

# 사물인터넷(IoT/ICT)환경의 임베디드 개발자 양성과정

과제 4 일차 이우석

## [배운 내용 복습 (goto, 파이프라인, for)]

\* goto 는 주로 kernel 에서 많이 사용하며, Buffer 를 사용하는 곳에서 주로 사용함( 가독성, 편리성 좋음)

\* 파이프라인 : CPU 와 같은 디지털 집적회로의 속도를 향상시키기 위한 회로

ex) 4 단 파이프라인 구조

1 단계	Fetch (명령어를 메모리에서 인출)
2 단계	Decode (명령어 해석)
3 단계	Execute (명령어 실행)
4 단계	Writeback (라이트백)

\* for 문 = 특정상황에서 while 문이나 do~while 문보다 편하게 반복문을 구성할 수 있도록 설계된 것.

ex) for( 초기식 ; 조건식 ; 증감식)  
{  
    // 반복의 대상이 되는 문장들  
}

초기식 : 반복을 위한 변수의 선언 및 초기화에 사용

조건식 : 반복의 조건을 검사하는 목적으로 선언됨

증감식 : 반복의 조건을 ‘거짓’ 으로 만드는 증가 및 감소 연

## 2. [문제 은행 (어제 내용을 for 문으로 작성)]

### 1 번문제 과정

```
#include <stdio.h>

int q1()
{
    int fee = 37500;
    int day=0;
    int result=0;

    for(day=1;day<8;day++){
        if(day>=3)
        {
            result =(day*fee)*0.8;
        }
        else
        {
            result=day*fee;
        }
        printf("%d일 대여 요금은 %d입니다\n", day, result);
    }

    return 0;
}

int main(void)
{
    q1();

    return 0;
}
```

### 1 번문제 결과

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
1일 대여 요금은 37500입니다
2일 대여 요금은 75000입니다
3일 대여 요금은 90000입니다
4일 대여 요금은 120000입니다
5일 대여 요금은 150000입니다
6일 대여 요금은 180000입니다
7일 대여 요금은 210000입니다
```

### 3번문제 과정

```
#include <stdio.h>

int q3()
{
    int num;

    int result=0;

    for(num=1;num<=1000;num++)
    {
        if(num%3==0)
        {
            result=result+num;
        }
    }
    return result;
}

int main(void)
{
    printf("1 ~ 1000 사이의 3의 배수의 합 : %d\n", q3());

    return 0;
}
```

### 3번문제 결과

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
1 ~ 1000 사이의 3의 배수의 합 : 166833
```

## 4번문제 과정

```
#include <stdio.h>
int q4()
{
    int num;
    int result=0;
    for(num=1;num<=1000;num++)
    {
        if((num%4==1)&&(num%6==1))
        {
            result=result+num;
        }
    }
    return result;
}

int main(void)
{
    printf("1~1000 사이에 4나 6으로 나뉘도 \n나머지가 1인 수의 합 : %d\n", q4());
    return 0;
}
```

## 4번문제 결과값

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
1~1000 사이에 4나 6으로 나뉘도
나머지가 1인 수의 합 : 41916
```

## 5번문제 과정

```
#include <stdio.h>

int q5(int input)
{
    int num;
    int result=0;

    for(num=1;num<input+1;num++){

        result=7*num;
        printf("%d 항의 값 : %d\n",num,result);
    }

    return 0;
}

int main(void)
{

    int input=0;
    printf("항의 개수를 입력하시오 : ");
    scanf("%d",&input);
    q5(input);

    return 0;
}
```

## 5번문제 결과값

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
항의 개수를 입력하시오 : 7
1 항의 값 : 7
2 항의 값 : 14
3 항의 값 : 21
4 항의 값 : 28
5 항의 값 : 35
6 항의 값 : 42
7 항의 값 : 49
```

## 10 번문제 과정

```
#include <stdio.h>
int main(void)
{
    int i;
    int h;
    int result=0;

    for(i=2;i<10;i++){
        printf("%d 단\n",i);

        for(h=1;h<10;h++){
            result=i*h;
            printf("%d * %d = %d\n",i,h,result);
        }

        return 0;
    }
}
```

## 10 번문제 결과값

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
2 단
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
3 단
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
4 단
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
5 단
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
```

### 3. [fib 함수 동작 분석(디버깅 및 그림그리기)]

(1)

```
Reading symbols from debug...done.
(gdb) b main
Breakpoint 1 at 0x400642: file fib.c, line 12.
(gdb) r
Starting program: /home/wooseok91/my_proj/Homework/sanghoonlee/homework1/debug

