

# System Programming & OS 실습

## Appendix 4. Network

1

최민국

Dankook University

mgchoi@dankook.ac.kr

# 네트워크 인터페이스 (NIC)

- NIC란?

- 컴퓨터나 다른 네트워크 장치가 네트워크에 연결되어 데이터를 주고받을 수 있게 해주는 하드웨어 장치
- 흔히 네트워크 카드 또는 이더넷 카드라고 부름

- NIC의 주요기능

- 데이터 전송 및 수신

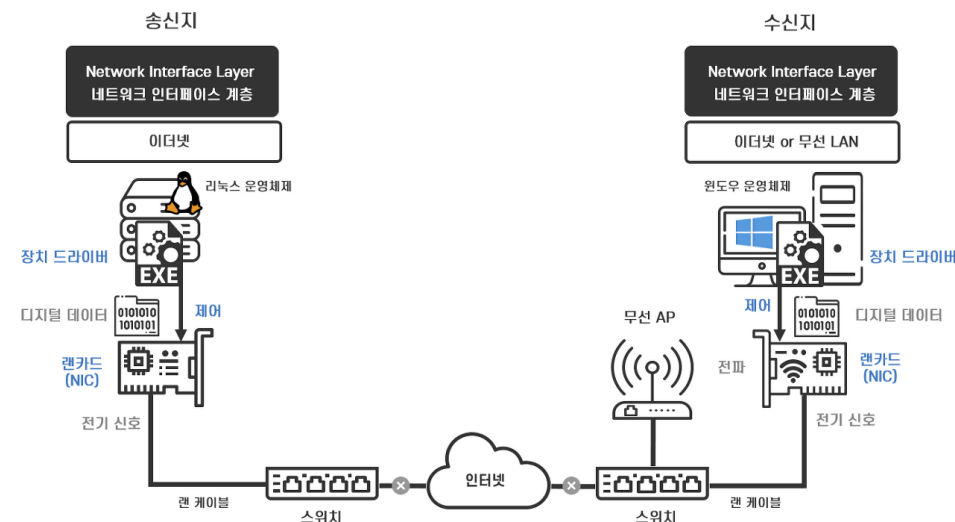
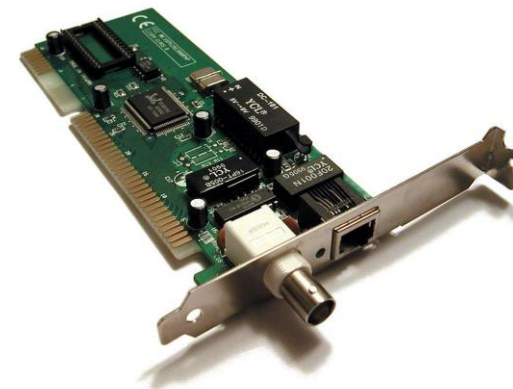
- NIC는 네트워크 상의 다른 장치로부터 데이터를 받아 컴퓨터에 전달
    - 컴퓨터에서 발생한 데이터를 네트워크를 통해 외부로 전달

- 주소 할당

- MAC 주소(Media Access Control 주소)는 NIC에 고유하게 할당된 48비트 하드웨어 주소
    - 네트워크 상에서 장치를 식별할 때 사용되며, 제조 시 NIC에 고정으로 부여

- 네트워크 프로토콜 처리

- NIC는 데이터를 전송하기 위해 사용되는 네트워크 프로토콜(예: TCP/IP, Ethernet 등)을 처리



# 네트워크 인터페이스 (NIC)

3

- NIC의 유형

- 이더넷 카드

- 유선 네트워크 연결에 사용
    - 가장 일반적인 유형

- 무선 NIC(Wi-Fi 카드)

