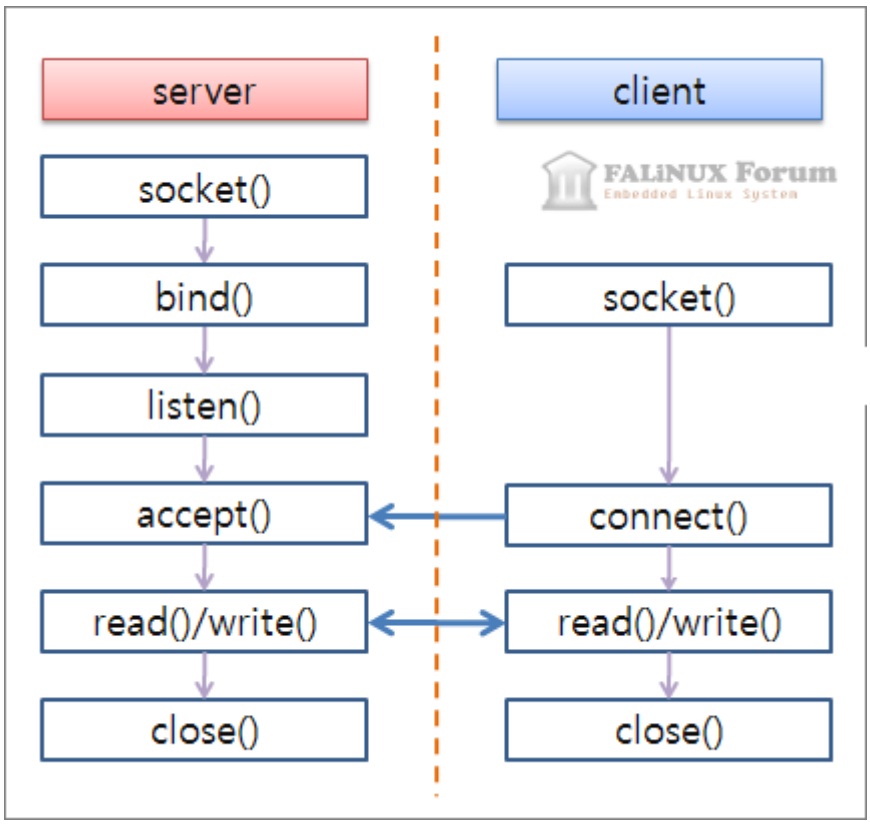
<소켓 프로그래밍의 원리 및 필요 함수>

- 소켓 프로그래밍의 원리



1. 소켓 생성

socket() 함수는 소켓을 생성하여 반환합니다.

헤더	**#include** <sys/types.h> **#include** <sys/socket.h>
형태	**int** socket(**int** domain, **int** type, **int** protocol);
인수	**int** domain : 인터넷을 통해 통신할 지, 같은 시스템 내에서 프로세스 끼리 통신할 지의 여부를 설정합니다.

domain	domain 내용
PF_INET, AF_INET	IPv4 인터넷 프로토콜을 사용합니다.
PF_INET6	IPv6 인터넷 프로토콜을 사용합니다.
PF_LOCAL, AF_UNIX	같은 시스템 내에서 프로세스 끼리 통신합니다.
PF_PACKET	Low level socket 을 인터페이스를 이용합니다.
PF_IPX	IPX 노벨 프로토콜을 사용합니다.

int type : 데이터의 전송 형태를 지정하며 아래와 같은 값을 사용할 수 있습니다.

type	type 내용
SOCK_STREAM	TCP/IP 프로토콜을 이용합니다.
SOCK_DGRAM	UDP/IP 프로토콜을 이용합니다.

int protocol : 통신에 있어 특정 프로토콜을 사용을 지정하기 위한 변수이며, 보통 0 값을 사용합니다.

반환	-1 이외 : 소켓 식별자 -1 : 실패

2. 소켓 주소 할당 및 포트 연결

bind() 함수는 소켓에 IP주소와 포트번호를 지정해 줍니다. 이로서 소켓을 통신에 사용할 수 있도록 준비가 됩니다.

