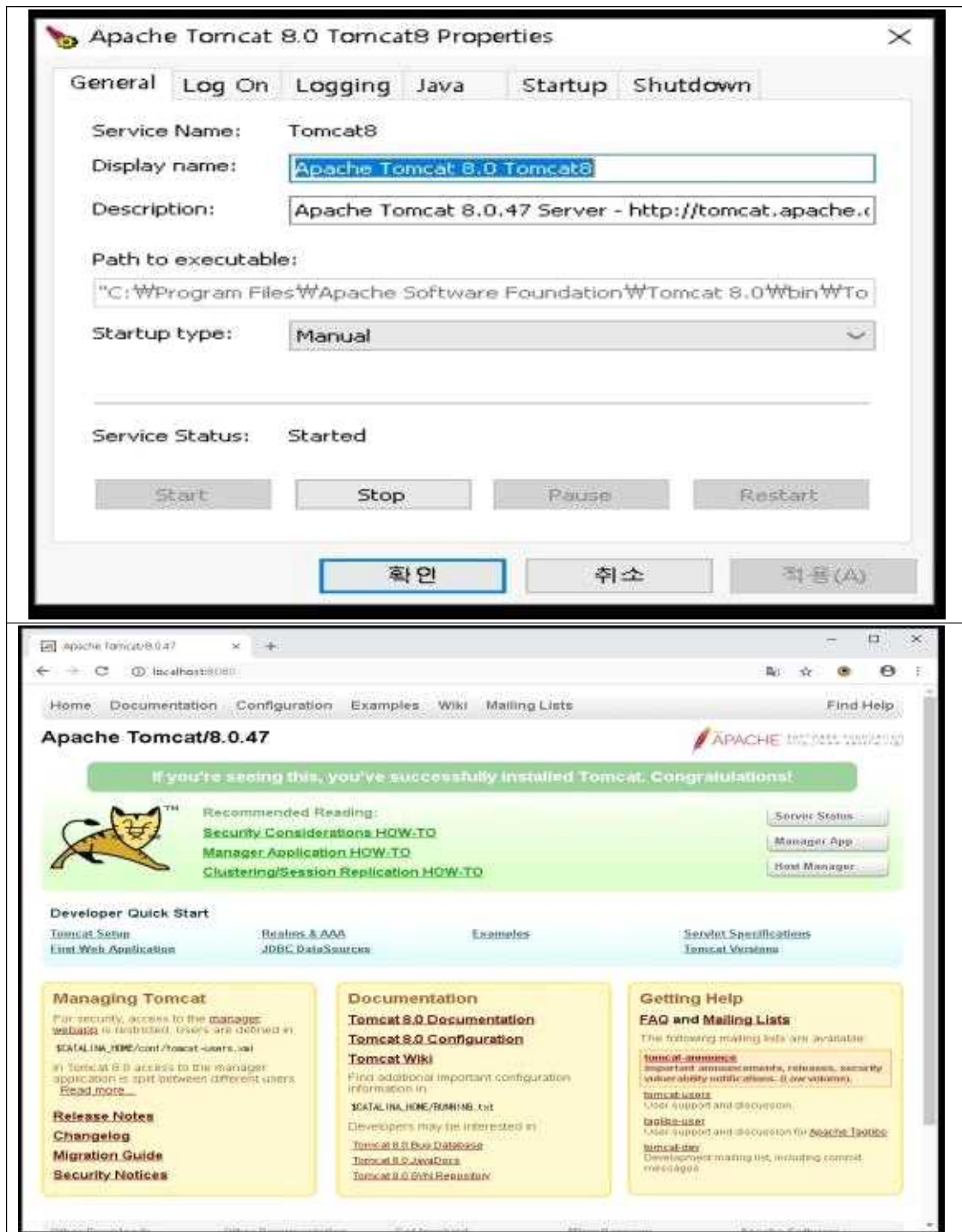
- <http://www.eclipse.org/>
- Download
- 설치 : zip파일 압축풀기, exe파일 실행
- 설정 : Server 설정, 프로젝트 생성