- 무선 네트워크에 연결하기 위한 카드

- 광 NIC

- 광섬유 케이블을 사용하는 NIC
    - 고속 데이터 전송이 필요한 경우

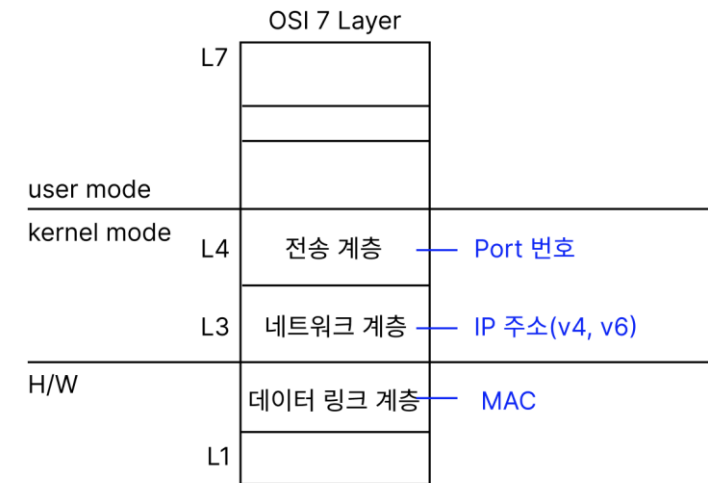


- 네트워크 인터페이스 정보 확인

- \$ ifconfig -a                   # 모든 네트워크 인터페이스 정보 표시
  - \$ ip addr show                # 더 최신 명령어
  - \$ ip link show                # MAC 주소 확인

# IP 주소와 포트(Port)

- IP 주소
  - 인터넷이나 로컬 네트워크에서 컴퓨터를 식별하는 고유한 주소
  - IPv4는 32비트 주소 체계 (ex.192.168.1.1)
- 포트(Port)
  - 해당 컴퓨터에서 실행 중인 특정 프로세스나 서비스를 식별
  - 운영 체제 통신에서의 종단점
  - 포트 번호: 0~65536번까지 존재
  - well-known port: 0~1023번까지는 이미 특정 통신 용도로 정해져 있음
- IP 주소와 포트의 관계
  - IP 주소: 컴퓨터의 집 주소
  - 포트: 집 안에서 각 프로세스가 위치한 방 번호
    - Ex: HTTP 통신 (80번 포트), HTTPS 통신(443번 포트)



- 실습: 현재 시스템의 열린 포트 확인하기
  - # 열린 포트와 리스닝 중인 프로세스 확인
    - \$ sudo netstat -tulpn
      - netstat : 네트워크 연결 정보를 확인하는 명령어
        - -t : TCP 프로토콜의 연결을 표시
        - -u : UDP 프로토콜의 연결을 표시
        - -l : 현재 리스닝(listening) 상태인 소켓(포트)을 표시
        - -p : 해당 포트를 사용하는 프로세스 ID(PID) 및 프로세스 이름을 표시
        - -n : 호스트 이름과 서비스 이름을 숫자로만 표시
    - \$ sudo lsof -i -P -n | grep LISTEN
      - lsof : 열린 파일 목록을 표시(네트워크 소켓도 파일로 취급)
        - -i : 네트워크와 관련된 파일(소켓)만 표시.
        - -P : 포트 번호를 서비스 이름으로 변환하지 않음.
        - -n : 호스트명을 IP 주소로 변환하지 않음(속도 향상)
        - | grep LISTEN : 수신 대기(Listen) 중인 포트만 필터링.

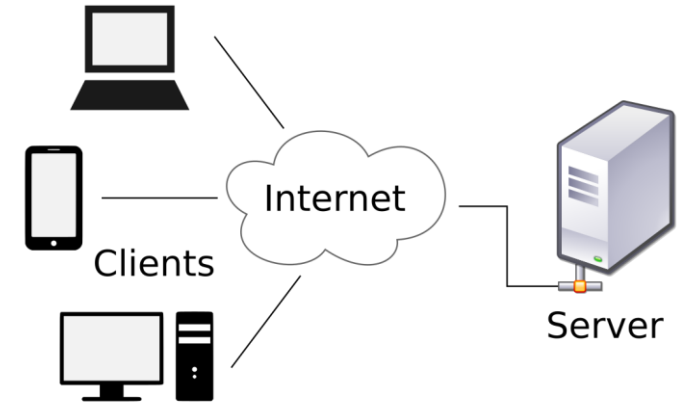
- Localhost란?
  - 루프백 네트워크 인터페이스
    - 현재 사용중인 자기 자신의 컴퓨터를 가리키는 호스트 이름
  - 웹 응용 프로그램이나 웹 사이트에 로컬로 액세스하고 테스트하는 데 사용
  - 자신의 컴퓨터에서 웹 서버를 실행할 수 있으므로 인터넷 연결 없이도 프로젝트를 개발하고 테스트 가능
- Localhost의 특징
  - 루프백 주소
    - localhost는 일반적으로 IP 주소 127.0.0.1과 연결
  - 테스트 환경
    - 테스트 및 개발을 위한 통제되고 격리된 환경을 제공
  - 오프라인 개발
    - 활성 인터넷 연결이 필요하지 않으므로 오프라인 개발이 가능

