

TCP/IP 프로토콜에 대한 이해 & 서버 클라이언트 실습

인하공업전문대학 컴퓨터정보과
최효현 교수

주요사항

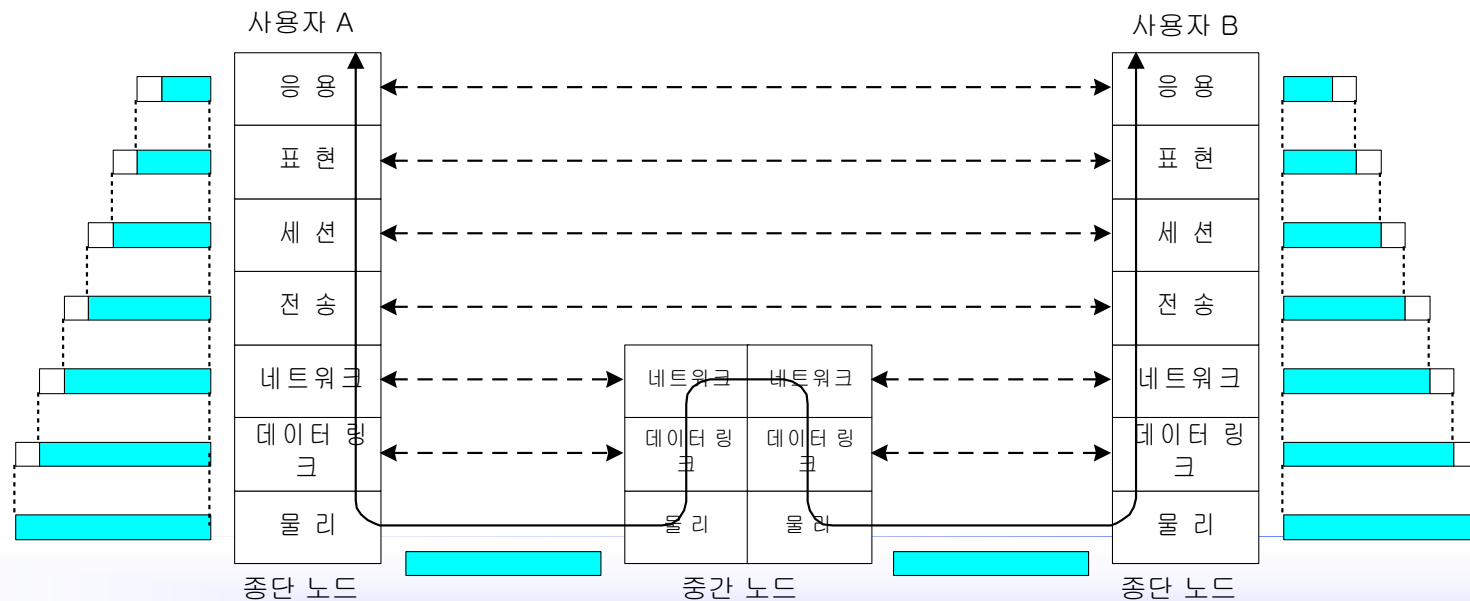
- ❑ OSI 모델
- ❑ TCP/IP 모델
- ❑ IP 주소
- ❑ DNS 개념

- ❑ 서버/클라이언트의 소켓 구현 - 실습
- ❑ 파일의 조작 - 실습

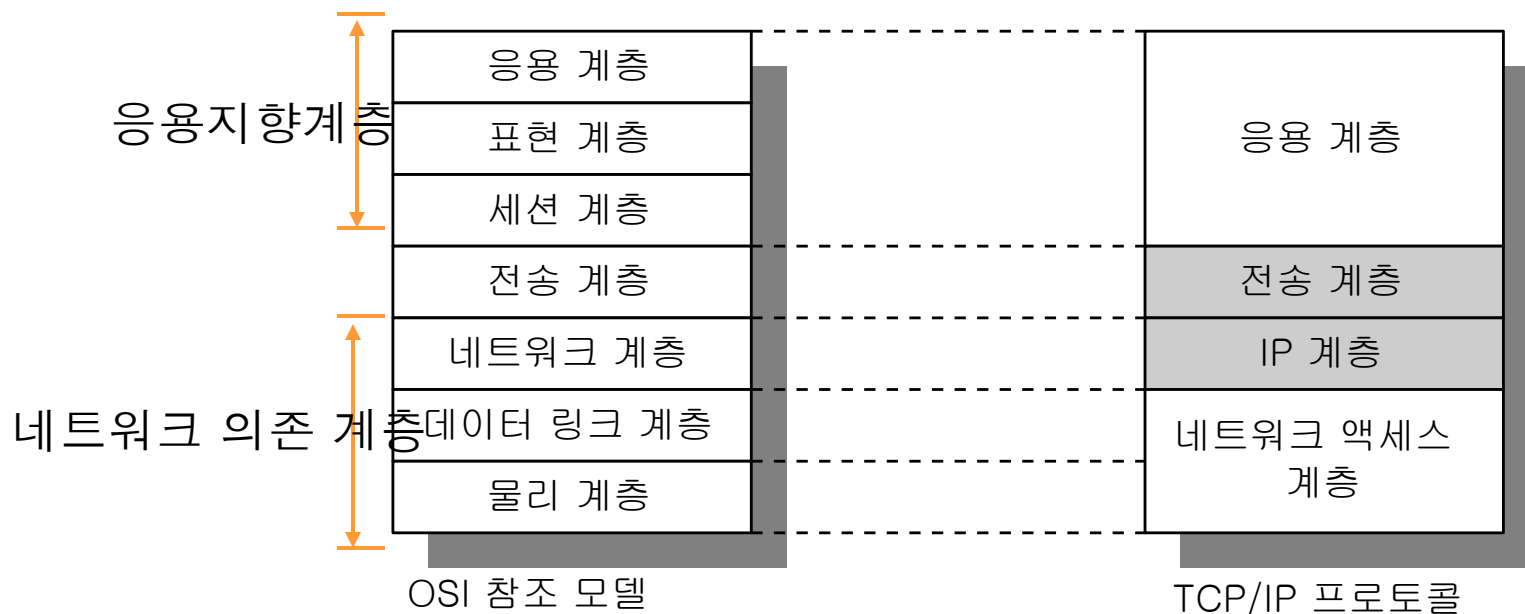
네트워크 참조모델(OSI 7Layer)

□ OSI 참조 모델(Reference Model)

- ✓ 서로 다른 시스템간의 통신을 위한 표준안을 제공
(통신에 방해가 되는 기술적인 문제점을 제거)
- ✓ 시스템간의 정보 교환을 하기 위한 상호 인터페이스를 정의



