**SharedMemory (POSIX)**

**□Class 구성**

- variable

└ **m\_nameShm** – 공유 메모리를 생성할 때 KEY 값이 되는 이름 값을 저장

└ **m\_fdShm** – 공유 메모리를 생성하면 반환되는 파일 디스크립터 값을 저장

└ **m\_sizeShm** – 공유 메모리의 사이즈를 저장

└ **m\_bufShm** – 공유 메모리의 버퍼 주소를 저장

- Function

└ **CSharedMemory**

– 공유 메모리 클래스 생성자(변수 초기화)

└ **~CSharedMemory**

– 공유 메모리 클래스 소멸자(Destroy 함수 호출)

└ **Create**

Parameter

- char\* name: 공유 메모리를 생성할 때 KEY 값이 되는 이름 값

- unsigned int size: 공유 메모리의 사이즈 값

1. shm\_open(공유 메모리를 생성하는 함수)를 사용하여 공유 메모리를 생성

ex) shm\_open(name, O\_CREAT | O\_RDWR, 0666)

2. 공유 메모리 성공 시 파일 디스크립터를 반환하고 실패 시 -1을 반환

3. furuncate(파일을 지정 크기로 변경하는 함수)를 사용하여 공유 메모리의 사이즈를 설정

4. 파일 크기 변경 성공 시 0을 반환하고 실패 시 -1을 반환

5. mmap(파일이나 장치를 메모리에 연결하는 함수)를 사용하여 공유 메모리를 연결하며 반환 값은 공유 메모리의 버퍼 주소

ex) mmap(NULL, m\_sizeShm, PROT\_READ | PROT\_WRITE, MAP\_SHARED, m\_fdShm, 0)

6. 연결 성공 시 메모리 주소를 반환하고 실패 시 errno(MAP\_FAILED)을 반환

└ **Destroy**

1. munmap(파일이나 장치를 메모리에서 연결 해제하는 함수)를 사용하여 공유 메모리를 연결 해제

2. shm\_unlink(공유 메모리를 제거하는 함수)를 사용하여 공유 메모리를 제거

3. 변수 초기화

└ **ReadMemory**

Parameter

- void\* pData: 복사할 버퍼의 주소

- unsigned int size: 복사할 사이즈

1. 공유 메모리의 사이즈보다 복사할 버퍼의 사이즈가 작을 경우만 실행

2. memcpy(메모리의 일부분을 복사하는 함수)를 사용하여 공유 메모리에 복사할 버퍼의 지정된 사이즈만큼 복사

└ **WriteMemory**

Parameter

- void\* pData: 복사될 버퍼의 주소

- unsigned int size: 복사될 사이즈

1. 공유 메모리의 사이즈보다 복사될 버퍼의 사이즈가 작을 경우만 실행

2. memcpy(메모리의 일부분을 복사하는 함수)를 사용하여 복사될 버퍼에 공유 메모리의 지정된 사이즈만큼 복사