(실습) 루프백 주소 통신  
\$ ping 127.0.0.1  
\$ ping localhost

# 서버-클라이언트 구조

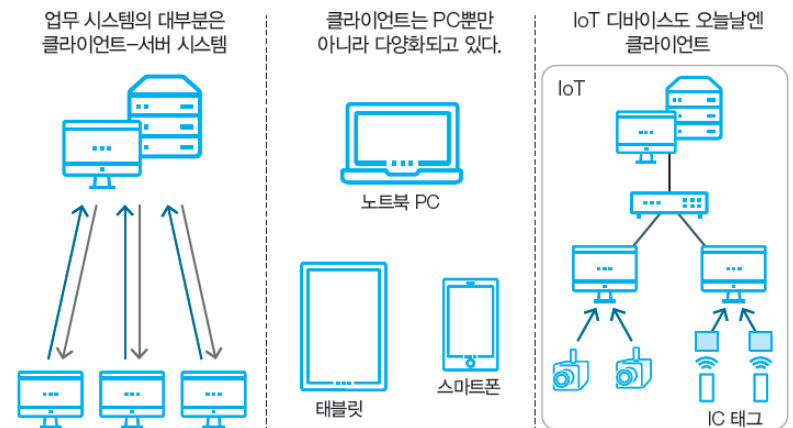
- 서버-클라이언트 모델

- 클라이언트:
  - 서비스를 요청(소비)하는 장치나 프로그램
- 서버
  - 클라이언트의 요청에 응답하여 서비스(자원)를 제공하는 장치나 프로그램



- 예시

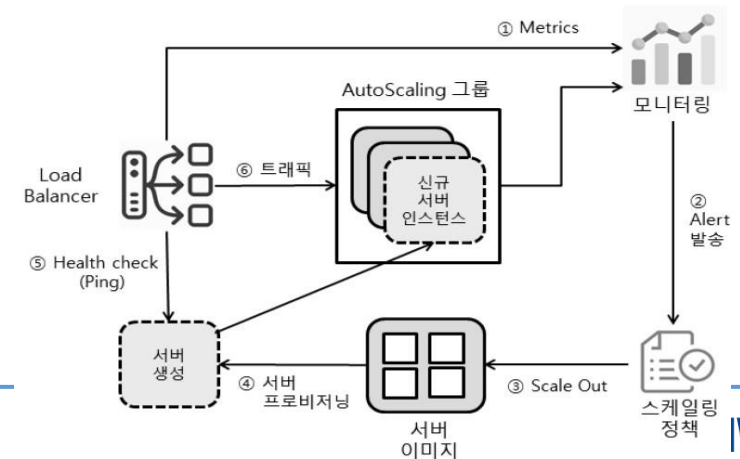
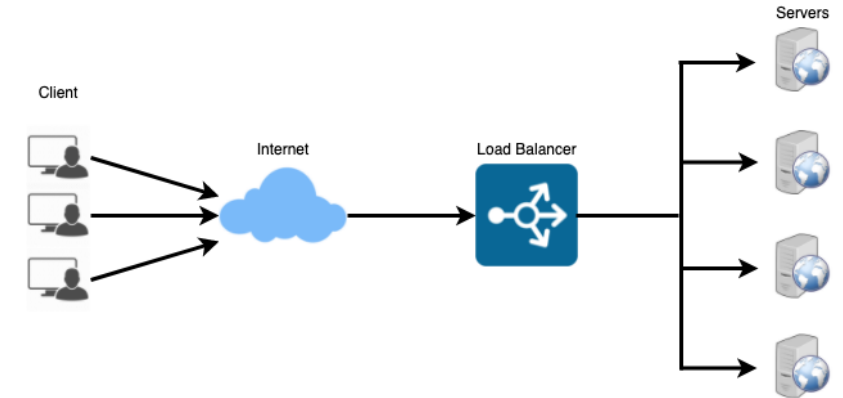
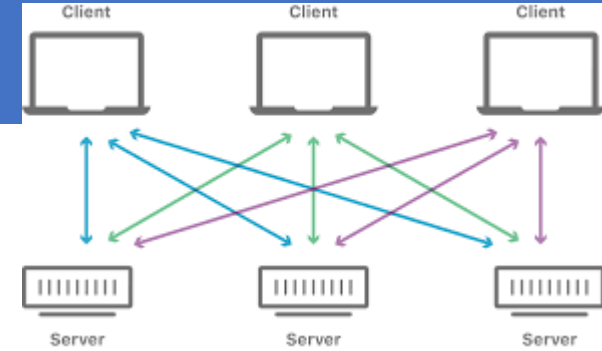
- 웹 서버
  - 클라이언트
    - 웹 브라우저를 통해 어떤 웹사이트에 접속하고 서비스 요청
  - 서버
    - 원격의 웹 서버는 요청을 받아 처리한 뒤 웹페이지 데이터를 클라이언트에 전송