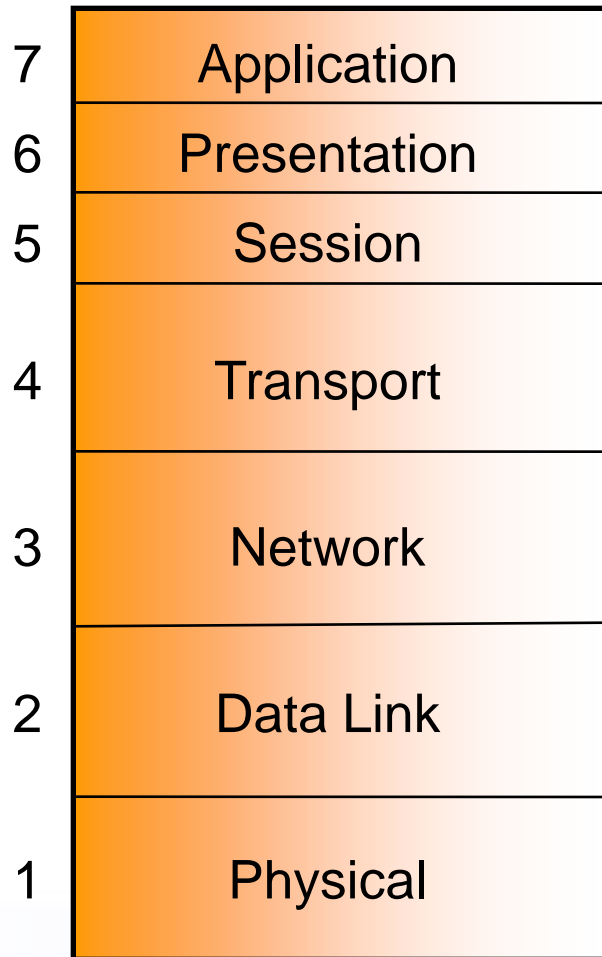
네트워크 참조모델(OSI 7Layer)



TCP/IP와 OSI 7계층 프로토콜 구조의 비교

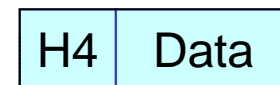
OSI 7 계층과 정보전송 단위

OSI 참조 모델 계층

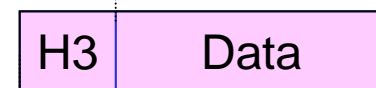


정보 전송 단위

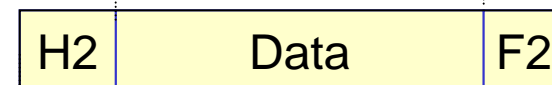
Segment (L4 PDU)



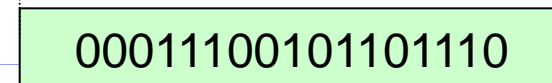
Packet (L3 PDU)



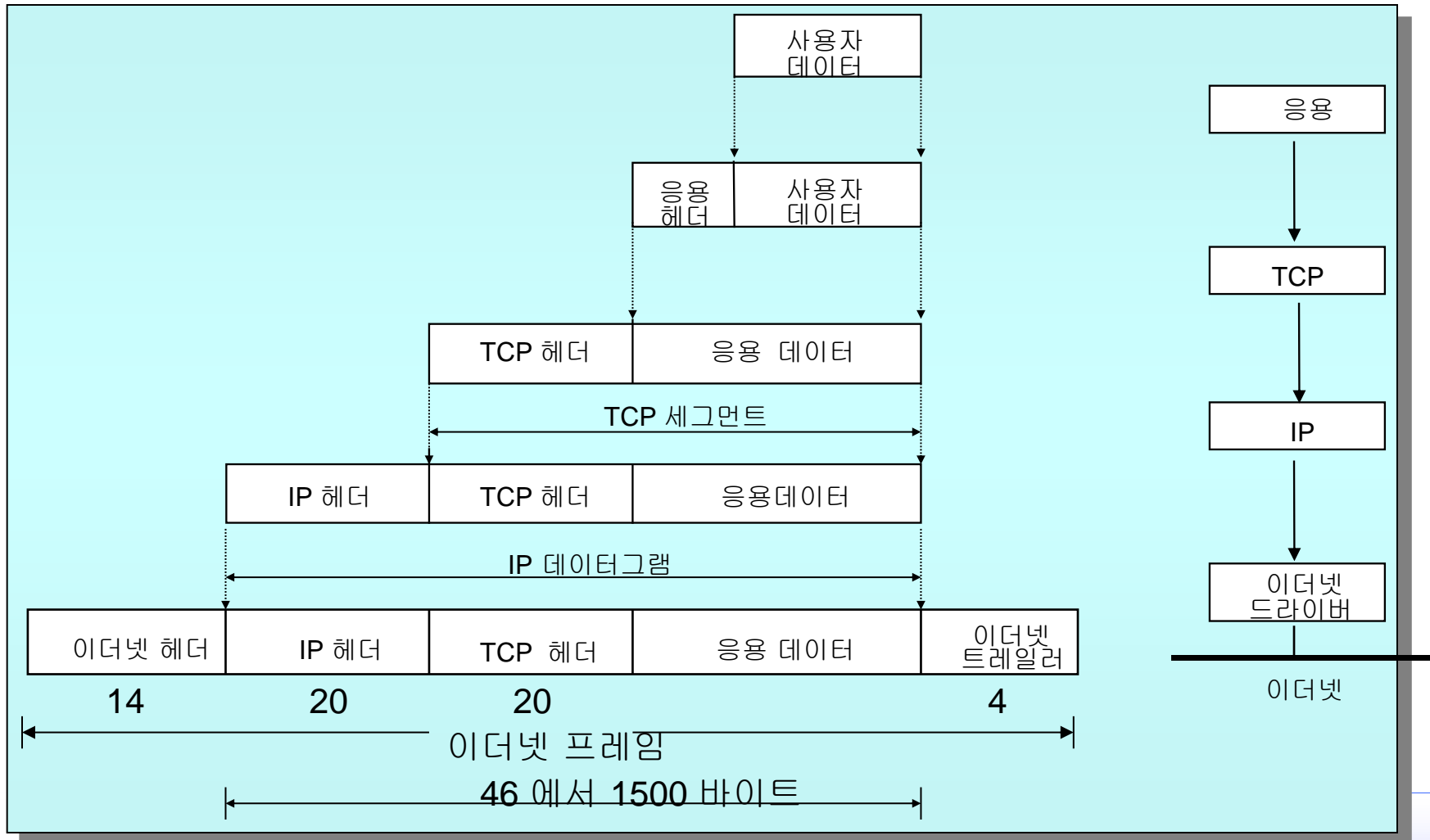
Frame (L2 PDU)



