

시스템 프로그래밍 2 장



과목명	시스템 프로그래밍
담당교수	박문주
학과	컴퓨터공학부
학년	3 학년
학번	201701524
이름	강은선
제출일	2019.09.16

2.60

<소스코드>

```
#include <stdio.h>

unsigned replace_byte(unsigned x, int i, unsigned char b)
{
    //change x to character, we can replace x's specific byte.
    unsigned char* p = &x;

    //Little Endian
    *(p + i) = b;

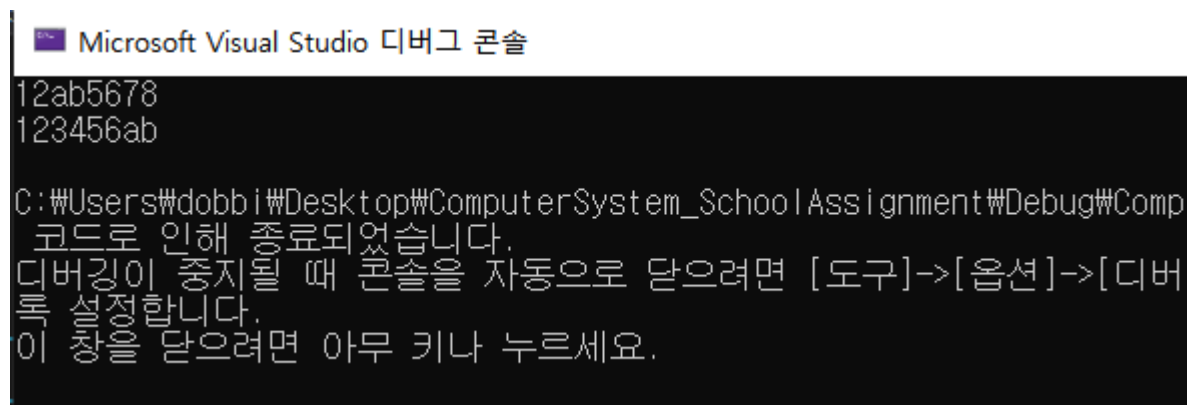
    //Big Endian
    /*(p + 3 - i) = b;

    return x;
}

int main(void)
{
    printf("%0x\n", replace_byte(0x12345678, 2, 0xAB));
    printf("%0x\n", replace_byte(0x12345678, 0, 0xAB));

    return 0;
}
```

<실행 결과>



Microsoft Visual Studio 디버그 콘솔

```
12ab5678
123456ab

C:\Users\#dobb\i\Desktop\ComputerSystem_School\Assignment\Debug\Comp
코드로 인해 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

2.64

<소스코드>

```
#include <stdio.h>

int any_odd_one(unsigned x)
{
    //show as bit, mask = 101010101010101010101010101010 (odd bit is all equal 1)
    unsigned mask = 0xaaaaaaaa;

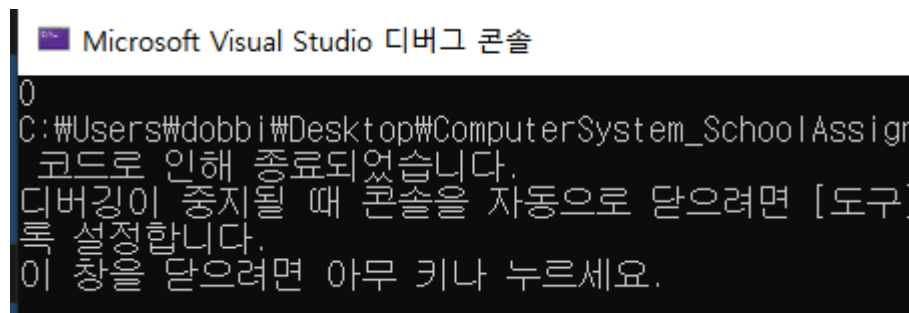
    //when x & mask, if any outcome has value, x has 1 in its odd bit.
    //if data has value, its "not" is 0, 0's "not" is 1. so if double "not" at value, it becomes 1.
    //double "not" at zero, it becomes 0 (0 -> 1 -> 0)

    return !(x & mask);
}

//Test
int main(void)
{
    unsigned x = 0x55555555;

    printf("%d", any_odd_one(x));
}
```

<실행 결과>



Microsoft Visual Studio 디버그 콘솔

```
0
C:\Users\dobbi\Desktop\ComputerSystem_School\Assign
코드로 인해 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]
를 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

2.67

<1>

: 정수타입을 시프트 할 때, 한번에 정수 크기 이상의 시프트가 불가능 한데, 코드의 6 번째 줄에서 beyond_msb 를 한번에 32 시프트 했기 때문이다.