### 3) 서버 설치

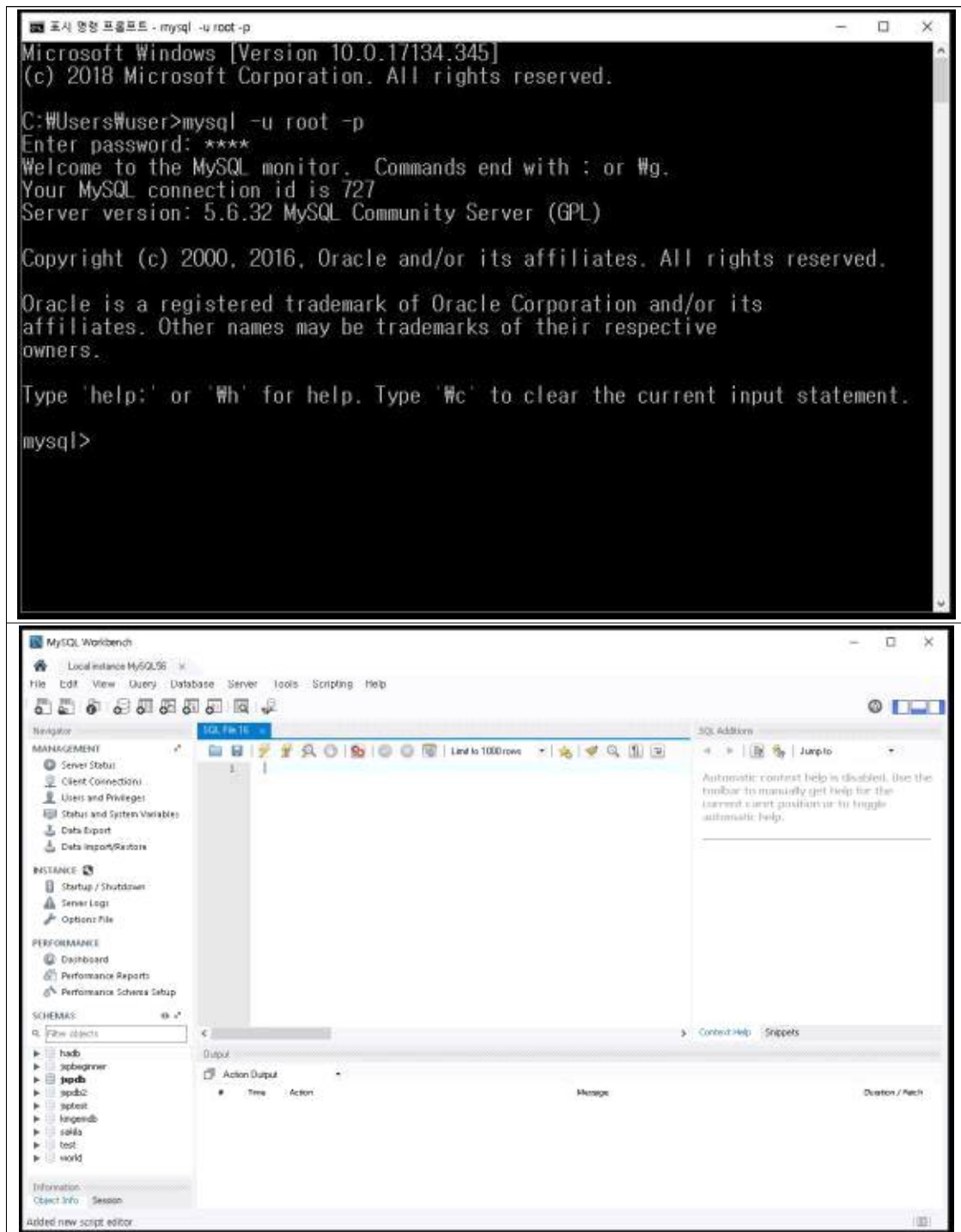
#### ① 웹 서버 설치

- <http://tomcat.apache.org/>
- Download
- 설치 : zip파일 압축풀기, exe파일 실행
- 서버 Start



## ② DB서버 설치

- <https://www.mysql.com/>
- Download MySQL Community Server
- 설치 : mysql-installer-web-community.msi 파일 실행
- 환경설정 : 제어판 - 시스템 - 고급시스템 설정- 환경변수
  - 시스템변수 - path - 편집 - 새로 만들기



## 2. 컴퓨터 시스템의 구성요소

### (1) 하드웨어

- 컴퓨터를 구성하는 기기 장치
- 컴퓨터 시스템을 구성하는 장치로  
중앙처리장치(CPU), 저장장치, 출력 장치 등으로 구성

#### 1) 중앙처리장치(CPU)

- 명령어를 해독하고 실행하는 장치
- 컴퓨터 두뇌에 해당되며 제어장치, 연산장치(ALU), 레지스터로 구성

제어장치	<ul style="list-style-type: none"> <li>- 중앙처리장치에서 일어나는 모든 작업을 통제하고 관리</li> <li>- 프로그램 명령을 해독하여 처리하도록 지시, 관리</li> </ul>
연산장치 (ALU)	<ul style="list-style-type: none"> <li>- 제어장치의 신호에 따라 산술 연산과 논리 연산 수행</li> <li>- 산술, 논리 연산 등과 같이 계산 기능을 수행</li> </ul>
레지스터	<ul style="list-style-type: none"> <li>- 중앙처리장치에서 명령어를 수행하는 동안 필요한 정보 저장</li> <li>- CPU내에 존재하는 임시기억장소</li> </ul>