Bit



캡슐화(Encapsulation)



네트워크 참조모델 (OSI 7Layer)

□ 물리계층(Physical Layer)

- ✓ 전송 매체
- ✓ 커넥터 규격, 전압 레벨, 전송속도 전기적 규격
- ✓ 물리적 환경 설정

□ 링크계층(Link Layer)

- ✓ 노드 사이의 신뢰성 있는 전송
- ✓ 링크의 개설과 해제, 프레임의 동기화, 오류제어, 흐름 제어
- ✓ 노드간 프레임 전달
- ✓ Ethernet이 대표적인 링크 계층

네트워크 참조모델 (OSI 7Layer)

□ 네트워크 계층(Network Layer)

- ✓ 주소를 이용하여 목적지로의 패킷 전달을 수행
- ✓ 논리 주소 지정 (예: IP 주소)
- ✓ 라우팅 (목적지까지의 길 찾기)
 - 라우터는 네트워크 계층의 대표적 연결 장비

□ 전달 계층(Transport Layer)

- ✓ 출발지와 목적지 호스트에서만 수행
- ✓ TCP(Transmission Control Protocol)
 - 신뢰성 있는 연결형 데이터 전달 서비스
- ✓ UDP(User Datagram Protocol)
 - 비 연결형 데이터 전달 서비스

네트워크 참조모델 (OSI 7Layer)

□ 세션 계층(Session Layer)

- ✓ 통신 서비스의 개설, 서비스 유지 및 종료
- ✓ 종점 호스트에 있는 프로세스 사이의 통신 프로토콜

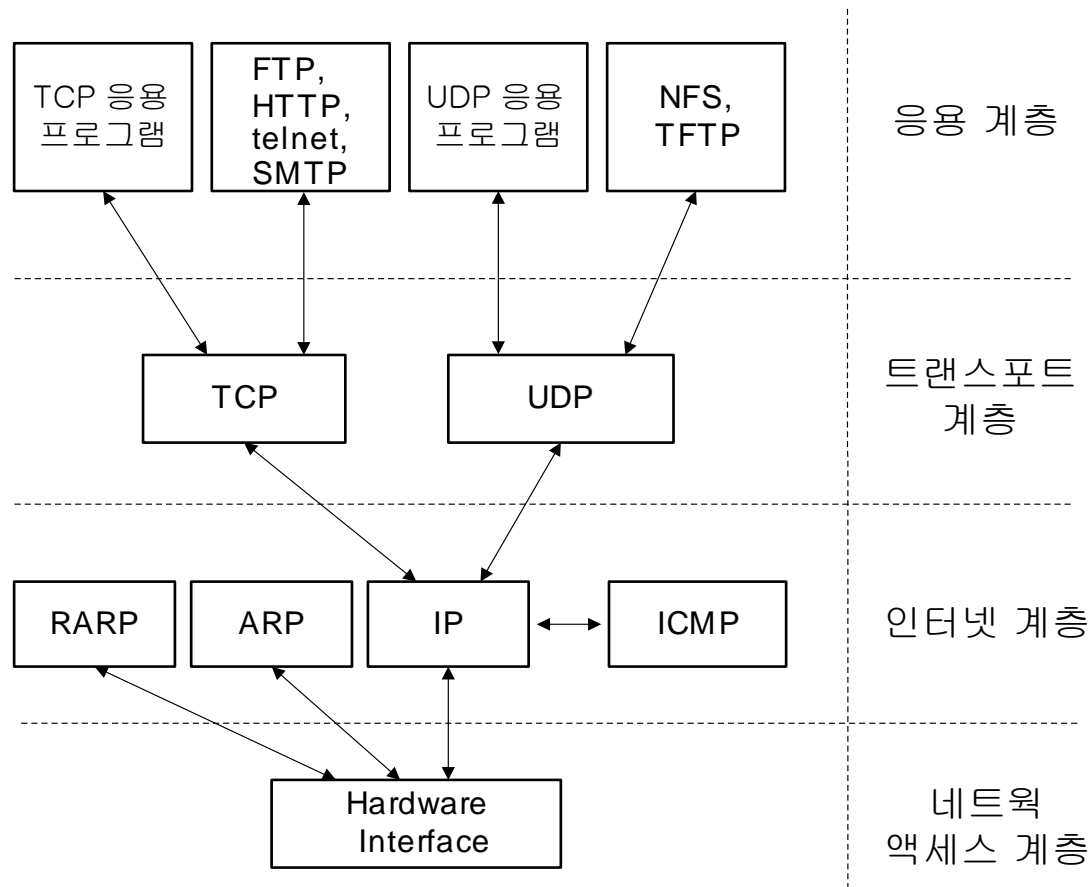
□ 표현 계층(Presentation Layer)

- ✓ 데이터의 표현 방식이 서로 다른 호스트 사이의 통신을 지원
- ✓ 표준화된 표현방식, 코드 변환, 데이터 압축, 데이터 암호화

□ 응용 계층(Application Layer)