<2>

```
int bad_int_size_is_32() {
    int set_msb = 1 << 31;

    //if we follow bit level integer coding rules
    int beyond_msb = set_msb << 1;
    // = 1 << 31 << 1

    //if we don't
    //int beyond_msb = set_msb * 2;

    return set_msb && !beyond_msb;
}
```

<3>

```
int bad_int_size_is_32_for_16bit() {
    int set_msb = 1 << 15 << 15 << 1;

    int beyond_msb = set_msb << 1;
    // = 1 << 15 << 15 << 1 << 1

    //int beyond_msb = set_msb * 2;

    return set_msb && !beyond_msb;
}
```

2.74

<소스코드>

```
#include <stdio.h>
#include <limits.h>

/*Return most significant bit*/
int msb(int x)
{
    return (x >= 0) ? 0 : 1;
}

/*
*this function determine whether arguments can be subtracted without overflow
*
*@param x          front one in the formula
*@param y          the other one in the formula
*
*@return int        if this operation causes overflow, return 1, otherwise return 0.
*/
int tsub_ok(int x, int y)
{
    //1. x = plus      y = plus      => nothing
    //2. x = plus      y = minus     => msb(x) != msb(x+y) (except, -8)
    //3. x = minus y = plus      => same with case 2.
    //4. x = minus y = minus     => same with case 1.

    return (((msb(x) != msb(y)) && (msb(x - y) != msb(x))) || (y == INT_MIN));
}

int main()
{
    //it's overflow
    int x1 = INT_MAX;
    int y1 = INT_MIN;

    //it's overflow
    int x2 = 1;
    int y2 = INT_MIN + 1;

    //it's not overflow
    int x3 = INT_MIN + 1;
    int y3 = 1;

    //it's overflow
    int x4 = INT_MIN + 1;
    int y4 = 2;

    printf("case 1 result is %d. it is W"%dW"Wn", x1 - y1, tsub_ok(x1, y1));
    printf("case 2 result is %d. it is W"%dW"Wn", x2 - y2, tsub_ok(x2, y2));
}
```

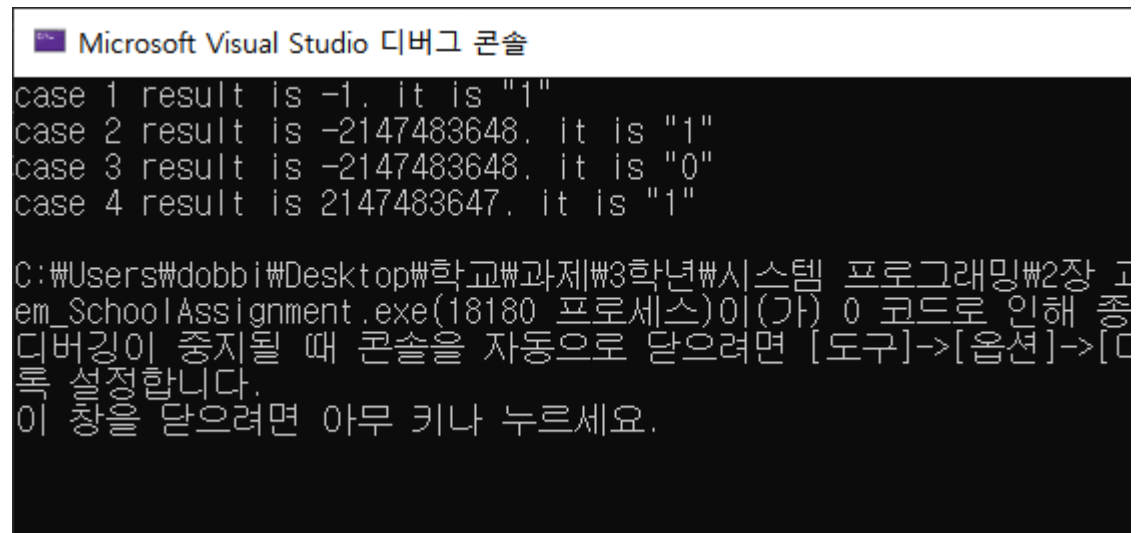
```

printf("case 3 result is %d. it is W"%dW"Wn", x3 - y3, tsub_ok(x3, y3));
printf("case 4 result is %d. it is W"%dW"Wn", x4 - y4, tsub_ok(x4, y4));

return 0;
}