#### 2) 저장장치

- 컴퓨터에 필요한 정보를 저장하는 장치
- CPU가 사용하는 주기억장치와 주기억장치를 보조하는 보조 기억장치로 구성

캐시 기억장치	중앙처리장치가 자주 필요로 하는 프로그램의 일부와 데이터를 저장
주기억장치	<p>CPU가 프로그램 수행 시 직접 접근하여 데이터를 읽거나 쓰기 위해서 사용하는 메모리로, CPU가 수행할 명령어와 데이터를 저장하는 메모리</p> <ul style="list-style-type: none"> <li>- 램 : 실행 중인 프로그램과 실행에 필요한 데이터를 일시적으로 저장</li> </ul>

주 기억장치	SRAM (Static RAM)	<ul style="list-style-type: none"> <li>- 전원이 공급되는 동안 저장된 데이터가 유지되는 메모리</li> <li>- DRAM에 비해 데이터 접근속도가 빠르나 가격이 고가</li> <li>- 접근속도가 빠르기 때문에 캐쉬메모리로 사용</li> <li>- 캐쉬메모리: 자주 사용되는 데이터를 저장하는 장소로 CPU가 제일 먼저 접근하는 메모리</li> </ul>
	DRAM (Dynamic RAM)	<ul style="list-style-type: none"> <li>- 전원이 공급되는 기간에도 주기적으로 재충전시켜 주어야 저장된 데이터가 유지되는 메모리</li> </ul>
	<ul style="list-style-type: none"> <li>- 롬: 부팅할 때 동작하는 바이오스 프로그램을 저장</li> </ul>	
보조 기억장치	중앙처리장치가 현재 필요로 하지 않는 많은 양의 프로그램과 데이터 저장	
	하드 디스크	<ul style="list-style-type: none"> <li>- 대용량 프로그램이나 데이터를 저장하기 위해 사용되는 보조 기억장치</li> </ul>
	광 디스크 시스템	<ul style="list-style-type: none"> <li>- 광디스크는 레이저 광선을 이용하여서 음성, 문자, 영상 데이터를 알미늄 금속성 원반에 기록</li> <li>- CD, DVD</li> </ul>
	USB 메모리	<ul style="list-style-type: none"> <li>- 플래시메모리와 USB포트가 결합한 장치</li> <li>- 휴대가 간편</li> </ul>
	메모리 카드	<ul style="list-style-type: none"> <li>- 디지털 카메라, 캠코더에 사용되는 카드 형태로 제작된 메모리</li> <li>- SD 메모리카드, 메모리스틱, CF메모리</li> </ul>



3) 입력장치

- 컴퓨터시스템 외부로부터 정보를 입력 받는 장치  
예) 마우스, 키보드, 터치패드, 광학 스캐너 등

4) 출력장치

- 컴퓨터시스템에서 처리된 결과물을 외부로 출력해주는 장치  
예) 프린터, 스피커, 모니터 등

(2) 소프트웨어

- 컴퓨터 시스템이나 주변기기 등의 하드웨어를 작동시키기 위한 모듈
- 하드웨어를 작동시키는 역할을 담당
- S/W는 펌웨어, 운영체제, 응용소프트웨어로 구성

1) 시스템 소프트웨어

펌웨어	컴퓨터 시작에 필요한 S/W로 초기 작동을 제어 - BIOS(Basic Input Output System) : Award, AMI, Phoenix
운영체제	하드웨어와 응용소프트웨어의 작동을 관리 - Windows , Linux 등
언어 번역 프로그램	사람이 이해하기 쉬운 형태로 작성된 프로그램을 컴퓨터가 이해할 수 있는 형식으로 번역 - 어셈블러, 컴파일러, 인터프리터 등
유틸리티 프로그램	운영체제의 기능을 보완해서 사용자가 컴퓨터를 편리하게 사용할 수 있도록 도와줌 - 시스템 관리 프로그램, 텍스트 편집기, 백신 프로그램 등

2) 응용소프트웨어

- 사용자가 사용하는 프로그램으로 운영체제 상에서 작동
- Internet Explorer, Microsoft Office 등

### 3. 컴퓨터 시스템의 데이터 표현

#### (1) 데이터 표현

- 컴퓨터는 문자, 정수, 실수, 그림, 소리, 동영상 등의 모든 정보를 2진수 형식으로 표현
- 컴퓨터에서는 수 체계가 2진법에 기초를 두고 있는데 이는 컴퓨터가 계산을 할 때 0과 1의 조합으로 계산을 하도록 설계
- 정보의 단위 :  
비트(Bit) - 니블(Nibble) - 바이트(Byte) - 워드(Word)

