

04. Array

(배열 사용시 주의해야 할 사항)



C++ 프로그래밍

국민대학교
소프트웨어학부

Array (배열)

- C, C++에서 배열 사용할 때 주의할 점
 - 배열 index 의 범위(range)를 검사하지 않음
 - C 는 UNIX OS를 개발하기 위한 언어
 - OS는 수행 속도가 가장 중요한 요소임
 - 따라서, C 에서 수행 속도를 느리게 하는 많은 기능을 제거함
 - 배열 index의 범위(range)를 검사하는 기능도 제거된 기능임
 - C/C++ 프로그램에서 배열을 많이 사용하므로, 이 기능을 제거하면 수행 속도를 높일 수 있음

Array (배열)

- C, C++에서 배열 사용할 때 주의할 점
 - 배열 index 의 범위(range)를 검사하지 않음

```
const int SIZE = 4;
int arr[SIZE];
int i;          // index
...
    for(i=0; i ... )
        ... arr[i] ...
...
```

배열 index 변수 **i** 의 범위(range) : 0 ~ 3

배열 index의 범위를 검사하는 기능:

- 배열 arr 의 index 변수 i 의 값이 0~3 인 경우에는 OK, 계속 정상수행
- 그렇지 않으면 프로그램 수행 중지

Array (배열)

- C, C++에서 배열 사용할 때 주의할 점
 - 따라서, 배열 index 값을 범위(range)내로 유지시키는 역할은 프로그래머에게 주어짐.
 - 즉, 프로그래머에서 빠른 속도로 수행되는 프로그램을 작성할 수 있는 기능을 제공하는 반면에 배열 index 값을 범위(range)내로 유지시키는 검증의 임무를 프로그래머에게 맡김.

Array (배열)

- C, C++에서 배열 사용할 때 주의할 점
 - C/C++ 프로그램에서 배열을 사용할 경우에, 가장 많은 실수는 배열 index가 범위(range)를 벗어나는 경우에 있음
 - 프로그래머가 제 책임을 다하지 않았음
 - 운이 좋다면, 아무런 문제가 발생하지 않을 수 있지만
 - 대부분 가혹한 결과가 기다리고 있음
 - 로켓이 발사되는 도중에 폭발할 수 있음

Array (배열)

- C, C++에서 배열 사용할 때 주의할 점

```
#include <iostream>
using namespace std;
const int SIZE = 4;
```

```
int main()
{
```

```
    int iWord = 1;
    int iArr1[SIZE] = {2, 3, 4, 5};
    int iArr2[SIZE] = {6, 7, 8, 9};
```

```
    iArr1[-1] = 10;    // iArr2[-1] = 10;
    iArr1[4] = 11;     // iArr2[4] = 11;
```

```
    cout << iWord << " " << iArr2[0] << endl;    // iArr1[0]
```

```
    return 0;
```

```
}
```

출력값은?

1. 1 6
2. 10 11
3. 1 2
4. 시스템마다 다름

Array (배열)

- C, C++에서 배열 사용할 때 주의할 점

```
#include <iostream>
using namespace std;
const int SIZE = 4;
```

outOfRange.cpp

```
int main()
{
    int iWord = 1;
    int iArr1[SIZE] = {2, 3, 4, 5};
    int iArr2[SIZE] = {6, 7, 8, 9};

    iArr1[-1] = 10;
    iArr1[4] = 11;

    cout << iWord << " " << iArr2[0] << endl;

    cout << &iWord << endl;          /* 각 변수의 주소값 출력 */
    cout << &iArr1[0] << endl;
    cout << &iArr2[0] << endl;
    cout << &iArr1[-1] << endl;
    cout << &iArr1[4] << endl;
    return 0;
}
```

Array (배열)

- Ubuntu 에서 실행한 결과

```
#include <iostream>
using namespace std;
const int SIZE = 4;

int main()
{
    int iWord = 1;
    int iArr1[SIZE] = {2, 3, 4, 5};
    int iArr2[SIZE] = {6, 7, 8, 9};

    iArr1[-1] = 10;
    iArr1[4] = 11;

    cout << iWord << " " << iArr2[0]

    cout << &iWord << endl;
    cout << &iArr1[0] << endl;
    cout << &iArr2[0] << endl;
    cout << &iArr1[-1] << endl;
    cout << &iArr1[4] << endl;
    return 0;
}
```

outOfRange.cpp

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
outOfRange.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ g++ -o outOfRange outOfRange.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
outOfRange  outOfRange.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ./outOfRange
10 11
0x7fff499ce2dc
0x7fff499ce2e0
0x7fff499ce2f0
0x7fff499ce2dc
0x7fff499ce2f0
jschoi@jschoi-VirtualBox:~/바탕화면$
```


Array (배열)

- Ubuntu 에서 실행한 결과

outOfRange.cpp

