TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 최대성
c3d4s19@naver.com

2018.05.16. – 055 일차 수업

<vector_op.h 헤더파일>

```c
#ifndef __VECTOR_OP_H__
#define __VECTOR_OP_H__
#include <stdio.h>
#include <math.h>
typedef struct _vector{
    double x;
    double y;
    double z;
}vect;
typedef struct _vector_op{
    void (*print)(vect);
    void (*magnitude)(vect, double*);
    void (*add)(vect, vect, vect*);
    void (*sub)(vect, vect, vect*);
    void (*scale)(double, vect, vect*);
    void (*dot)(vect, vect, double*);
    void (*cross)(vect, vect, vect*);
    void (*proj)(vect, vect, vect*);
    void (*Gram_Schmidt_Orth)(vect*, vect*, vect*);
}vect_op;


void print_vect(vect v);
void magnitude_vect(vect v, double* res);
void add_vect(vect a, vect b, vect* res);
void sub_vect(vect a, vect b, vect* res);
void scale_vect(double a, vect b, vect* res);
void dot_vect(vect a, vect b, double* res);
void cross_vect(vect a, vect b, vect* res);
void proj_vect(vect a, vect b, vect* res);
void Gram_Schmidt_Orth(vect* a, vect* b, vect* c);
#endif
```

<vector_op.c 파일>

```c
#include "vector_op.h"
void print_vect(vect v){
        printf("%lf   %lf   %lf\n", v.x, v.y, v.z);
}


void magnitude_vect(vect v, double* res){
    *res = sqrt(v.x * v.x + v.y * v.y + v.z * v.z);
}


void add_vect(vect a, vect b, vect* res){
    res->x = a.x + b.x;
    res->y = a.y + b.y;
    res->z = a.z + b.z;
}


void sub_vect(vect a, vect b, vect* res){
    res->x = a.x - b.x;
    res->y = a.y - b.y;
    res->z = a.z - b.z;
}


void scale_vect(double a, vect b, vect* res){
    res->x = a * b.x;
    res->y = a * b.y;
    res->z = a * b.z;
}


void dot_vect(vect a, vect b, double* res){
    *res = a.x * b.x + a.y * b.y + a.z * b.z;
}
```

```
void cross_vect(vect a, vect b, vect* res){
    res->x = a.y * b.z - a.z * b.y;
    res->y = a.z * b.x - a.x * b.z;
    res->z = a.x * b.y - a.y * b.x;
}

void proj_vect(vect a, vect b, vect* res){
    double tmp;
    double a_dot_b;
    double magnitude_a;

    dot_vect(a, b, &a_dot_b);
    magnitude_vect(a, &magnitude_a);
    tmp = a_dot_b / (magnitude_a * magnitude_a);
    scale_vect(tmp, a, res);
}

void Gram_Schmidt_Orth(vect* a, vect* b, vect* c){
    vect tmp1;
    vect tmp2;
    // a 벡터 기준으로 b 벡터 직교화
    proj_vect(*a, *b, &tmp1);
    sub_vect(*b, tmp1, b);

    // a 와 b 벡터 기준으로 c 벡터 직교화
    proj_vect(*a, *c, &tmp1);
    proj_vect(*b, *c, &tmp2);
    sub_vect(*c, tmp1, c);
    sub_vect(*c, tmp2, c);
    // a, b, c 벡터값 변경됨
}
```

<vector_main.c 함수>

```
#include "vector_op.h"
vect_op op = {
    print_vect,
    magnitude_vect,
    add_vect,
    sub_vect,
    scale_vect,
    dot_vect,
    cross_vect,
    proj_vect,
    Gram_Schmidt_Orth
};

int main(){
    vect a = {0, 4, 0};
    vect b = {2, 2, 1};
    vect c = {1, 1, 1};

    vect res_vect;
    printf("--------------------------------\n");
    op.sub(a, b, &res_vect);
    op.print(res_vect);
    printf("--------------------------------\n");
    op.cross(a, b, &res_vect);
    op.print(res_vect);
    printf("--------------------------------\n");
    op.Gram_Schmidt_Orth(&a, &b, &c);
    op.print(a);
    op.print(b);
    op.print(c);
    printf("--------------------------------\n");
}
```