비트(Bit, Binary digit)	0 또는 1로 구성되는 정보 표현의 최소단위
니블(Nibble)	4bit 로 구성된 단위
바이트(Byte)	1byte는 8bit 로 구성 정보처리를 위해 사용되는 비트의 집합 문자를 표현하는 최소 단위
워드(Word)	컴퓨터가 한 번에 처리할 수 있는 명령 단위 연산처리의 기본단위 (2Byte, 4Byte, 8Byte) 일반적으로 32비트(4바이트)

- 기억 용량 단위 :  
KB - MB - GB - TB - PB  
킬로 - 메가 - 기가 - 테라 - 페타

#### 1) 문자표현

아스키 (ASCII)코드	- 각 문자를 7비트로 표현 - 최대 표현할 수 있는 문자수는 128(27)개
확장 아스키 (extended ASCII)코드	- 각 문자를 8비트로 표현 - 기존 7비트 아스키 코드의 가장 왼쪽에 0을 삽입 - 총 256(28)개의 문자를 표현할 수 있음
유니코드 (Unicode)	- 언어와 상관없이 모든 문자를 2Byte(16bit)로 표현 세계 통합코드 - 사용 중인 운영체제, 프로그램, 언어 에 관계없이 문자마다 고유한 코드값을 제공 - 최대 65,536(216)개의 문자를 표현할 수 있음

## 2) 정수표현

- 32비트로 정수를 표현  
: 정수를 저장하는 기억공간의 크기
- 2의 보수 2's complement 표기법

양수	<ul style="list-style-type: none"> <li>- 가장 왼쪽 비트가 0, 나머지 비트를 이용해서 해당 수를 2진수로 표현</li> <li>- 최상의 비트 0</li> </ul>
음수	<ul style="list-style-type: none"> <li>- 0은 1로, 1은 0으로 바꾼 후 1을 더함</li> <li>- 최상의 비트 1</li> </ul>

- n비트로 표현할 수 있는 수의 범위 :  $-2^{n-1} \sim 2^{n-1}-1$

## 3) 실수표현

- 지수 형식

$0.8 \times 10^{-5}$	$m \times r^e$ m: 가수 r: 밑수 e: 지수
----------------------	-------------------------------------

- 표준 실수 표현 방식

단일 정밀도 형식	<ul style="list-style-type: none"> <li>- 가수의 부호 : 양수 0, 음수 1</li> <li>- 8비트 지수의 4바이트 형식</li> </ul>
이중 정밀도 형식	<ul style="list-style-type: none"> <li>- 가수의 부호 : 양수 0, 음수 1</li> <li>- 11비트 지수의 8바이트 형식</li> </ul>

## (2) 진법 변환

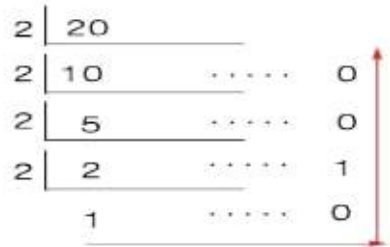
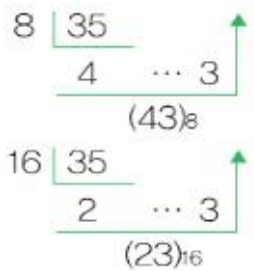
## 1) 진법

- 사용할 수 있는 숫자의 개수와 각 숫자의 위치 값을 정의한 수 체계

## 2) 진법의 종류

10진법	0에서 9까지의 10개 숫자를 사용하여 한 자리 숫자를 표현 - $365_{10}$ $= 3 \times 10^2 + 6 \times 10^1 + 5 \times 10^0$
2진법	0, 1 숫자를 사용하여 한 자리 숫자를 표현 - $1011_2$ $= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
8진법	0에서 7까지의 8개 숫자를 사용하여 한 자리 숫자를 표현 - $567_8$ $= 5 \times 8^2 + 6 \times 8^1 + 7 \times 8^0$
16진법	0에서 9, A(10), B(11), C(12), D(13), E(14), F(15)까지의 16개 숫자를 사용하여 한 자리 숫자를 표현 - $AB1_{16}$ $= A \times 16^2 + B \times 16^1 + 1 \times 16^0$