# 서버-클라이언트 구조

8

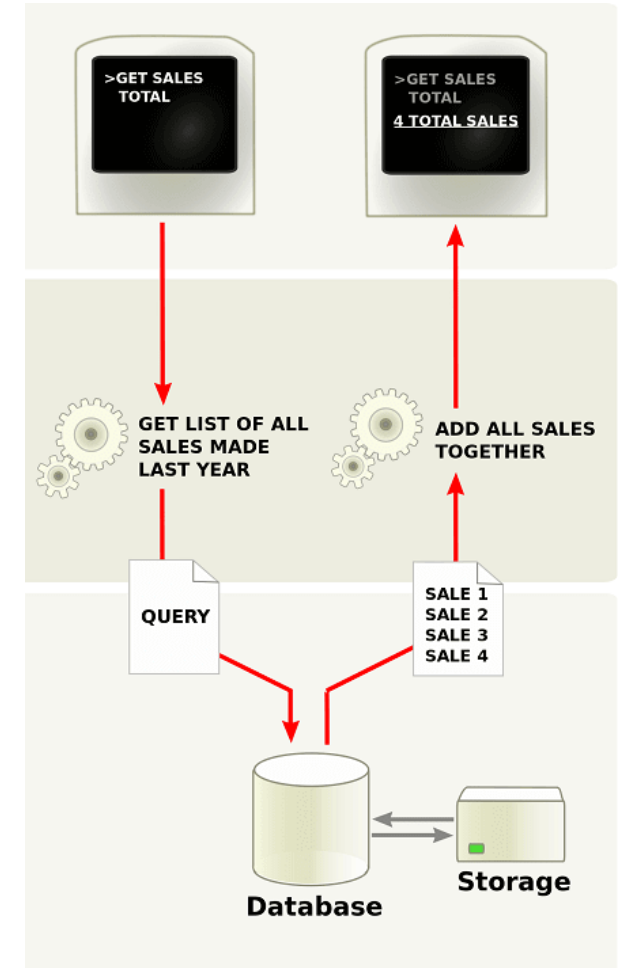
- 특징
  - 한 서버는 동시에 여러 클라이언트의 요청을 처리 가능
  - 한 클라이언트가 여러 서버에 요청가능
- 장점
  - 클라이언트-서버 모델의 장점은 역할 분담으로 효율을 높임
  - 여러 클라이언트를 중앙 서버가 일괄 관리함으로써 데이터 일관성을 유지
- 단점:
  - 서버에 부하가 집중되거나, 서버가 다운되면 서비스 전체가 중단