- ✓ 네트워크를 이용한 서비스, 네트워크 응용프로그램

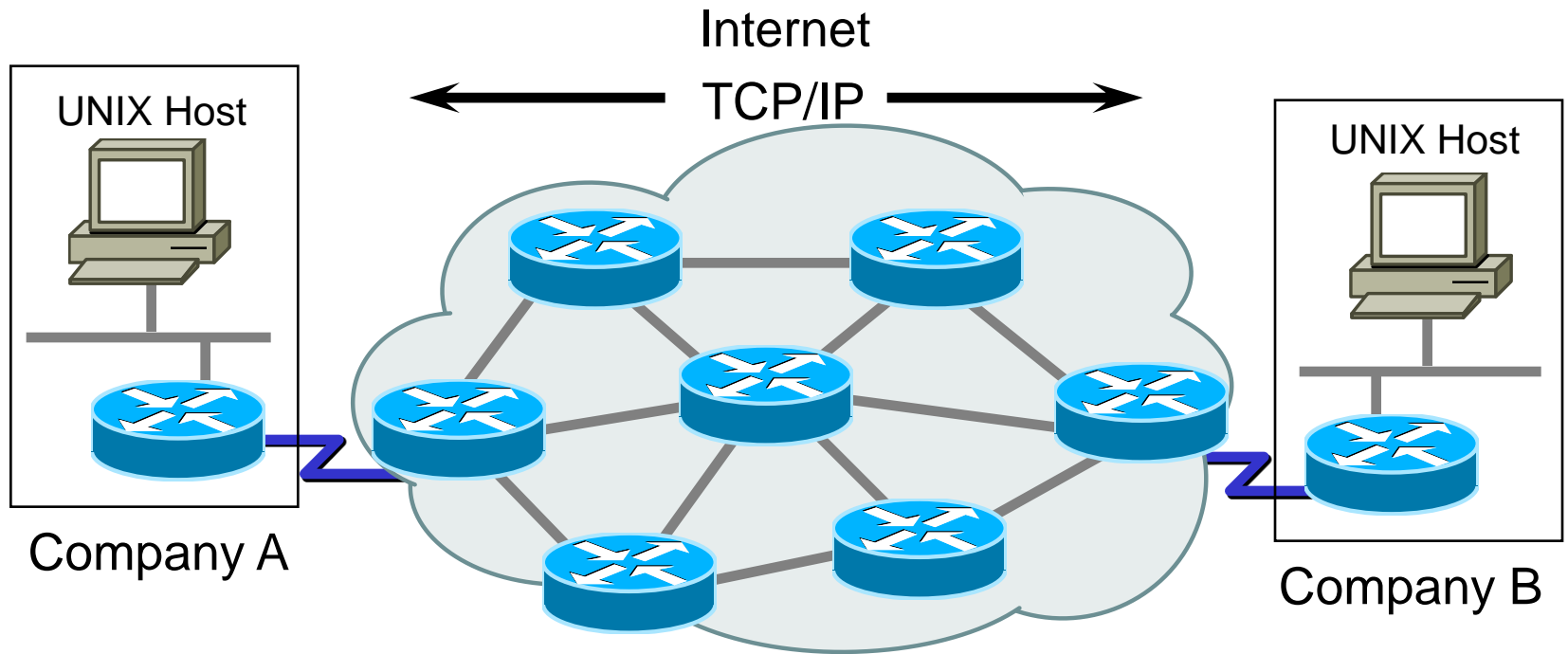
TCP/IP 이해 – 내부구조(1)



TCP/IP 이해 – 내부구조(2)

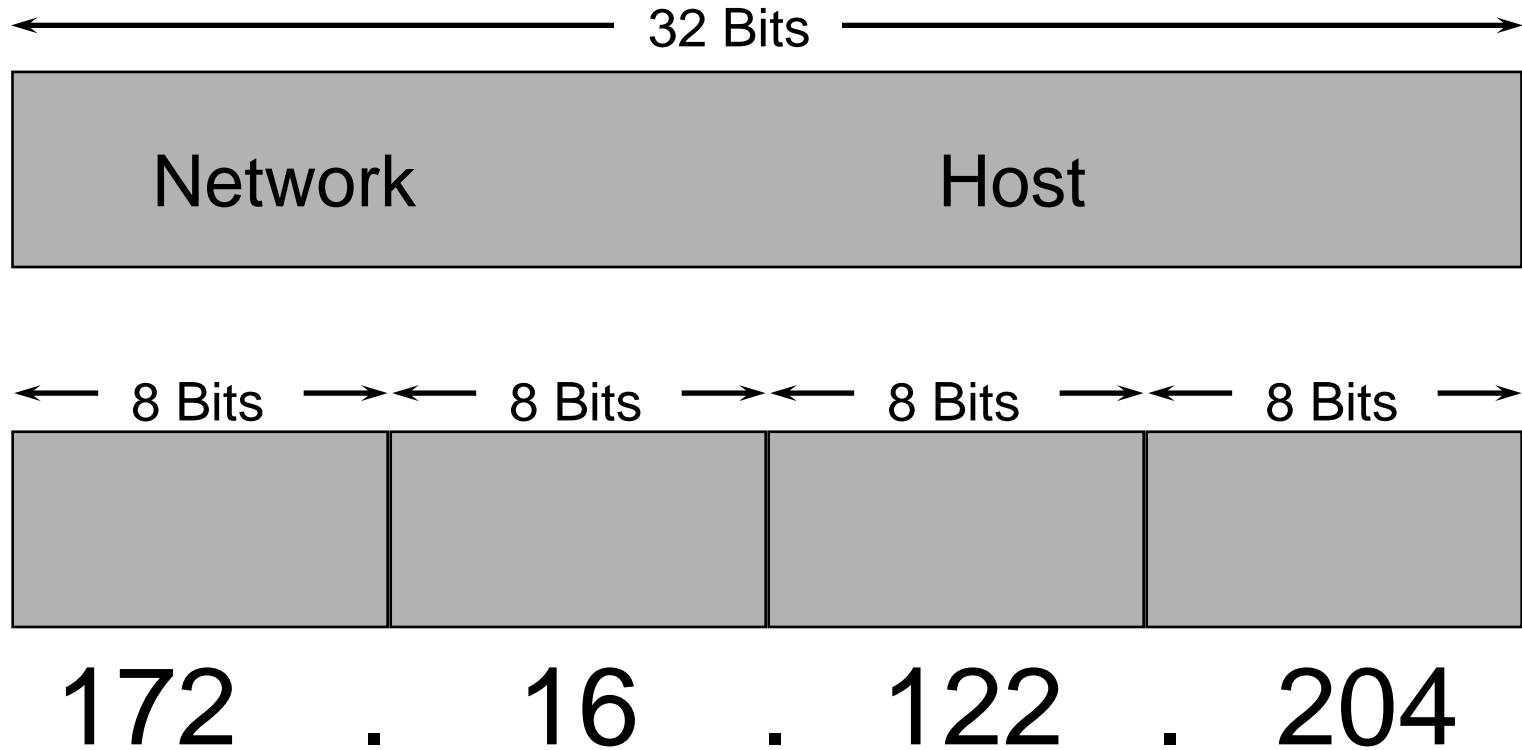
트랜스포트 프로토콜	응용 계층 서비스
TCP	<ul style="list-style-type: none">- FTP- Telnet- SMTP- HTTP
TCP, UDP (모두 지원)	<ul style="list-style-type: none">- NFS(Network File System)- DNS(Domain Name System)
UDP	<ul style="list-style-type: none">- Trivial FTP(TFTP)

TCP/IP 주소의 소개



- 각각의 스테이션이 고유의 주소를 가지고 서로 통신한다.
- 주소는 연결된 경로와 위치를 나타낸다
- 경로선정은 위치에 의한다
- 클래스화 되므로 주소로써 위치를 나타낼 수 있다.