### 3) 진수 변환

<p>2 진수 → 10진수 변환</p>	<ul style="list-style-type: none"> <li>- 각 진수를 쓴다.</li> <li>- 각 자리 값을 쓴다.</li> <li>- 진수와 자리 값 곱해서 더한다.</li> <li>- 2진수 1101 → 10진수 변환</li> </ul> $  \begin{array}{cccc}  2^3 & 2^2 & 2^1 & 2^0 \\  1 & 1 & 0 & 1 \\  8 & 4 & 2 & 1 \\  1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13  \end{array}  $ $  \begin{aligned}  1101_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0  \end{aligned}  $
<p>8, 16진수 → 10진수 변환</p>	$  \begin{aligned}  567_8 &= 5 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 \\  AB1_{16} &= A \times 16^2 + B \times 16^1 + 1 \times 16^0  \end{aligned}  $
<p>10진수 → 2진수 변환</p>	 <p>주어진 값을 원하는 진수 값으로 나눔. 나누어지지 않을 때까지 계속 나누어 나머지를 역순으로 배열 결과 : 10100<sub>(2)</sub></p>
<p>10진수 → 8, 16진수 변환</p>	
<p>2진수 → 8, 16진수 변환</p>	<ul style="list-style-type: none"> <li>- 2진수를 3자리씩 묶어서 해당하는 8진수 숫자로 바꿀 수 있음</li> <li>- 2진수를 4자리씩 묶어서 해당하는 16진수 숫자로 바꿀 수 있음</li> </ul>

## 4. 프로그래밍 이해

### (1) 프로그램 개념

- 프로그램 : 처리하고자 하는 업무를 컴퓨터가 이해할 수 있는 언어(Programming Language)를 사용하여 일정한 문법(명령 사용)에 따라 업무처리 방법과 처리순서를 지정하는 명령문 등의 집합체
- 프로그래밍: 프로그램을 작성하는 과정
- 프로그래머: 프로그램을 작성하는 사람 또는 직업

### (2) 프로그래밍 언어(Programming Language)

- 컴퓨터가 직접 이해할 수 있는 언어로 기계어(Machine Language)가 있는데 작성이 불편하고, 수정도 어려우므로, 좀 더 편리하게 사용할 수 있는 인간이 사용하는 언어로 프로그램을 작성할 수 있도록 개발된 것

저급 언어	<ul style="list-style-type: none"> <li>- 하드웨어 지향의 기계 중심 언어</li> </ul> <table border="1" data-bbox="574 999 1254 1348"> <tr> <td data-bbox="574 999 715 1189">기계어</td><td data-bbox="715 999 1254 1189"> <ul style="list-style-type: none"> <li>- 컴퓨터를 수행하기 위해 프로그램을 숫자들로 인코딩된 명령들로 이루어진 것</li> <li>- 인코딩체계를 기계어라 부름</li> <li>- 01010001011</li> </ul> </td></tr> <tr> <td data-bbox="574 1189 715 1348">어셈블리 언어</td><td data-bbox="715 1189 1254 1348"> <ul style="list-style-type: none"> <li>- 기계 명령을 표현하기 위한 기호 체계 즉, 프로그램 표현에 사용되는 기호체계</li> <li>- ADD AX, BX</li> </ul> </td></tr> </table>	기계어	<ul style="list-style-type: none"> <li>- 컴퓨터를 수행하기 위해 프로그램을 숫자들로 인코딩된 명령들로 이루어진 것</li> <li>- 인코딩체계를 기계어라 부름</li> <li>- 01010001011</li> </ul>	어셈블리 언어	<ul style="list-style-type: none"> <li>- 기계 명령을 표현하기 위한 기호 체계 즉, 프로그램 표현에 사용되는 기호체계</li> <li>- ADD AX, BX</li> </ul>
기계어	<ul style="list-style-type: none"> <li>- 컴퓨터를 수행하기 위해 프로그램을 숫자들로 인코딩된 명령들로 이루어진 것</li> <li>- 인코딩체계를 기계어라 부름</li> <li>- 01010001011</li> </ul>				
어셈블리 언어	<ul style="list-style-type: none"> <li>- 기계 명령을 표현하기 위한 기호 체계 즉, 프로그램 표현에 사용되는 기호체계</li> <li>- ADD AX, BX</li> </ul>				
고급 언어	<ul style="list-style-type: none"> <li>- 사람이 이해하기 쉬운 일상 언어와 기호를 사용한 인간 중심의 언어</li> <li>- 기계 독립적이며, 컴파일러(compiler)라고 불리는 프로그램에 의해 기계어로 변환된다.</li> <li>- C, C++, 자바, 파이썬 등</li> </ul>				

### (3) 컴파일러와 인터프리터

- 컴파일러와 인터프리터는 고급 언어로 작성된 원시 프로그램을 목적 프로그램으로 번역하는 번역 프로그램이며, 프로그램 번역 방식에 따라 구분한다.

