EDDI
Electronic Design
Development Institute

에디로봇아카데미
임베디드 마스터 Lv1 과정

제 3기

2022. 01. 07

김원석

CONTENTS

EDDI
Electronic Design Development Institute

- x & ~(2의 y승 - 1)의 의미
- 변수의 정확한 정의
- 16진수 체계
- xor를 이용한 대소문자 변환 방식
- Memory 구조(Stack, Heap, Data, Text)
- gdb를 사용하여 디버깅 하기 위해 필요한 컴파일 옵션
- 변수의 주소값과 실제 메모리의 값이 일치하는지 확인(디버깅)
- 두 개의 변수를 입력받아 대소 비교 프로그래밍
- 피보나치 수열을 표현할 수 있도록 프로그래밍
- 주사위 게임 프로그래밍

x & ~(2의 y승 - 1)의 의미

EDDI
Electronic Design Development Institute

- 2의 y승 -1을 이진수로 나타내면 y-1만큼 1이 존재한다. (y는 0보다 큼)

  ex) 2^3 -1 = 0b11, 2^5 -1 = 0b1111
- 여기서 NOT(~)연산을 해주면 y-1만큼 0이 존재하게 된다.

  ex) ~0b11 = 0b000, ~0b1111 = 0b00000
- 이것을 임의의 수 x와 AND 연산을 하게 되면 최하위 비트부터 y-1개만큼은 무조건 0이 된다.
- 따라서 연산 결과는 y번째 비트에 해당하는 값의 배수로 나타나게 된다.
- x & ~(2^3 -1) -> 8의 배수, x & ~(2^5 -1) -> 32의 배수

x & ~(2의 y승 - 1)의 의미

EDDI
Electronic Design Development Institute

```c
#include <time.h>
#include <stdio.h>
#include <stdlib.h>

#define MAX                     10
#define BIT_ALIGN       32

int main(void)
{
        int i, num;

        srand(time(NULL));

        for (i = 0; i < MAX; i++)
        {
                // 501 ~ 1000
                num = rand() % 500 + 501;
                // AND NOT
                printf("정렬전 결과 = %d, bit 정렬 결과 = %d\n", num, num & ~(BIT_ALIGN - 1));
        }

        return 0;
}
```

정렬 결과 32의 배수들만 나타나게 된다.

변수의 정확한 정의

EDDI
Electronic Design Development Institute

- 컴퓨터에서 사용하는 모든 정보는 메모리에 로드(적재)된다.
- PC에서 메모리라 부르는 것이 DRAM에 해당하고 DRAM에 데이터가 올라가는 것 자체가 변수에 해당한다.
- 다시 말해 변수란 특정한 데이터를 가지고 있고 정보를 저장할 수 있는 메모리 공간이다.

16진수 체계

EDDI
Electronic Design Development Institute

- 16진 수는 0에서 9까지의 숫자 10개와 영문자 A, B, C, D, E, F 등의 문자 6개, 즉 16개의 숫자와 문자로 구성된다.
- 16진수를 나타내기 위해서는 16개의 숫자가 필요하지만 10진수의 숫자는 10개밖에 없으므로 6개의 영문 자를 추가한 것이다.

컴퓨터 분야에서 16진수를 사용하는 이유?

- 2진수로는 숫자의 크기 등을 판단하기 힘들고 관리하는데 어려움이 따르기 때문이다.
- 2진수 4자리는 16진수 한자리에 해당하기 때문에 4 대 1로 변환이 이루어져 크기판단과 관리가 용이하다.
- 16진수 자릿수 2개는 8비트에 해당하고, 8비트는 곧 1바이트이다.

xor를 이용한 대소문자 변환 방식

EDDI
Electronic Design Development Institute

- 아스키 코드는 1963년 미국 ANSI에서 표준화한 정보교환용 7비트 부호체계이다.
- 대문자 A~Z는 아스키 코드에서 65~90(0x41~0x5A)에 해당한다.
- 소문자 a~z는 대문자 A~Z의 아스키 코드에서 32를 더한 97~122(0x61~0x7A)에 해당한다.
- 즉, 대문자에서 32를 빼면 소문자가, 소문자에서 32를 더하면 대문자가 된다.
- 어떤 알파벳 문자의 아스키 코드를 2진수로 표현해 보면 소문자라면 32에 해당하는 비트가 1이고, 대문자라면 32에 해당하는 비트가 1이 아니라는 차이만 존재한다.
  ex) A -> 0x1000001, a -> 0x1100001
- 따라서 32에 해당하는 비트만 xor연산을 해준다면 대문자는 소문자로, 소문자는 대문자로 변환할 수 있게 된다.

xor를 이용한 대소문자 변환 방식

EDDI
Electronic Design Development Institute

```c
#include <stdio.h>

int main(void)
{
        char c = '\0';

        printf("문자를 입력하세요: ");
        scanf("%c", &c);

        if ((c > 96 && c < 123) || (c > 64 && c < 91))
        {
                printf("xor는 뭐든 가능하다!\n");
                printf("입력값 = %c, 변환값 = %c\n", c, c ^ 0x20);
        }

        return 0;
}
```

결과 대문자를 입력하면 소문자, 소문자를 입력하면 대문자가 출력된다.