IP Address 체계



HOST : 네트워크에 접속되는 단말등의 네트워크 기기

IP Address Classes

Class A:

0NNNNNNN	Host	Host	Host
Range (1-126)			

Class B:

10NNNNNN	Network	Host	Host
Range (128-191)			

Class C:

110NNNNN	Network	Network	Host
Range (192-223)			

Class D:

1110MMMM	Multicast Group	Multicast Group	Multicast Group
Range (224-239)			

Class E: 연구목적으로만 사용

IP 주소를 보고 클래스를 안다 (First Octet Rule)

High Order Bits	Octet in Decimal	Address Class
0	1 – 126	A
10	128 – 191	B
110	192 – 223	C
1110	224 – 239	D

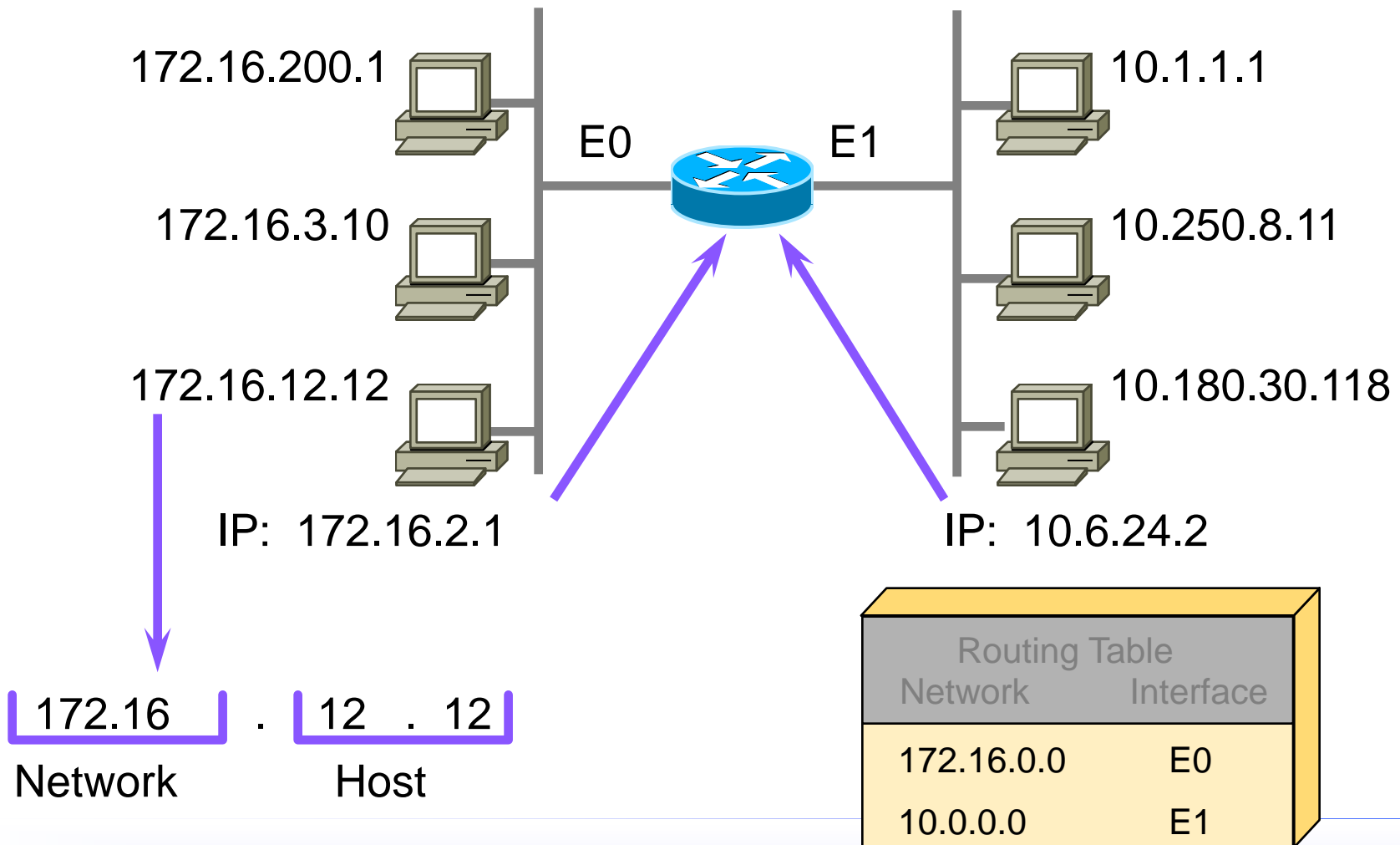
IP 주소 개념

Address	Class	Network	Host
10.2.1.1	A	10.0.0.0	0.2.1.1
128.63.2.100	B	128.63.0.0	0.0.2.100
201.222.5.64	C	201.222.5.0	0.0.0.64
192.6.141.2	C	192.6.141.0	0.0.0.2
130.113.64.16	B	130.113.0.0	0.0.64.16
256.241.201.10	Non existent		

IP 네트워크 주소 체계

- IPv4 (IP version 4) addresses ←우리가 지금 쓰는 방식
 - ✓ 32-bit addresses in four octets
 - ✓ a network number and a host number
 - ✓ Use dotted decimal format
 - ✓ Only 6% of IPv4 addresses are unassigned
- IPv6 (IP version 6) addresses
 - ✓ **128-bit addresses** ← 전세계 인구보다 많음
 - ✓ Will replace IPv4
 - ✓ Example:
5F1B:DF00:CE3E:E200:0020:0800:5AFC:2B36
 - ✓ Example: 0:0:0:0:0:0:192.1.1.17

Host Addresses



도메인 명

도메인명 시스템(DNS : Domain Name System)