컴파일러	<ul style="list-style-type: none"> <li>- 고급 언어로 작성된 프로그램 전체를 목적 프로그램으로 번역한 후, 링킹 작업을 통해 컴퓨터에서 실행 가능한 실행 프로그램을 생성한다.</li> <li>- 번역과 실행과정을 거쳐야 하기 때문에 번역 과정이 번거롭고 번역 시간이 오래 걸리지만, 한번 번역한 후에는 다시 번역하지 않으므로 실행 속도가 빠르다.</li> <li>- FORTRAN, COBOL, PASCAL, C, C++, PL/1 등</li> </ul>
인터프리터	<ul style="list-style-type: none"> <li>- 고급 언어로 작성된 프로그램을 한 줄 단위로 받아들여 번역하고, 번역과 동시에 프로그램을 한 줄 단위로 즉시 실행시키는 프로그램</li> <li>- 프로그램이 직접 실행되므로 목적 프로그램이 생성되지 않는다.</li> <li>- 줄 단위로 번역·실행되기 때문에 시분할 시스템에서 유용하며 원시 프로그램의 변화에 대한 반응이 빠르다.</li> <li>- 번역 속도는 빠르지만 프로그램 실행 시 매번 번역해야 하므로 실행 속도는 느리다.</li> <li>- BASIC, SNOBOL, LISP, APL 등</li> <li>- CPU 사용 시간의 낭비가 크다.</li> </ul>

#### (4) 프로그래밍의 순서

- 컴퓨터를 이용하여 업무를 효율적으로 처리하고자 할 경우에는 논리적이고 체계적인 프로그램을 작성하는 것이 중요

업무 분석	- 전산화하는 업무 범위의 결정과 경제적, 능률적인 측면에서 가장 효과적인 방법을 모색하고, 분석, 검토하여 최적의 방법을 선택함
입출력 설계	- 업무처리에 필요한 자료를 어느 매체를 이용할 것이며, 입출력 형식은 어떠한 형식으로 설정할 것인가를 설계함
순서도 작성	- 프로그램을 작성하기 용이하게 하기 위하여 업무처리의 순서와 방법을 순서도 작성 기호를 사용하여 단계별로 알아보기 쉽게 체계적으로 작성함
코딩	- 업무처리에 적합한 언어를 선택하여 문법에 의거하여 순서대로 코딩함
원시 프로그램 언어	- 코딩된 프로그램을 입력장치를 통하여 입력함
디버깅 (Debugging)	- 원시 프로그램은 컴파일러에 의해 목적 프로그램으로 번역 - 문법이나 논리적인 면에서 잘못된 부분이 있으면 에러가 발생 - 에러가 발생하면 에러 내용을 분석하여 원시 프로그램을 수정 - 에러가 발생하지 않고 정상적으로 동작하는 프로그램이 되도록 함
프로그램 테스트	- 시험 자료를 입력하여 실제 업무처리를 하는 것과 같이 실행
실행	- 업무 자료를 입력하여 실제 동작을 실행함
문서화	- 프로그램이 완성된 경우에도 업무처리에 불편한 점이 있으면 프로그램을 보완 - 프로그램이 만족하게 동작되면 프로그램에 관계되는 모든 내용을 문서로 남겨두어야 함



## 5. 불 대수와 디지털 논리회로

### (1) 불 대수 (Boolean Algebra)

- 2진 변수와 논리적인 동작을 취급하는 대수
- 두 개 또는 그 이상의 요소를 갖는 변수 집합과 세 개의 기본적인 논리 동작인 OR 와 AND 및 보수(complement)

AND 연산	- 연산 결과 두 개의 값 모두 1이면 1이 되고, 하나라도 0이 있으면 0이 됨
OR 연산	- 연산 결과 두 개의 값 중 하나라도 1이면 1이 되고, 두 개의 값 모두 0이면 0이 됨
NOT 연산	- 연산 결과로 0은 1이 되고, 1은 0이 됨

#### 1) 불 대수 법칙

- A, B, C가 0 또는 1의 값을 갖는 변수라 할 때, 불 대수와 관련된 법칙

교환법칙	<ul style="list-style-type: none"> <li>- <math>X + Y = Y + X</math></li> <li>- <math>X \cdot Y = Y \cdot X</math></li> </ul>
결합법칙	<ul style="list-style-type: none"> <li>- <math>X + (Y + Z) = (X + Y) + Z</math></li> <li>- <math>X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z</math></li> </ul>
분배법칙	<ul style="list-style-type: none"> <li>- <math>X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)</math></li> <li>- <math>X + (Y \cdot Z) = (X + Y) \cdot (X + Z)</math></li> </ul>

