

인터넷 프로토콜 6장

중급 소켓 프로그래밍 (1)

목차

- ▶ 제 6장 중급 소켓 프로그래밍
 - ▶ 6.1 소켓 옵션
 - ▶ 6.2 시그널
 - ▶ 6.3 년블로킹 입/출력
 - ▶ 6.4 멀티태스킹
 - ▶ 6.5 멀티플렉싱
 - ▶ 6.6 다수의 수신자 처리

소켓 옵션

▶ 소켓 옵션(socket options)

▶ 소켓의 기본 동작을 변경

- ▶ 소켓 코드와 프로토콜 구현 코드에 대한 세부적인 제어 가능

▶ 소켓 옵션 관련 함수

```
#include <sys/types.h>
#include <sys/socket.h>

int setsockopt(int s, int level, int opt, const char *optval, int optlen);
int getsockopt(int s, int level, int opt, const char *optval, int *len);
```

▶ s : 소켓번호

▶ level : 프로토콜 레벨

- ▶ SOL_SOCKET: 소켓의 일반적인 옵션 변경
- ▶ IPPROTO_IP: IP 프로토콜에 관한 옵션 변경
- ▶ IPPROTO_TCP: TCP에 관한 옵션 변경

▶ opt : 사용하고자 하는 옵션

▶ optval : 옵션 지정에 필요한 값의 포인터

▶ optlen : optval의 크기

Socket Option Layer

▶ level

▶ SOL_SOCKET

- ▶ 프로토콜과 무관한 소켓 그 자체

▶ IPPROTO_TCP

- ▶ TCP에 관련된 옵션

▶ IPPROTO_IP

- ▶ IP에 관련된 옵션

SOL_SOCKET

▶ Option name

- ▶ SO_BROADCAST: 방송형 메시지 전송 허용
- ▶ SO_DEBUG: DEBUG 모드를 선택
- ▶ SO_REUSEADDR: 주소 재사용 선택
- ▶ SO_LINGER
 - ▶ 소켓을 닫을 때 미전송된 데이터가 있어도 지정된 시간만큼 기다렸다가 소켓을 닫음
- ▶ SO_KEEPALIVE: TCP의 keep-alive 동작 선택
- ▶ SO_OOBINLINE: OOB 데이터를 일반 데이터처럼 읽음
- ▶ SO_RCVBUF: 수신버퍼의 크기 변경
- ▶ SO_SNDBUF: 송신버퍼의 크기 변경

소켓 옵션 예제

Socket 내부 buffer 변경 (1)

- ▶ TCP, UDP는 송신버퍼와 수신버퍼를 가짐
 - ▶ TCP의 경우 write() 호출 시 데이터를 송신 버퍼로 복사
 - ▶ 데이터가 송신버퍼에 모두 복사되면 시스템이 데이터를 전송
 - ▶ 전송 데이터는 유지하고 있다가 ACK를 수신 후 삭제
 - ▶ 송신버퍼가 가득 차면 write()는 블록됨
 - ▶ 송신/수신버퍼의 크기를 사용자가 지정할 수 있음
- ▶ SO_SNDBUF
 - ▶ 송신 버퍼의 크기 확인 및 지정
- ▶ SO_RCVBUF
 - ▶ 수신 버퍼의 크기 확인 및 지정
- ▶ 송신/수신 버퍼의 크기 지정 방법
 - ▶ 연결설정(3-way handshake) 후에는 버퍼 크기 변경이 불가
 - ▶ 서버의 경우 listen() 호출 이전에 설정
 - ▶ 클라이언트의 경우 connect() 호출 이전에 설정

소켓 옵션 예제

Socket 내부 buffer 변경 (2)

```
int optval;
int optlen = sizeof(optval);
if(getsockopt(listen_sock, SOL_SOCKET, SO_RCVBUF,
    (char *)&optval, &optlen) == SOCKET_ERROR)
    err_quit("getsockopt()");

printf("수신 버퍼 크기 = %d 바이트\n", optval);

optval = 2;
if(setsockopt(listen_sock, SOL_SOCKET, SO_RCVBUF,
    (char *)&optval, sizeof(optval)) == SOCKET_ERROR)
    err_quit("setsockopt()");
```

소켓 옵션 예제

SO_REUSEADDR 옵션 (1)

▶ 용도

- ▶ 사용 중인 IP 주소와 포트 번호를 재사용
 - ▶ 사용 중인 IP 주소와 포트 번호로 bind() 함수를 (성공적으로) 호출할 수 있음

▶ 목적

- ▶ 서버 종료 후 재실행시 bind() 함수에서 오류가 발생하는 것을 방지
 - ▶ fork()의 부모 프로세스 문제 등

소켓 옵션 예제

SO_REUSEADDR 옵션 (2)

```
serv_sock=socket(PF_INET, SOCK_STREAM, 0);
optlen = sizeof(option);
option = TRUE;    // #define TRUE 1
setsockopt(serv_sock, SOL_SOCKET, SO_REUSEADDR,
           &option, sizeof(option));
```

IPPROTO_IP

- ▶ IP_TTL
 - ▶ Time To Live 변경
- ▶ IP_MULTICAST_TTL
 - ▶ 멀티캐스트 데이터그램의 TTL 변경
- ▶ IP_ADD_MEMBERSHIP
 - ▶ 멀티캐스트 그룹에 가입
- ▶ IP_DROP_MEMBERSHIP
 - ▶ 멀티캐스트 그룹에서 탈퇴
- ▶ IP_MULTICAST_LOOP
 - ▶ 멀티캐스트 데이터그램의 loopback 허용 여부
- ▶ IP_MULTICAST_IF
 - ▶ 멀티캐스트 데이터그램 전송용 인터페이스 지정

IPPROTO_TCP

- ▶ TCP_KEEPALIVE
 - ▶ keep-alive 확인 메시지 전송 시간 지정
- ▶ TCP_MAXSEG
 - ▶ TCP의 MSSS(최대 메시지 크기) 지정
- ▶ TCP_NODELAY
 - ▶ Nagle 알고리즘의 선택