Breakpoint 1, main () at fib.c:12
12      {
(gdb) l
7          else
8              return fib(num-1) + fib(num-2);
9      }
10
11      int main(void)
12      {
13          int result, final_val;
14          printf("피보나치 수열의 항의 개수를 입력하시오 : ");
15          scanf("%d", &final_val);
16          result=fib(final_val);
(gdb) s
14      printf("피보나치 수열의 항의 개수를 입력하시오 : ");
(gdb) n
15      scanf("%d", &final_val);
(gdb) n
피보나치 수열의 항의 개수를 입력하시오 : 5
16      result=fib(final_val);
(gdb) s
fib (num=5) at fib.c:5
5          if(num == 1 || num == 2)
(gdb) bt
#0  fib (num=5) at fib.c:5
#1  0x0000000000400680 in main () at fib.c:16
(gdb) s
8          return fib(num-1) + fib(num-2);
(gdb) s
fib (num=4) at fib.c:5
5          if(num == 1 || num == 2)
(gdb) s
8          return fib(num-1) + fib(num-2);
(gdb) s
fib (num=3) at fib.c:5
5          if(num == 1 || num == 2)
(gdb) s
8          return fib(num-1) + fib(num-2);
(gdb) s
fib (num=2) at fib.c:5
```

(2)

```
(gdb) bt
#0  fib (num=2) at fib.c:5
#1  0x0000000000400622 in fib (num=3) at fib.c:8
#2  0x0000000000400622 in fib (num=4) at fib.c:8
#3  0x0000000000400622 in fib (num=5) at fib.c:8
#4  0x0000000000400680 in main () at fib.c:16
(gdb) finish
Run till exit from #0  fib (num=2) at fib.c:5
0x0000000000400622 in fib (num=3) at fib.c:8
8          return fib(num-1) + fib(num-2);
Value returned is $1 = 1
```

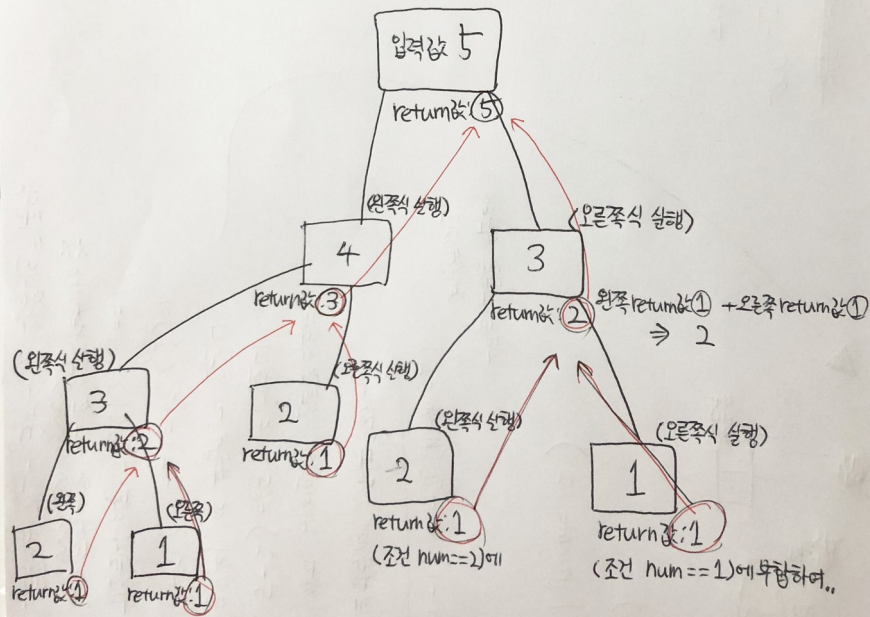
항의 입력값을 5로 주고, 계산은 **왼쪽식**부터 계산이 되며 조건에 부합할때 까지 되었을때 return 값이 **1**이 나오는걸 확인할 수 있다. 이대로 반복하면 오른쪽 식으로도 계산이 될것이다.



(3)

```
int fib(int num)
{
    if (num == 1 || num == 2)
        return 1;
    else
        return fib(num-1) + fib(num-2);
}
```

예) 피보나치 수열의 항의 개수를 '5'로 입력 (식은 오른쪽부터 계산예정)



이 그림은 **입력값**을 **5**로 주었으며,  
여기서는 식을 **오른쪽** 서부터 **실행**하였다.  
조건이 맞을때까지 **반복**이 되었으며  
조건이 성립 되었을 경우, **return 값 1**을  
받는것을 알수있다. 두 조건이 둘다 맞을경우  
return 값을 더하면 입력값의 return 값을  
알 수 있다.