드모르간 (Demorgan) 법칙	$-(X + Y)' = X' \cdot Y'$ $-(X \cdot Y)' = X' + Y'$ $-(X + Y + Z)' = X'Y'Z'$ $-(X \cdot Y \cdot Z)' = X' + Y' + Z'$
동일법칙	$X \cdot X = X$ $X + X = X$
항등원	$X \cdot 1 = X$ $X + 0 = X$
이중부정 법칙	$-(X')' = X$
흡수법칙	$X + X \cdot Y = X$ $X \cdot (X + Y) = X$
연산 법칙	$X \cdot 0 = 0$ $X + 1 = 1$ $X + X' = 1$ $X \cdot X' = 0$

## 2) 논리 함수의 간소화

- 불 대수의 정리를 이용하여 간소화하거나 카르노 맵 (Karnaugh map) 방법을 이용하여 간소화한다.
- 2변수 맵

$x \backslash y$	0	1
0	$X' \cdot Y'$	$X' \cdot Y$
1	$X \cdot Y'$	$X \cdot Y$

- 3변수 맵

X \ YZ	00	01	11	10
0	$X' \cdot Y' \cdot Z'$	$X' \cdot Y' \cdot Z$	$X' \cdot Y \cdot Z$	$X' \cdot Y \cdot Z'$
1	$X \cdot Y' \cdot Z'$	$X \cdot Y' \cdot Z$	$X \cdot Y \cdot Z$	$X \cdot Y \cdot Z'$

-  $F(x, y, z) = \sum(0, 1, 4, 5, 7)$


X \ YZ	00	01	11	10
0	1	1	0	0
1	1	1	1	0

전달 함수값이 1인 것을 2의 승수개로 묶는다. 각 그룹 항목들을 논리합(OR)을 취하고, 3변수 맵에서 4개의 항은 1개의 변수로, 2개의 항은 2개의 변수로 표현할 수 있으므로,  $Y' + X \cdot Z$  이 된다.

(2) 디지털 논리회로


1) AND 논리회로

- 불 대수의 AND 연산을 하는 게이트
- 두 개의 입력을 받아 둘 다 1인 경우에만 1 출력, 나머지 경우에는 0 출력

논리식	AB, A∧B, A∩B, A.B , A AND B															
논리도																
진리표	<table><tr><th>A</th><th>B</th><th>A.B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	A.B	0	0	0	0	1	0	1	0	0	1	1	1
A	B	A.B														
0	0	0														
0	1	0														
1	0	0														
1	1	1														

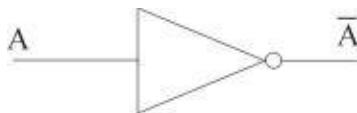
## 2) OR 논리회로

- 불 대수의 OR 연산을 하는 게이트
- 두 개의 입력을 받아 둘 중 하나라도 1이면 1 출력, 둘 다 0이면 0 출력

논리식	$A+B$ , $A\vee B$ , $A\cup B$ , $A$ OR $B$															
논리도																
진리표	<table><tr><th>A</th><th>B</th><th>A + B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	A + B	0	0	0	0	1	1	1	0	1	1	1	1
A	B	A + B														
0	0	0														
0	1	1														
1	0	1														
1	1	1														

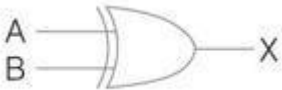
## 3) NOT 논리회로

- 불 대수의 NOT 연산을 하는 게이트
- 한 개의 입력만 받아 1이면 0 출력, 0이면 1 출력

논리식	$\bar{A}$ , $A'$ , NOT $A$						
논리도							
진리표	<table border="1"> <thead> <tr> <th>A</th><th><math>\bar{A}</math></th></tr> </thead> <tbody> <tr> <td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td></tr> </tbody> </table>	A	$\bar{A}$	0	1	1	0
A	$\bar{A}$						
0	1						
1	0						


#### 4) XOR 논리회로

- 두 개의 입력을 받아 입력값이 같으면 0 출력, 다르면 1 출력

논리식	$A \oplus B$															
논리도																
진리표	<table><tr><th>A</th><th>B</th><th>A XOR B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	A XOR B	0	0	0	0	1	1	1	0	1	1	1	0
A	B	A XOR B														
0	0	0														
0	1	1														
1	0	1														
1	1	0														

