

Embedded Programmer 과정 4일차 Homework

김형주

Goto 문

- Jmp명령어로 이루어진 구문이다.
- Goto LABEL;
LABEL:
- 과 같은 형태로 사용된다. 주로 if문 안에 넣어, 이중/삼중 루프를 탈출할 때(에러처리 등) 사용할 수 있다.
- Jmp or call 등의 분기 명령어는 기본적으로 파이프라인을 파괴하기 때문에, 적게 사용할 수록 좋다.
- 이중/다중 루프를 탈출하는 데, break문을 사용하게 되면, 루프 수만큼 jmp명령어가 필요하므로 goto문을 사용하는 것이 효율적이고 좋음.

파이프라인

- Cpu는 1번에 1개의 동작만을 실행한다. 반면, HW회로는 병렬회로를 구성할 수 있듯, 모든 회로가 동시에 동작할 수 있다. 파이프라인이란 cpu에 구성된 회로이기 때문에, 동시에 모든 모듈을 실행할 수 있다.
- 가장 기본적인, 3단계의 파이프라인은 fetch, discode, execute로 이루어져 있다.

For문 3번문제

(1~1000, 3의 배수의 합)

```
howard@ubuntu: ~/Mytest
#include <stdio.h>

void searchNumFunc(int start, int end, int val);

int main(void){
    int start = 1, end = 1000, val = 3;

    searchNumFunc(start,end,val);

    return 0;
}

void searchNumFunc(int start, int end, int val){
    int i, sum = 0;
    for(i=start;i<end;i++){
        if(!(i%3))                //i is multiples of 3
            sum+=i;
    }
    printf("sum = %d\n",sum);
}
```

For문 4번문제 (4or6나뉘었을 때 나머지1인 수 합)

```
howard@ubuntu: ~/Mytest
#include <stdio.h>

void searchNumFunc(int start, int end, int val1, int val2);

int main(void){
    int start = 1, end = 1000, val1 = 4, val2 = 6;

    searchNumFunc(start,end,val1,val2);

    return 0;
}

void searchNumFunc(int start, int end, int val1, int val2){
    int i, sum = 0;
    for(i=start;i<end;i++){
        if(!(i%4-1)||!(i%6-1))           //i is
multiples of 4 or 6
        sum+=i;
    }
    printf("sum = %d\n",sum);
}
~
~
```

For문 10번문제 (구구단)

```
howard@ubuntu: ~/Mytest
#include <stdio.h>
void multipleTables(int start, int end);
int main(void){
    int start = 2, end = 9;

    multipleTables(start, end);

    return 0;
}

void multipleTables(int start, int end){
    int i, j;
    for(i=start; i<=end; i++){
        for(j=1; j<=9; j++){
            printf("%d*%d=%d\n", i, j, i*j);
        }
    }
}
```

재귀함수 디버깅(피보나치수열)