◆ 구조 : 호스트명.소속단체.단체성격.소속국가

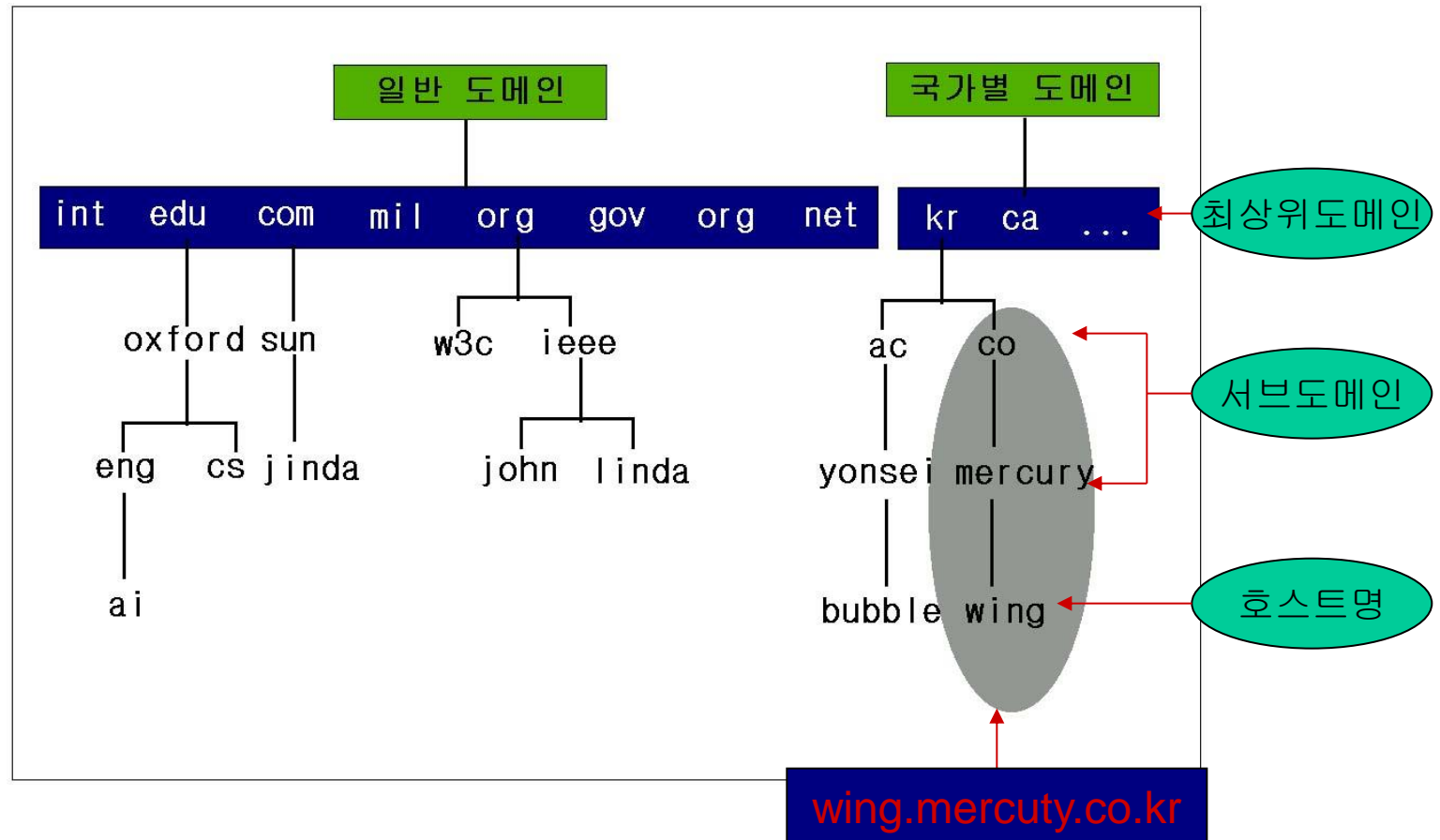
예) www.inhatc.ac.kr == 221.154.90.151

◆ 특 성

- IP 주소의 불편함 해소 : 도메인명 입력시 IP주소로 변환함
- 컴퓨터가 속한 기관이나 단체에 따라 계층적 구성
- 최상위 도메인 : 일반 도메인(미국의 기관)이나 국가를 의미함
- 부도메인 : 소속단체나 단체성격을 나타내는 도메인명

도메인 명

◆ 도메인명 영역



도메인 명

◆ 최상위 도메인명 영역

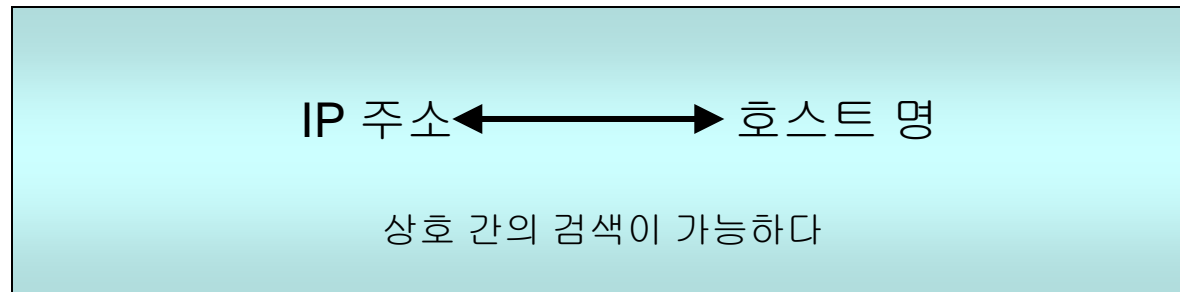
도메인	기관명	도메인	국가명	도메인	국가명
edu	교육 기관	kr	Korea, South	pl	Poland
com	사 업 체	kp	Korea, North	uk	United Kingdom
gov	정부 기관	jp	Japan	ua	Ukraine
int	국제 기구	ca	Canada	th	Thailand
org	비영리 공공기관	de	Germany	id	Indonesia
net	네트워크 관련 기관	ch	Switzerland	my	Malaysia
mil	미국 국방성 관련	fr	France	sg	Singapore

◆ 부도메인명(Subdomain) 영역

부도메인	출 처	의 미	예 제
co	Company	회 사	http://www.lg.co.kr
re	Research	연 구 소	http://etri.re.kr
ac	Academy	학술기관	http://www.yonsei.ac.kr
go	Government	정부기관	http://bluehouse.go.kr

DNS(Domain Name System)