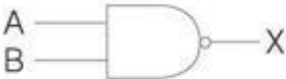
#### 5) NOR 논리회로

- OR 게이트 오른쪽에 NOT 게이트를 연결한 것과 같음
- OR 게이트와 반대로 동작

논리식	$(A + B)'$															
논리도																
진리표	<table><tr><th>A</th><th>B</th><th>A NOR B</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	A NOR B	0	0	1	0	1	0	1	0	0	1	1	0
A	B	A NOR B														
0	0	1														
0	1	0														
1	0	0														
1	1	0														

#### 6) NAND 논리회로

- AND 게이트 오른쪽에 NOT 게이트를 연결한 것과 같음
- AND 게이트와 반대로 동작

논리식	$(A \cdot B)'$															
논리도																
진리표	<table><tr><th>A</th><th>B</th><th>A NAND B</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	A NAND B	0	0	1	0	1	1	1	0	1	1	1	0
A	B	A NAND B														
0	0	1														
0	1	1														
1	0	1														
1	1	0														

### (3) 조합논리회로 (Combinational Circuit)

- 현재의 입력 조합에 의하여 출력이 직접 결정되는 논리 게이트로 구성
- 불 대수의 집합들에 의하여 완전히 논리적으로 명시되는 특별한 정보 처리 동작을 수행
- 반가산기, 전가산기, 반감산기, 전감산기, 인코더, 디코더, 디멀티플렉서 (Demultiplexer), 멀티플렉서(Multiplexer) 등

#### 1) 조합 회로의 설계 절차

- 문제가 제시된다.
- 입력과 출력 변수에 문자 기호를 붙인다.
- 입력과 출력 사이의 관계를 정의하는 진리표를 유도한다.
- 각 출력에 대한 간소화된 불 함수를 얻는다.
- 논리도를 작성한다.

#### 2) 반가산기(Half Adder)

- 2진수 한자리 덧셈 회로로서 두 개의 입력과 하나의 출력, 자리 올림수가 있는 회로
- 두 개의 연산자(operand), 즉 가수와 피가수를 더하여 합을 만들기 위한 회로로 덧셈뿐만 아니라 뺄셈, 곱셈, 나눗셈 연산을 위하여 사용
- 한 개의 XOR 게이트와 한 개의 AND 게이트로 구현

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = AB' + A'B \quad C = AB$$

#### 3) 전가산기(Full Adder)

- 반가산기의 회로에 이전 자리에서 발생한 자리올림 수를 처리할 수 있도록 한 회로
- 회로는 반가산기 두 개와 한 개의 OR 게이트로 구성

A	B	Z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = A'B'Z + A'BZ' + AB'Z' + ABZ$$

$$= A \oplus B \oplus Z$$

$$C = A'BZ + AB'Z + ABZ' + ABZ$$

$$= A \cdot B + (A \oplus B) \cdot Z$$

#### 4) 멀티플렉서(MUX : 선택기)

- 여러 개의 입력선 중에서 하나를 선택하여 단일 출력선으로 연결하는 조합 논리 회로

#### 5) 디멀티플렉서(DMUX : 분배기)

- 하나의 입력선으로부터 정보를 받아 여러 개의 출력 단자 중 하나의 출력선으로 정보를 출력하는 회로

#### 6) 디코더(해독기 : Decoder)

- 암호 형태로 전달된 정보를 원래 상태로 복원해 주는 장치

#### 7) 엔코더(부호기 : Encoder)

- 어떤 특정한 장치에서 사용되는 정보를 다른 곳으로 전송하기 위해 일정한 규칙에 따라 암호로 변환하는 장치

#### 8) 플립플롭 (Flip-Flop)

- 플립플롭 한 개 비트의 2진정보를 보관하며 외부에서 입력을 할 때까지 현재 상태를 유지하는 조합 회로
- RS 플립플롭 : S와 R선의 입력을 조절하여 임의의 값을 그대로 유지시키거나 무조건 0 또는 1의 값을 기억시키기 위해서 사용되는 플립플롭

S	R	Q(t+1)	설명
0	0	Q(t)	현 상태가 그대로 출력
0	1	0	0을 출력(Reset)
1	0	1	1을 출력(Set)
1	1	?	알 수 없다

- JK 플립플롭 : JK 플립플롭은 RS 플립플롭이  $R=S=1$ 의 입력을 허용하지 않는 결점을 보완한 것으로,  $J=K=1$ 일 때 현재 상태의 보수 출력을 하도록 한 플립플롭이다.

J	K	$Q(t+1)$	설명
0	0	$Q(t)$	현 상태가 그대로 출력
0	1	0	0을 출력(Reset)
1	0	1	1을 출력 (Set)
1	1	$Q(t)'$	반전