```

<실행 결과>



Microsoft Visual Studio 디버그 콘솔

```

case 1 result is -1. it is "1"
case 2 result is -2147483648. it is "1"
case 3 result is -2147483648. it is "0"
case 4 result is 2147483647. it is "1"

```

C:\Users\dobbi\Desktop\학교과제\3학년\시스템 프로그래밍\2장 고
 em_SchoolAssignment.exe(18180 프로세스)이(가) 0 코드로 인해 종
 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디
 록 설정합니다.
 이 창을 닫으려면 아무 키나 누르세요.

2.76

<소스코드>

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>

#define assert(condition)                                W
do {
    W
    if ((condition)) {
        W
        printf("Test success: %s\n", #condition);        W
    } else {
        W
        printf("Test FAIL: %s\n", #condition);            W
    }
    W
} while(0)

void* mock_calloc(size_t nmemb, size_t size);
int msb(int x);
int size_t_multi_ok(size_t x, size_t y);

int main()
{
    //It is overflow
    assert(mock_calloc((size_t)1 << 63, (size_t)1 << 63) == NULL);

    //size is 0
    assert(mock_calloc(0, 5) == NULL);

    //size is 0
    assert(mock_calloc(5, 0) == NULL);

    //well done!
    assert(mock_calloc(1234, 3) != NULL);
}

/*Implement calloc
*
*@param nmemb  a number of data type
*@param size   size of data type
*
*@return                               pointer of allocated data
*/
void* mock_calloc(size_t nmemb, size_t size)
{
    //if size multipliance is overflow, return NULL
    //if size of nmemb is 0, return NULL
```

```

    if (nmemb == 0 || size == 0 || !(size_t_multi_ok(nmemb, size)))
        return NULL;

    void* data = malloc(nmemb * size);

    memset(data, 0, nmemb * size);

    return data;
}

/*Return most significant bit*/
int msb(int x)
{
    return (x >= 0) ? 0 : 1;
}

/*
 *this function determine whether unsigned arguments can be multiplied without overflow
 *
 *@param x      the front data in the operation
 *@param y      next data in the operation
 *
 *@return        if operation success without overflow, return 0, otherwise return 1.
 */
int size_t_multi_ok(size_t x, size_t y)
{
    return (x == 0 || (x * y) / x == y);
}

```

<실행 결과>

Microsoft Visual Studio 디버그 콘솔

```

Test success: mock_calloc((size_t)1 << 63, (size_t)1 << 63) == NULL
Test success: mock_calloc(0, 5) == NULL
Test success: mock_calloc(5, 0) == NULL
Test success: mock_calloc(1234, 3) != NULL

C:\Users\dobbi\Desktop\학교\과제\3학년\시스템 프로그래밍\2장 과제\ComputerSystem_SchoolAssignment.exe(15592 프로세스)이(가) 0 코드로 인해 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버깅]->[디버깅이 중지되면 콘솔 닫기]를 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.

```


2.80

<소스코드>

```
#include <stdio.h>

#define assert(condition)                                W
do {
    W
    if ((condition)) {
        W
        printf("Test success: %s\n", #condition);        W
    } else {
        W
        printf("Test FAIL: %s\n", #condition);            W
    }
    W
} while(0)

int main(void)
{
    assert(threefourths(2147483647) == 1610612735);
}

/*
 *this function operate 3/4x
 *
 *@param x
 *
 *@return      3/4x
 */
int threefourths(int x)
{
    //first, compare (x / 4 * 4) and x
    //second, x / 4 and * 3
    //plus the distance of first comparison / 4 * 3

    int comparison = x / 4 * 4;
    int before = x / 4 * 3;

    return before + (x - comparison) * 3 / 4;
}
```

<실행결과>

Microsoft Visual Studio 디버그 콘솔

```
Test success: threefourths(2147483647) == 1610612735
```

```
C:\Users\dobbi\Desktop\학교\과제\3학년\시스템 프로그래밍\2장 과제\ComputerSystem_SchoolAssignment.exe(14440 프로세스)이(가) 0 코드로 인해 종료되었습니다.  
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버깅]->[디버깅이  
록 설정합니다.  
이 창을 닫으려면 아무 키나 누르세요.
```