**SOCKET**

**□Class 구성**

- Variable

└ **m\_sockFD** – 소켓을 생성하면 반환되는 파일 디스크립터 값을 저장

└ **m\_addrSize** – 소켓 구조체의 사이즈를 저장

└ **m\_sSockAddr** – 서버 소켓 주소 구조체를 저장

└ **m\_cSockAddr** – 클라이언트 소켓 주소 구조체를 저장

└ **m\_mreq** – 멀티캐스트 관련 구조체를 저장

- Function

└ **CSocket**

– 소켓 클래스 생성자(변수 초기화)

└ **~CSocket**

– 소켓 클래스 소멸자(Destroy 함수 호출)

└ **Create**

- 소켓 생성

Parameter

- int sockType: 소켓 생성 타입(TCP or UDP)

1. socket(소켓 생성 함수)를 사용하여 소켓을 생성

ex) socket(AF\_INET, sockType, 0)

2. 성공 시 파일 디스크립터를 반환하고 실패 시 -1을 반환

└ **Destroy**

– 소켓 제거

1. close(파일 닫기 함수)를 사용하여 소켓을 제거

2. 변수 초기화

└ **IsCreated**

– 소켓 생성 확인

1. 소켓 파일 디스크립터를 확인하여 -1이 아닌경우 true를 반환 -1이면 false를 반환

└ **SetServerAddr**

– 서버 소켓 주소 구조체 IP, PORT 설정

Parameter

- int port: 소켓으로 통신하려는 포트번호

- in\_addr\_t addr: 소켓으로 통신하려는 주소(inet\_addr 또는 htonl 함수를 사용)

1. memset(메모리의 일부분 값을 지정 함수)를 사용하여 m\_sSockAddr 구조체를 초기화

2. m\_sSockAddr 구조체의 주소 체계, IP, PORT를 설정 후 반환

└ **SetClientAddr**

– 클라이언트 소켓 주소 구조체 사이즈 설정

1. m\_cSockAddr 구조체 크기를 m\_addrSize 변수에 저장

2. m\_cSockAddr 구조체를 반환

└ **SetBroadcast**

– 브로드 캐스트 설정

1. m\_broadcast 변수를 1값(허용)으로 선언

2. setsockopt(소켓의 옵션을 설정하는 함수)를 사용하여 생성된 소켓이 브로드 캐스트 기능을 하도록 설정

ex) setsockopt(m\_sockFD, SOL\_SOCKET, SO\_BROADCAST, &m\_broadcast, sizeof(m\_broadcast))

└ **SetMulticast**

– 멀티 캐스트 설정

1. m\_mreq 구조체의 멀티 캐스트 IP, 인터페이스의 IP를 설정

2. setsockopt(소켓의 옵션을 설정하는 함수)를 사용하여 생성된 소켓이 멀티 캐스트 기능을 하도록 설정

└ **Bind**

– 소켓 결속

1. bind(소켓을 결속하는 함수)를 사용하여 서버에 소켓을 결속

ex) bind(m\_sockFD, (struct sockaddr\*)&m\_sSockAddr, sizeof(m\_sSockAddr))

2. 성공 시 0을 반환하고 실패 시 -1을 반환

└ **Listen**

- 소켓 대기

Parameter

- int nListen: 소켓 대기 가능 개수

1. listen(소켓을 대기 상태로 설정하는 함수)를 사용하여 소켓을 대기 상태로 설정

ex) listen(m\_sockFD, nListen)

2. 성공 시 0을 반환하고 실패 시 -1을 반환

└ **Accept**

– 소켓 연결 허용

1. accept(소켓 연결을 허용하는 함수)를 사용하여 서버로 연결되는 소켓을 허용

ex) accept(m\_sockFD, (struct sockaddr\*)&m\_cSockAddr, (socklen\_t\*)&m\_addrSize)

2. 성공 시 새로운 소켓 디스크립터를 반환하고 실패 시 -1을 반환

└ **Connect**

– 소켓 연결

1. connect(소켓을 연결하는 함수)를 사용하여 클라이언트가 설정된 서버의 주소로 연결

ex) connect(m\_sockFD, (struct sockaddr\*)&m\_sSockAddr, sizeof(m\_sSockAddr))

2. 성공 시 0을 반환하고 실패 시 -1을 반환

└ **Send**

– TCP 데이터 전송

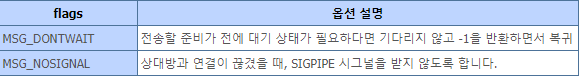
Parameter

- int sockFD: 연결된 소켓의 파일 디스크립터

- void\* pData: 복사할 버퍼의 주소

- int size: 복사할 사이즈

- int flag: DEFAULT는 0



1. send(TCP/IP에서 소켓으로 데이터를 전송하는 함수)를 사용하여 버퍼의 지정된 사이즈만큼 데이터를 전송한다.

ex) send(sockFD, pData, size, flag)

2. 성공 시 전송한 바이트 수를 반환하고 실패 시 -1을 반환

└ **Recv**

– TCP 데이터 수신

Parameter

- int sockFD: 연결된 소켓의 파일 디스크립터

- void\* pData: 복사될 버퍼의 주소

- int size: 복사될 사이즈

- int flag: DEFAULT는 0, send의 flag와 내용이 동일

1. recv(TCP/IP에서 소켓으로부터 데이터를 수신하는 함수)를 사용하여 복사될 버퍼에 데이터의 지정된 사이즈만큼 복사

ex) recv(sockFD, pData, size, flag)

2. 성공 시 수신한 바이트 수를 반환하고 실패 시 -1을 반환

└ **SendTo**

– UDP 데이터 전송

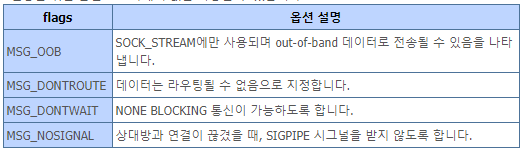
Parameter

- sockaddr\_in\* sockAddr: 연결된 송/수신자의 주소 체계, IP, PORT가 설정된 구조체

- void\* pData: 복사할 버퍼의 주소

- int size: 복사할 사이즈

- int flag: DEFAULT는 0



1. sendto(UDP/IP에서 소켓으로 데이터를 전송하는 함수)를 사용하여 버퍼의 지정된 사이즈만큼 송/수신자의 주소로 데이터를 전송한다.

ex) sendto(m\_sockFD, pData, size, flag, (struct sockaddr\*)sockAddr, sizeof(\*sockAddr))

2. 성공 시 전송한 바이트 수를 반환하고 실패 시 -1을 반환

└ **RecvFrom**

– UDP 데이터 수신

Parameter

- sockaddr\_in\* sockAddr: 연결된 송/수신자의 주소 체계, IP, PORT가 설정된 구조체

- void\* pData: 복사될 버퍼의 주소

- int size: 복사될 사이즈

- int flag: DEFAULT는 0, sendto와 내용이 동일

1. recvfrom(UDP/IP에서 소켓으로부터 데이터를 수신하는 함수)를 사용하여 복사될 버퍼에 데이터의 지정된 사이즈만큼 복사

ex) recvfrom(m\_sockFD, pData, size, 0, (struct sockaddr\*)sockAddr, &m\_addrSize)

2. 성공 시 수신한 바이트 수를 반환하고 실패 시 -1을 반환