헤더	**#include** <sys/types.h> **#include** <sys/socket.h>
형태	**int** bind(**int** sockfd, **struct** sockaddr *myaddr, socklen_t addrlen);
인수	**int** sockfd : 소켓 디스크립터 **struct** sockaddr *myaddr : 주소 정보로 인터넷을 이용하는 AF_INET인지 시스템 내에서 통신하는 AF_UNIX에 따라서 달라집니다. 인터넷을 통해 통신하는 AF_INET인 경우에는 struct sockaddr_in을 사용합니다. `struct sockaddr_in { sa_family_t sin_family; /* Address family */ unsigned short int sin_port /* Port number */ struct in_addr sin_addr; /* Internet address */ /* Pad to size of 'struct sockaddr'. */ unsigned char __pad[__SOCK_SIZE__ - sizeof(short int) - sizeof(unsigned short int) - sizeof(struct in_addr)]; };` 시스템 내부 통신인 AF_UNIX인 경우에는 struct sockaddr을 사용합니다. `struct sockaddr { sa_family_t sa_family; /* address family, AF_xxx */ char sa_data[14]; /* 14 bytes of protocol address */ };` socklen_t addrlen : myadd 구조체의 크기
반환	0 : 성공 -1 : 실패

3. 클라이언트 접속 요청 설정

listen() 함수는 소켓을 통해 클라이언트의 접속 요청을 기다리도록 설정합니다.

헤더	**#include** <sys/socket.h>
형태	**int** listen(**int** s, **int** backlog);
인수	**int** s : 소켓 디스크립터 **int** backlog : 대기 메시지 큐의 개수
반환	0 : 성공 -1 : 실패

4. 서버로 접속 요청

connect() 함수는 생성한 소켓을 통해 서버로 접속을 요청합니다.

헤더	**#include** <sys/types.h> **#include** <sys/socket.h>
형태	**int** connect(**int** sockfd, **const struct** sockaddr *serv_addr, socklen_t addrlen);
인수	**int** sockfd : 소켓 디스크립터 **struct** sockaddr *serv_addr : 서버 주소 정보에 대한 포인터 socklen_t addrlen : **struct** sockaddr *serv_addr 포인터가 가르키는 구조체의 크기
반환	0 : 성공 -1 : 실패

5. 클라이언트 접속 요청 수락

accept() 함수는 클라이언트의 접속 요청을 받아드리고 클라이언트와 통신하는 전용 소켓을 생성합니다.

헤더	**#include** <sys/types.h> **#include** <sys/socket.h>
형태	**int** accept(**int** s, **struct** sockaddr *addr, socklen_t *addrlen);
인수	**int** s : 소켓 디스크립터 **struct** sockaddr *addr : 클라이언트 주소 정보를 가지고 있는 포인터 socklen_t addrlen : **struct** sockaddr *addr 포인터가 가르키는 구조체의 크기
반환	-1 이외 : 새로운 소켓 디스크립터 -1 : 실패

6. 연결된 서버 및 클라이언트로 데이터 전송

send() 함수는 연결된 서버나 클라이언트로 데이터를 전송합니다.

헤더	**#include** <sys/types.h> **#include** <sys/socket.h>
형태	**int** send(**int** s, **const void** *msg, size_t len, **int** flags);
인수	**int** s : 소켓 디스크립터 **void** *msg : 전송할 데이터 size_t len : 데이터의 바이트 단위 길이 **int** flags : 아래와 같은 옵션을 사용할 수 있습니다.
반환	-1 이외 : 실제 전송한 바이트 수 -1 : 실패

flags	옵션 설명
MSG_DONTWAIT	전송할 준비가 전에 대기 상태가 필요하다면 기다리지 않고 -1을 반환하면서 복귀
MSG_NOSIGNAL	상대방과 연결이 끊겼을 때, SIGPIPE 시그널을 받지 않도록 합니다.

7. 연결된 서버 및 클라이언트로부터 데이터 수신

recv() 함수는 소켓으로부터 데이터를 수신합니다.

헤더	**#include** <sys/types.h> **#include** <sys/socket.h>
형태	**int** recv(**int** s, **void** *buf, size_t len, **int** flags);
인수	**int** s : 소켓 디스크립터 **void** *buf : 수신할 버퍼 포인터 데이터 size_t len : 버퍼의 바이트 단위 크기 **int** flags : 아래와 같은 옵션을 사용할 수 있습니다.
반환	-1 이외 : 실제 수신한 바이트 수 -1 : 실패

flags	옵션 설명
MSG_DONTWAIT	수신을 위해 대기가 필요하다면 기다리지 않고 -1을 반환하면서 바로 복귀
MSG_NOSIGNAL	상대방과 연결이 끊겼을 때, SIGPIPE 시그널을 받지 않도록 합니다.

출처 : http://forum.falinux.com/zbxe/