~ IP 주소와 호스트 이름간의 대응관계 정보를 제공하는 분산 데이터 베이스





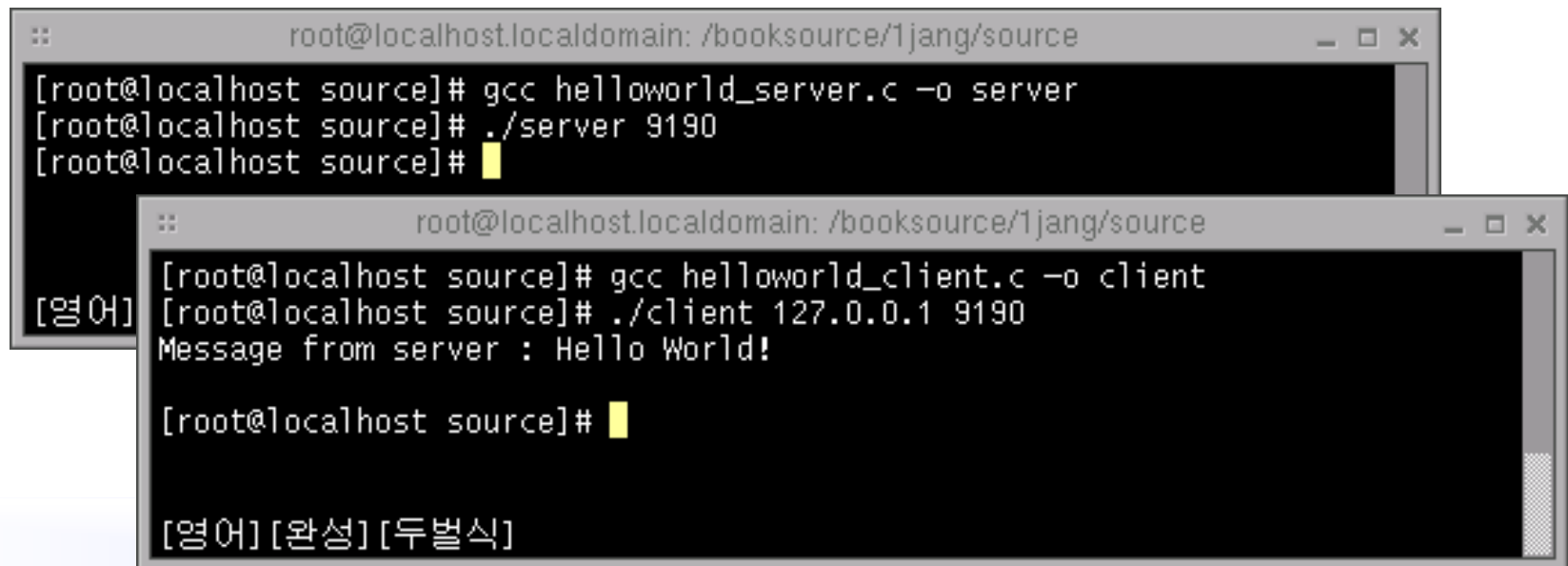
1. 네트워크 프로그래밍과 소켓의 이해

□ “Hello World!” 서버 / 클라이언트

1. 프로그램 예제

– helloworld_server.c, helloworld_client.c

2. 실행하기



```
root@localhost.localdomain: /booksource/1jang/source
[root@localhost source]# gcc helloworld_server.c -o server
[root@localhost source]# ./server 9190
[root@localhost source]#

[영어]
root@localhost.localdomain: /booksource/1jang/source
[root@localhost source]# gcc helloworld_client.c -o client
[root@localhost source]# ./client 127.0.0.1 9190
Message from server : Hello World!

[root@localhost source]#

[영어] [완성] [두벌식]
```


실습

- ❑ 두 명이 팀원을 이루어 각자의 서버에 접속하여 데이터를 읽어 오는지 확인
 - ✓ 각자 서버 출력에 고유의 내용을 입력
 - 예: server.c

```
23: char message[] = "Hello World!\n";  
→ char message[] = "학번 이름\n";
```
 - ✓ ./server Port_번호
 - ✓ ./client IP_주소 Port_번호

1. 네트워크 프로그래밍과 소켓의 이해

□ 파일의 조작

✓ 저수준 파일 입출력(Low-Level File Access)

- 리눅스 혹은 윈도우즈 자체에서 제공해 주는 파일 입출력 함수를 사용하여 파일을 관리(파일의 생성 및 삭제, 데이터 입력 및 출력) 하는 것을 의미
- 리눅스에서는 모든 것을 파일로 관리
 - 파일, 소켓, 표준 입력, 표준 출력
 - ☞ 주의 : 표준 입출력 함수.
- 파일에 파일 디스크립터를 할당해서 관리
 - 파일 디스크립터는 정수

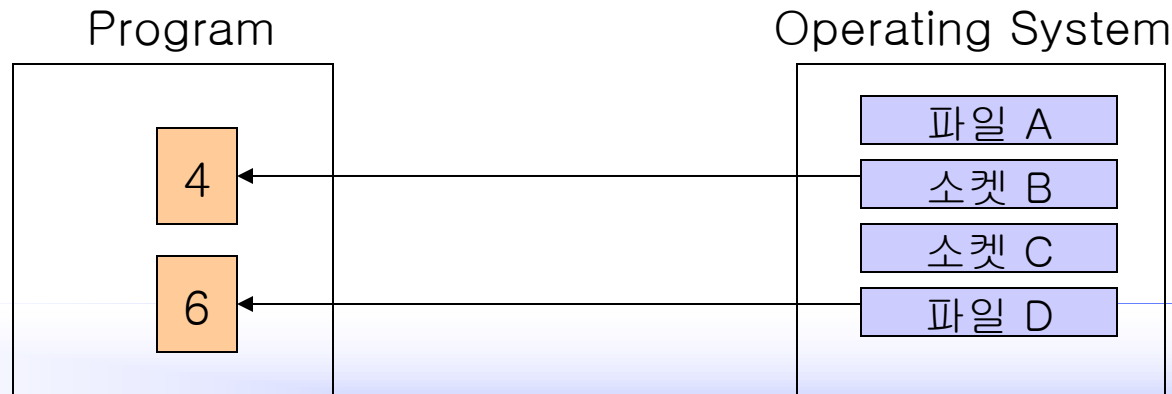
1. 네트워크 프로그래밍과 소켓의 이해

□ 파일의 조작

✓ 파일 디스크립터(File Descriptor)

- 파일을 관리하기 위해서 모든 파일(파일, 소켓 표준 입력, 표준 출력)에 파일 디스크립터를 할당 해 준다.

파일 디스크립터	대 상
0	표준 입력
1	표준 출력
2	표준 에러 출력



1. 네트워크 프로그래밍과 소켓의 이해

□ 파일의 Open 및 Close

```
#include <fcntl.h>  
#include <sys/types.h>  
#include <sys/stat.h>
```

```
int open(const char *path, int flag);
```

```
#include <unistd.h>
```

```
int close(int fildes);
```

1. 네트워크 프로그래밍과 소켓의 이해

□ 파일 오픈 모드

MODE	의 미
O_CREAT	필요한 경우 파일을 생성
O_TRUNC	존재하던 데이터를 모두 삭제
O_APPEND	존재하는 데이터를 보존하고 뒤에 이어서 저장
O_RDONLY	읽기 전용 모드로 파일을 오픈
O_WRONLY	쓰기 전용 모드로 파일을 오픈
O_RDWR	읽기 쓰기 겸용 모드로 파일을 오픈

✓ 하나 이상의 모드를 Bitwise OR (|) 로 묶을 수 있다.