Memory 구조(Stack, Heap, Data, Text)

EDDI
Electronic Design Development Institute

스택

- 함수의 호출과 관계되는 지역 변수와 매개변수가 저장되는 영역이다.
- 함수의 호출과 함께 할당되며, 함수의 호출이 완료되면 소멸한다.
- 메모리의 높은 주소에서 낮은 주소의 방향으로 할당된다.

힙

- 메모리의 힙(heap) 영역은 사용자가 직접 관리할 수 있는 메모리 영역이다.
- 사용자에 의해 메모리 공간이 동적으로 할당되고 해제된다.
- 메모리의 낮은 주소에서 높은 주소의 방향으로 할당된다.

Memory 구조(Stack, Heap, Data, Text)

EDDI
Electronic Design Development Institute

데이터

- 프로그램의 전역 변수와 정적(static) 변수가 저장되는 영역이다.
- 프로그램의 시작과 함께 할당되며, 프로그램이 종료되면 소멸한다.

텍스트

- 실행할 프로그램의 코드가 저장되는 영역으로 코드(code) 영역이라고도 부른다.
- CPU는 텍스트 영역에 저장된 명령어를 하나씩 가져가서 처리하게 된다.

gdb를 사용하여 디버깅 하기 위해 필요한 컴파일 옵션

EDDI
Electronic Design Development Institute

- -g 옵션을 추가해서 컴파일 한다.

  ex) gcc [파일명].c -g

변수의 주소값과 실제 메모리의 값이 일치하는지 확인(디버깅)

EDDI
Electronic Design Development Institute

```
(gdb) p/x &var
$1 = 0x7fffffffde84
(gdb) n
var = 7
6               printf("var mem addr = 0x%x\n", &var);
(gdb) n
var mem addr = 0xffffde84
8               return 0;
(gdb)
```

두 개의 변수를 입력받아 대소 비교

EDDI
Electronic Design Development Institute

```c
#include <stdio.h>

int main(void)
{
        int x, y;

        printf("두 변수의 대소 비교\n");
        printf("x를 입력하세요.\n");
        scanf("%d", &x);
        printf("y를 입력하세요.\n");
        scanf("%d", &y);

        if(x > y)
                printf("x가 더 큽니다.\n");
        else if(x < y)
                printf("y가 더 큽니다.\n");
        else
                printf("x와 y는 같습니다.\n");

        return 0;
}
```

```
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
두 변수의 대소 비교
x를 입력하세요.
32
y를 입력하세요.
64
y가 더 큽니다.
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ vi compare.c
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
두 변수의 대소 비교
x를 입력하세요.
32
y를 입력하세요.
32
x와 y는 같습니다.
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
두 변수의 대소 비교
x를 입력하세요.
64
y를 입력하세요.
32
x가 더 큽니다.
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$
```

피보나치 수열을 표현할 수 있도록 프로그래밍

EDDI
Electronic Design Development Institute

```c
#include <stdio.h>

int main(void)
{
        int i = 0, j, a = 0, b = 1, tmp;

        printf("피보나치 수열 개수 지정 : ");
        scanf("%d", &j);

        for(i; i<j; i++)
        {
                printf("%d ", b);
                tmp = b;
                b += a;
                a = tmp;
        }

        printf("\n");

        return 0;
}
```

```
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ gcc Fibonacci.c
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
피보나치 수열 개수 지정 : 5
1 1 2 3 5
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
피보나치 수열 개수 지정 : 10
1 1 2 3 5 8 13 21 34 55
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$
```

주사위 게임

EDDI
Electronic Design Development Institute

```c
int main(void)
{
        int com_dice, user_dice;

        srand(time(NULL));

        com_dice = rand() % 6 + 1;
        user_dice = rand() % 6 + 1;

        printf("컴퓨터의 주사위 결과 : %d\n", com_dice);
        printf("사용자의 주사위 결과 : %d\n", user_dice);

        if(com_dice > user_dice)
                printf("컴퓨터 승\n");
        else if(com_dice < user_dice)
                printf("사용자 승\n");
        else
                printf("비겼습니다.\n");

        return 0;
}
```

```
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
컴퓨터의 주사위 결과 : 6
사용자의 주사위 결과 : 5
컴퓨터 승
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
컴퓨터의 주사위 결과 : 6
사용자의 주사위 결과 : 6
비겼습니다.
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
컴퓨터의 주사위 결과 : 1
사용자의 주사위 결과 : 6
사용자 승
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$ ./a.out
컴퓨터의 주사위 결과 : 5
사용자의 주사위 결과 : 2
컴퓨터 승
dama@dama-15ZD990-VX5BK:~/EmbeddedMasterLv1/3기/WSK/homework$
```