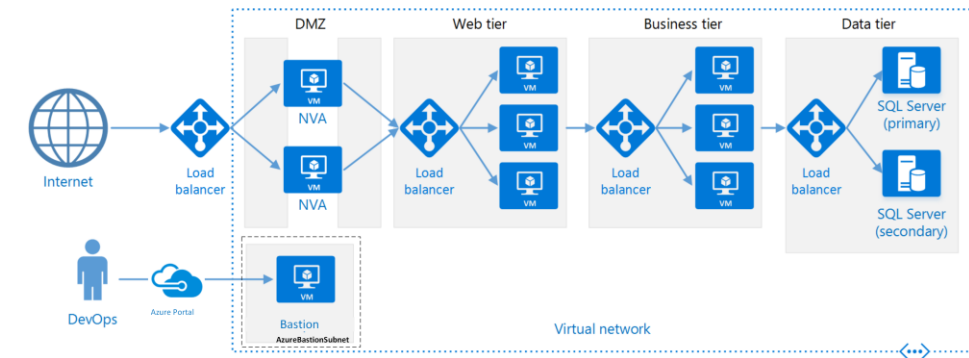
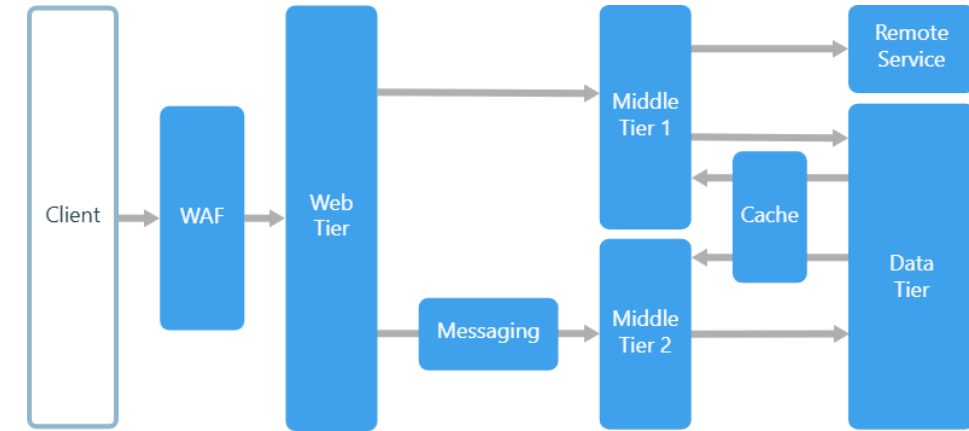
- N-Tier Architecture

- 애플리케이션을 여러 계층으로 나누어 각 계층이 특정 기능을 담당
  - 일반적인 3-Tier Architecture:
    - 프레젠테이션 계층
      - 사용자 인터페이스 담당 (UI를 표시하고 사용자 입력을 받아 서버로 전달)
      - 웹 브라우저, 모바일 앱, 데스크톱 프로그램
    - 애플리케이션 계층
      - 비즈니스 로직 처리 (클라이언트로부터 요청을 받아 해석하고 필요한 연산을 수행)
    - 데이터 계층:
      - 데이터 저장 및 관리 (데이터베이스 관리 시스템(DBMS)이 동작)
      - 애플리케이션 로직 계층의 요청에 따라 데이터를 조회하거나 저장
      - 클라이언트와 직접 통신하지 않고, 중간 로직 계층을 통해서만 접근



- N-Tier Architecture

- 장점: 유연성과 확장성
  - 각 계층이 독립적으로 개발되고 배포
    - 예: 프론트엔드 웹 UI 수정이 데이터베이스에 영향 주지 않음
- 계층별로 부하 분산이나 교체가 가능
  - 급격한 사용자 트래픽 증가
    - > 웹 서버나 WAS 인스턴스를 여러 개 두고 로드 밸런싱 (Application 계층을 확장)
- 데이터 계층에서 병목 발생
  - > DB서버를 클러스터링하거나 고성능 장비로 변경



- Client-WebServer-WAS-DBMS 구조
  - 클라이언트 + 3티어 (웹서버, 앱서버, DB) 구조
  - 구성
    - 클라이언트(Client)
      - 웹 브라우저나 모바일 앱과 같이 사용자와 직접 상호작용하는 부분
    - 웹 서버(Web Server)
      - 정적 콘텐츠(HTML, CSS, 이미지 등) 제공 및 요청 라우팅 (예: Apache, Nginx)
    - WAS(Web Application Server)
      - 동적 콘텐츠 생성 및 비즈니스 로직 처리 (예: Tomcat, JBoss, uWSGI)
    - DBMS(Database Management System)
      - 데이터 저장 및 관리 (예: MySQL, PostgreSQL, Oracle)

## Web Service Architecture