```
int iWord = 1;
int iArr1[SIZE] = {2, 3, 4, 5};
int iArr2[SIZE] = {6, 7, 8, 9};

iArr1[-1] = 10;
iArr1[4] = 11;
```

주소수식

주소값

&iWord 0x7fff499ce3dc

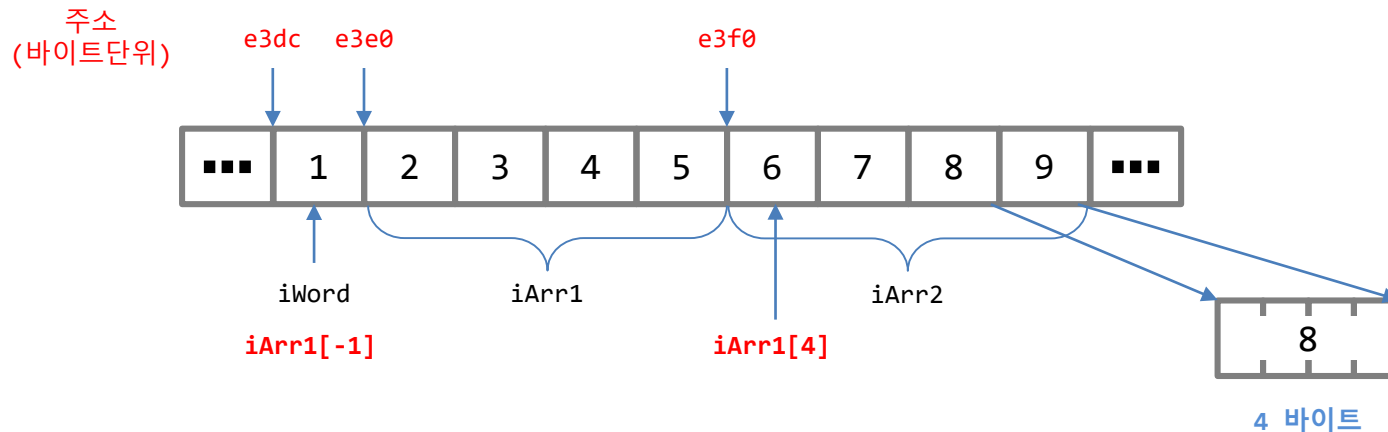
&iArr1[0] 0x7fff499ce3e0

&iArr2[0] 0x7fff499ce3f0

&iArr1[-1] 0x7fff499ce3dc

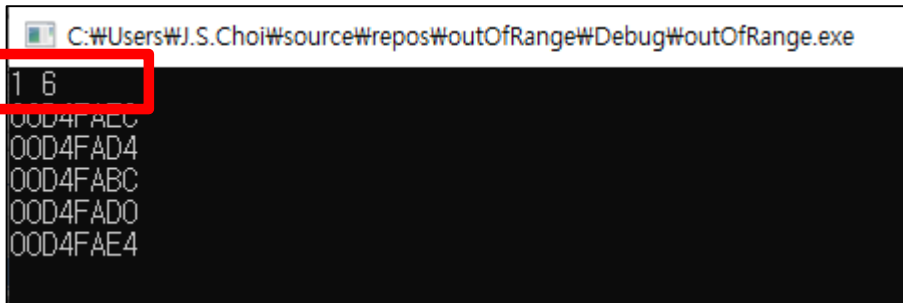
&iArr1[4] 0x7fff499ce3f0

(참고) 주소는 수행할 때마다 바뀔 수 있음

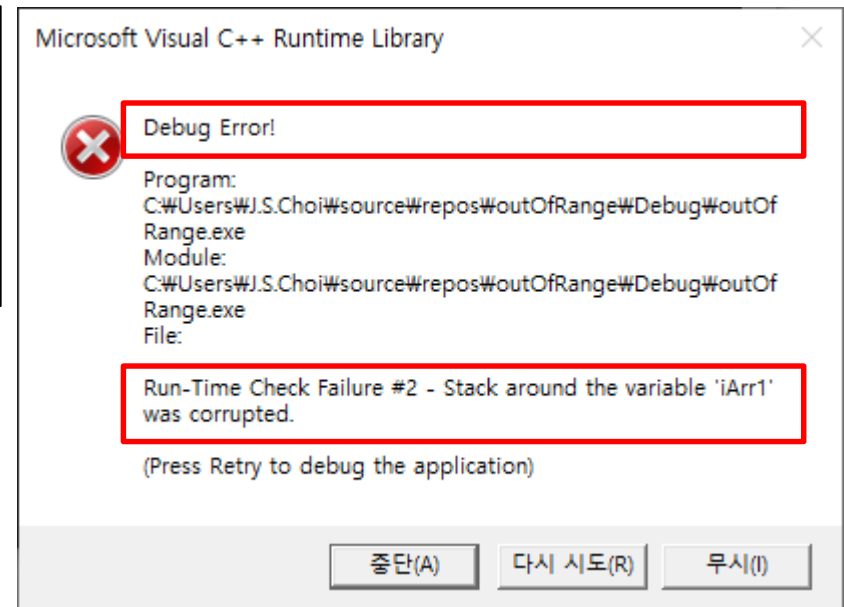


Array (배열)

- 같은 프로그램을 MS Visual Studio로 수행한 결과
 - (Release Mode) 프로그램 실행 중단
 - (Debug Mode) 아래와 같은 Run-Time Error 발생