➤ O_CREAT | O_WRONLY | W_TRUNC

- → 생성하고 쓰기 전용 모드로 열며 파일의 기존의 내용을 지운다

1. 네트워크 프로그래밍과 소켓의 이해

❑ 데이터의 Read, Write

```
#include <unistd.h>
```

```
ssize_t write(int fildes, const void * buf, size_t nbytes);
```

```
#include <unistd.h>
```

```
ssize_t read(int fildes, void *buf, size_t nbytes);
```

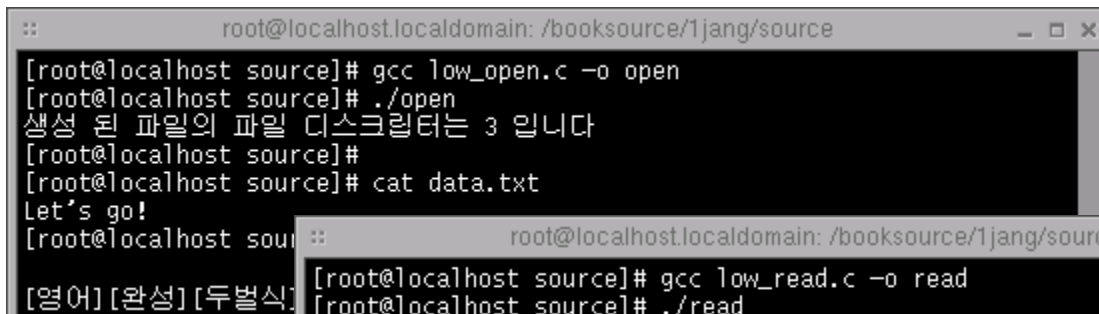
1. 네트워크 프로그래밍과 소켓의 이해

□ 예제 프로그램

1. 프로그램 예제

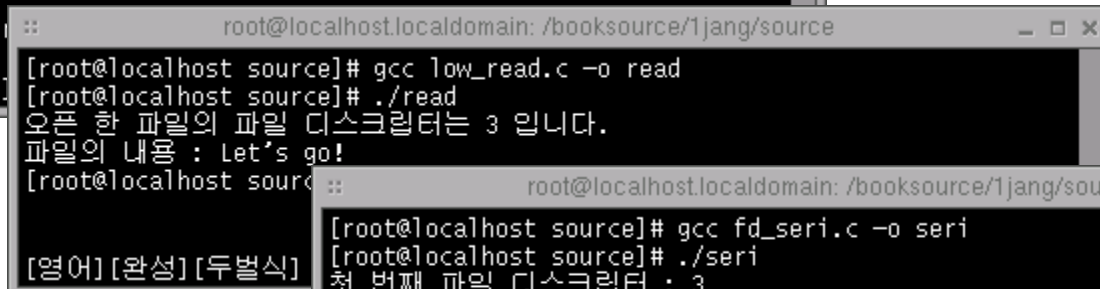
– low_open.c, low_read.c, fd_seri.c

2. 실행하기



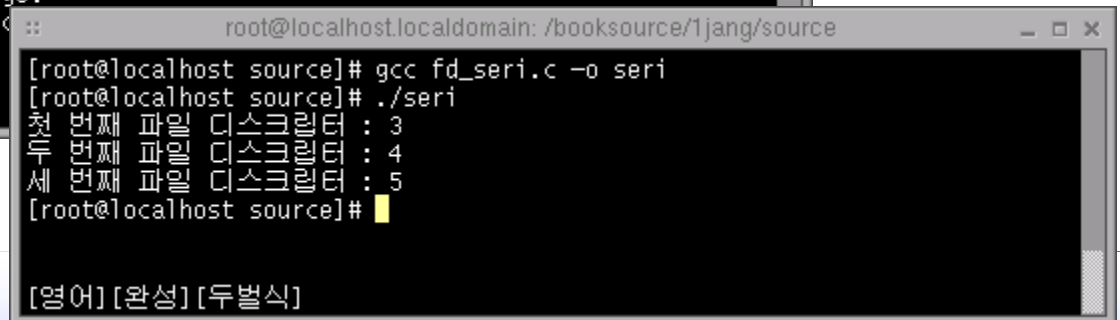
```
root@localhost.localdomain: /booksource/1jang/source
[ root@localhost source ]# gcc low_open.c -o open
[ root@localhost source ]# ./open
생성 된 파일의 파일 디스크립터는 3 입니다
[ root@localhost source ]#
[ root@localhost source ]# cat data.txt
Let's go!
[ root@localhost source ]#
```

< 그림 1-3 >



```
root@localhost.localdomain: /booksource/1jang/source
[ root@localhost source ]# gcc low_read.c -o read
[ root@localhost source ]# ./read
오른 한 파일의 파일 디스크립터는 3 입니다.
파일의 내용 : Let's go!
[ root@localhost source ]#
```

< 그림 1-4 >



```
root@localhost.localdomain: /booksource/1jang/source
[ root@localhost source ]# gcc fd_seri.c -o seri
[ root@localhost source ]# ./seri
첫 번째 파일 디스크립터 : 3
두 번째 파일 디스크립터 : 4
세 번째 파일 디스크립터 : 5
[ root@localhost source ]#
```

< 그림 1-5 >

참고 - `_t`로 끝나는 자료형

□ 고전적인 (primitive) 자료형

- ✓ `ssize_t`, `size_t`, `pid_t`
- ✓ 일반적으로 `<sys/types.h>`에 선언되어 있음
- ✓ 컴퓨터의 성능이 바뀌어도 쉽게 컴파일 할 수 있게 하기 위해 존재
- ✓ 설치하는 기계의 성능이 바뀌었을 때 헤더 파일의 자료형만 재정의
 - 예: 16 bit machine에서 32 bit 또는 64 bit machine으로 바뀐 경우 → `size_t` 만 재정의 해줌

실습

- ❑ File을 open 하고 write 할 때 기존의 내용이 지워지지 않고 새로운 내용이 update 되도록 함
 - ✓ 파일을 오픈할 때 오픈 모드에 O_APPEND가 들어가도록 함
 - low_open.c
- ```
20: fildes=open("data.txt",O_CREAT|O_WRONLY|O_TRUNC);
→ fildes=open("data.txt",O_CREAT|O_WRONLY|O_APPEND);
```

# Q&A

