



C언 - HW2

임베디드스쿨1기

Lv1과정

2021. 03. 26

이충재

변수란?

특정 데이터타입을 저장 할 수 있는 메모리 공간

* 포인터의 크기는 ALU에 의하여 결정된다.
ALU에 있는 범용 레지스터에 의하여 연산 할 수 있는 최대치 결정되기 때문

* 가상 메모리 크기는 포인터의 크기를 따라간다.
범용 레지스터의 크기는 64비트 = 8바이트
레지스터의 크기 64비트를 모두 활용하지 않는것은 낭비이다.
레지스터의 비트를 낭비하지 않기 위해 가상메모리를 64비트로 표현

* 물리 메모리의 최소단위는 페이지 프레임(4kB)

메모리를 마구잡이로 할당하면 문제생길수 있다.

예를들어 80바이트 크기의 메모리를 10000번 할당한다고 하면,

메모리 손해 = $(4096 - 80) * 10000 = 40\text{MB}$

실제 사용량 = $80 * 10000 = 800\text{kB}$

실제 메모리 사용량보다 버려지는것이 더 많다.

이러한 문제를 해결하기 위하여 슬랩 할당자와 버디 할당자가 생김

슬랩 할당자: 작은단위 메모리 할당시 메모리를 쪼개서 나눠준다.

버디 할당자: 큰 메모리를 할당하는 경우 여러 페이지를 제공한다.

0 ~ 100까지 숫자중 홀수만 출력하시오.

```
int main(void)
{
    int i;

    for(i = 0; i <= 100; i++)
    {
        if(i % 2)
        {
            printf("%d", i);
        }
        else if((i % 20)==0)
        {
            printf("\n");
        }
        else
        {
            printf(" ");
        }
    }

    return 0;
}
```

먼저 i가 홀수일때 i값을 출력하고

한줄에 10개의 수를 나타내기 위하여
20의 배수마다 줄을 바꿔주었다.

짝수일때 띄어쓰기를 하여 한칸씩
공백이 생기도록 하였다.

```
1 3 5 7 9 11 13 15 17 19
21 23 25 27 29 31 33 35 37 39
41 43 45 47 49 51 53 55 57 59
61 63 65 67 69 71 73 75 77 79
81 83 85 87 89 91 93 95 97 99
```

1 ~ 50까지 숫자의 합을 구하는 프로그램을 작성하세요.

```
#include <stdio.h>

int main(void)
{
    int i;
    int sum = 0;

    for(i = 1; i <= 50; i++)
    {
        sum += i;
    }

    printf("%d\n", sum);

    return 0;
}
```

i = 50까지 sum에 i를 더하여
1에서 50까지 합을 구하였다.

i = 1, sum = 1

i = 2, sum = 1 + 2

i = 3, sum = 1 + 2 + 3

(중간생략)

i = 50, sum = 1 + 2 + 3 + ... + 50

결과값으로 1275를 얻었다.

1 ~ 33까지의 숫자중 3의 배수의 합만 구해보세요.

```
#include <stdio.h>

int main(void)
{
    int i;
    int sum = 0;

    for(i = 1; i <= 33; i++)
    {
        if((i % 3) == 0)
        {
            sum += i;
        }
    }

    printf("%d\n ", sum);

    return 0;
}
```

for문 안에서 if문을 사용하여
1 ~ 33 범위 내에서 3의배수의 합을 구하였다.

결과값으로 198이 나왔다.

1 ~ 100의 숫자중 2의 배수의 합과 3의 배수의 합을 각각 구해봅시다.

```
#include <stdio.h>

int main(void)
{
    int i;
    int sum1 = 0;
    int sum2 = 0;

    for(i = 1; i <=100; i++)
    {
        if((i%2) == 0 )
            sum1 += i;

        if((i%3) == 0 )
            sum2 += i;
    }

    printf("2의배수의 합은 %d, 3의배수의 합은 %d 입니다.\n", sum1, sum2);

    return 0;
}
```

sum1에는 2의배수의 합, sum2에는 3의배수의 합을 저장하였다.

결과

2의배수의 합은 2550, 3의배수의 합은 1683 입니다.

피보나치 수열의 n 번째 항을 구하는 프로그램을 만들어봅시다.

```
#include <stdio.h>

int main(void)
{
    int a, b, c, i, n;

    b = 1;
    c = 1;

    printf("몇번째 항? ");
    scanf("%d", &n);

    if((n == 1) || (n == 2))
    {
        a = 1;
    }

    if(n >= 3)
    {
        for(i = 3; i <= n; i++)
        {
            a = b + c;
            b = c;
            c = a;
        }
    }

    printf("n번째 항은 %d 입니다.\n", a);

    return 0;
}
```

결과를 나타내는 변수 a,
앞의 두 수를 나타내는 변수 b, c

if문을 사용하여 첫번째 항과
두번째 항을 1로 두었고 세번째 항부터
앞의 두수의 합으로 두었다.

반복횟수가 늘어날때마다
c에 있는수를 b로, a에 있는수를 c로 옮겼다

```
lee@lee-15ND540-UX5SK:~/바탕화면$ ./a.out
몇번째 항? 2
n번째 항은 1 입니다.
lee@lee-15ND540-UX5SK:~/바탕화면$ ./a.out
몇번째 항? 8
n번째 항은 21 입니다.
```


1, 1, 1, 1, 2, 3, 4, 5, 7, 10, 14, 19, 26, 36, 50, ... 으로 진행되는 수열이 있다.
여기서 25번째 항의 숫자값을 구해보시다.

```
#include <stdio.h>

int main(void)
{
    int i, n, a, b, c, d, e;

    a = 1;
    b = 1;
    c = 1;
    d = 1;
    e = 1;

    printf("몇번째 항? ");
    scanf("%d", &n);

    if (n <= 0)
    {
        printf("잘못된 값입니다.\n");
    }

    if((n > 0) && (n < 5))
    {
        printf("n번째 항은 %d 입니다.\n", a);
    }
}
```

```
if (n >= 5)
{
    for(i = 5; i <= n; i++)
    {
        b = c;
        c = d;
        d = e;
        e = a;
        a = b + e;
    }

    printf("n번째 항은 %d 입니다.\n", a);
}

return 0;
```

```
몇번째 항? 0
잘못된 값입니다.
lee@lee-15ND540-UX5SK:~/바탕화면$ ./a.out
몇번째 항? 3
n번째 항은 1 입니다.
lee@lee-15ND540-UX5SK:~/바탕화면$ ./a.out
몇번째 항? 7
n번째 항은 4 입니다.
```

앞의 4수를 변수 b, c, d, e 로 두고 반복횟수가 늘어날때마다
 $b \leftarrow c, c \leftarrow d, d \leftarrow e, e \leftarrow a$ 로 수를 옮겼다. a는 결과값을 나타내는 변수이다.

질문: 1. ALU 범용 레지스터가 연산의 최대치를 결정하기 때문에 포인터의 크기를 결정한다.
이 말이 잘 이해가 되지 않습니다.

포인터는 메모리 주소를 저장하는 변수로 알고있는데 레지스터와 어떤 관계를 가지기에
레지스터가 포인터의 크기를 결정하는지 잘 모르겠습니다.

질문2. 레지스터의 비트를 낭비하지 않기 위해 가상메모리를 64비트로 표현한다.
이 말도 잘 이해가 가지 않습니다.

레지스터의 비트를 낭비하지 않아야겠다는 것은 알겠으나
레지스터와 가상메모리의 관계를 잘 모르겠습니다.