```
C:\Users\WJ.S.Choi\source\repos\outOfRange\Debug\outOfRange.exe
1 6
00D4FAEC
00D4FAD4
00D4FABC
00D4FAD0
00D4FAE4
```



Array (배열)

- 같은 프로그램을 MS Visual Studio로 수행한 결과

outOfRange.cpp

```
int iWord = 1;  
int iArr1[SIZE] = {2, 3, 4, 5};  
int iArr2[SIZE] = {6, 7, 8, 9};  
  
iArr1[-1] = 10;  
iArr1[4] = 11;
```

주소수식

주소값

&iWord

0x00D4FAEC

&iArr1[0]

0x00D4FAD4

&iArr2[0]

0x00D4FABC

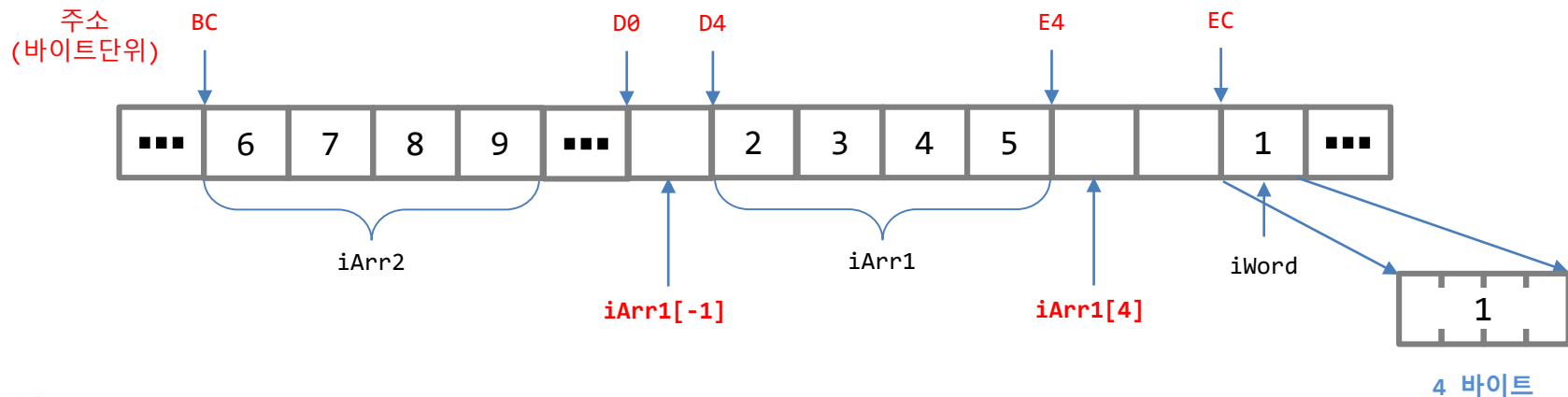
&iArr1[-1]

0x00D4FAD0

&iArr1[4]

0x00D4FAE4

(참고) 주소는 수행할 때마다 바뀔 수 있음



Array (배열)

- C, C++에서 배열 사용할 때 주의할 점 (결론)
 - 프로그램에서 배열 index가 범위(range)를 벗어나는 부분이 있을 경우
 - 어떤 시스템에서는 정상적으로 동작할 수 있음
 - 또 다른 시스템에서는 프로그램 오류로 프로그램이 중지될 수 있음
 - 따라서, 어떻게 동작할 지에 대해서는 알 수 없음
 - 오류에 대한 모든 책임은 프로그래머에게 있음
 - 초급/중급 프로그래머가 배열을 사용할 경우에 발생하는 가장 많이 발생하는 그리고 심각한 오류 중의 하나임

Array (배열)

- C, C++에서 배열 사용할 때 전형적인 오류의 예

```
#include <iostream>
using namespace std;
const int SIZE = 100;

int main()
{
    int iArr[SIZE];

    ...
    for(int i=0; i<=SIZE; i++)
        iArr[i] = ...
    ...

    return 0;
}
```

어느 곳이 잘못?

Array (배열)

- C, C++에서 배열 사용할 때 전형적인 오류의 예

```
const int SIZE = 100;

int main()
{
    int iArr[SIZE];

    ...
    for(int i=0; i<SIZE; i++)
        iArr[i] = iArr[i+1];
    ...
    for(int i=0; i<SIZE; i++)
        iArr[i-1] = iArr[i];
    ...

    return 0;
}
```

어느 곳이 잘못?