

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

2018-02-27 (5회차)

정유경

5회차 Review

- char형 배열이 필요한 이유 ?

string인 문자열 "I'm Marth Kim"은 문자열 상수로 변경이 불가능
char형 배열은 문자열 변수이므로 내부 데이터 변경이 가능하다

- Null Character는 무엇인가 ?

NULL 문자는 '\0'으로 어느 부분이 배열의 마지막인지를 명시

- 고차원배열의 초기화

```
arr[2][2][3] = {  
    {{1, 2, 3}, {1, 2, 3}},  
    {{1, 2, 3}, {1, 2, 3}}  
};
```

- 배열의 이름이 배열의 주소이다

- 함수의 인자로 배열을 전달하려면 ?

함수 호출시 배열의 주소값을 전달함

- Pointer와 배열은 무슨 관계 ?

배열은 주소값이 정해져있음 포인터는 주소값을 저장할 수 있음 즉, 포인터를 이용해 배열의 요소에 접근 가능

- 배열은 메모리 상에 순차적으로 배열이 되어 있다

- 배열이 필요한 이유는 ?

77 GHz 차량용 레이더, 라이다에서 수십 GB 로 쌓이는 센서 정보

- `int (*pArr)[3];`

배열을 가리키는 포인터 즉 배열포인터

Int형 변수를 요소로 지니고 포인터 연산시 4칸씩 이동하는 2중 배열을 가리키는 포인터이다.

- `int* pArr[3];`

Int형 변수의 주소값 4개를 저장할 수 있는 배열!

즉 포인터 배열

- 배열의 이름이 가리키는 요소가 배열이름의 포인터 타입을 결정한다.

`int* arr1[10]` 에서 `arr1`이 가리키는 배열의 첫번째 요소가 int형 포인터 이므로 `arr1`의 포인터 타입은 int형 더블포인터(`int**`)

Segmentation Fault

접근하면 안되는 메모리 영역에 접근하였기에 엄밀하게는 Page Fault(물리 메모리 할당되지 않음)가 발생하게 되고 원래는 Interrupt가 발생해서

Kernel이 Page Handler(페이지 제어기)가 동작해서 가상 메모리에 대한 Paging 처리를 해주고 실제 물리 메모리를 할당해주는데 문제는 이것이 User 쪽에서 들어온 요청이므로 Kernel 쪽에서 강제로 기각해버리면서 Segmentation Fault가 발생하는 것이다.

실제 Kernel 쪽에서 들어온 요청일 경우에는

위의 메커니즘에 따라서 물리 메모리를 할당해주게 된다.

배열 연습문제

1. 배열에 문자열을 입력 받고,
각 배열 요소가 짝수인 경우만을 출력하는 함수를 작성하라.
3. 아래와 같은 숫자들이 배열에 들어 있다고 가정한다.
3, 77, 10, 7, 4, 9, 1, 8, 21, 33
이 요소들을 배열에 거꾸로 집어넣어보자.
4. 위의 숫자 3, 77, 10, 7, 4, 9, 1, 8, 21, 33에서
홀수 번째 요소의 합과 짝수 번째 요소의 합을 곱하십시오.
6. 행렬의 곱셈, 덧셈, 나눗셈, 뺄셈에 대해 조사하십시오.
숫자를 예로 들어서 계산도 해보십시오.

배열 연습문제 1번

배열에 문자열을 입력 받고, 각 배열 요소가 짝수인 경우만을 출력하는 함수를 작성하라.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int i = 0;
6      char str[20] = "Do not Give Up";
7
8      for (i = 0; i < 20; i++)
9      {
10         if (!(str[i] % 2 == 0))
11         { //각 배열요소가 짝수일 경우에만 출력!
12             printf(" %c ", str[i]);
13         }
14     }
15     printf("\n");
16     return 0;
17 }
```

배열 연습문제 2번

아래와 같은 숫자들이 배열에 들어 있다고 가정한다. 3, 77, 10, 7, 4, 9, 1, 8, 21, 33
이 요소들을 배열에 거꾸로 집어넣어보자.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int Arr[] = { 3,77,10,7,4,9,1,8,21,33 };
6      int Reverse[10] = { 0, };
7
8      int i = 0;
9      int size = sizeof(Arr)/sizeof(int);
10     //printf("%d",size);
11
12     for (i = 0; i < size; i++)
13     {
14         Reverse[size-i-1]=Arr[i];
15     }
16
17     for (i = 0; i < size; i++)
18     {
19         printf("%3d",Reverse[i]);
20     }
21     printf("\n");
22     return 0;
23 }
```


배열 연습문제 3번

위의 숫자 3, 77, 10, 7, 4, 9, 1, 8, 21, 33에서 홀수 번째 요소의 합과 짝수 번째 요소의 합을 곱하시오.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5
6      int Arr[] = { 3,77,10,7,4,9,1,8,21,33 };
7
8      int i = 0;
9      int size = sizeof(Arr)/sizeof(int);
10     int evenSum=0, oddSum = 0;
11
12     for (i = 0; i < size; i++)
13     {
14         if (!(i % 2 == 0))//짝수
15             evenSum += Arr[i];
16         else
17             oddSum += Arr[i];
18     }
19     printf("짝수합: %d\n",evenSum);
20     printf("홀수합: %d\n", oddSum);
21     return 0;
22 }
```

배열 연습문제 4번

행렬의 곱셈, 덧셈, 나눗셈, 뺄셈에 대해 조사하시오. 숫자를 예로 들어서 계산도 해보시오.

아래와 같이 두 행렬이 있다고 가정하자.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

1. 행렬의 덧셈

$$A + B = \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{pmatrix}$$

2. 행렬의 뺄셈

$$A - B = \begin{pmatrix} a_{11}-b_{11} & a_{12}-b_{12} \\ a_{21}-b_{21} & a_{22}-b_{22} \end{pmatrix}$$

3. 행렬의 곱셈

$$A \times B = \begin{pmatrix} a_{11}*b_{11}+a_{12}*b_{21} & a_{11}*b_{12}+a_{12}*b_{22} \\ a_{21}*b_{11}+a_{22}*b_{21} & a_{21}*b_{12}+a_{22}*b_{22} \end{pmatrix}$$

3. 역행렬

아래와 같이 정의된 행렬A가 있다고 하자.

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

아래조건을 만족하는 행렬B가 있다고 하면

$$A \times B = E$$

행렬B는 행렬A의 역행렬이다.

즉, $B = A^{-1}$ 이다.

행렬A의 역행렬은 다음과 같이 정의된다.

$$A^{-1} = \frac{1}{\|A\|} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

역행렬이 존재하기 위한 조건이 있다.

$\|A\| = ad - bc \neq 0$ 을 만족해야 한다.

포인터 연습문제

1. 정수 2004016을 변수에 저장하고 이것을 char형 포인터로 받는다.
그리고 정수형은 총 4byte로 구성되므로 총 4개의 byte를 볼 수 있을것이다.
각 byte에 숫자가 어떻게 배치되었는지 확인해보자.
2. 우리는 예제에서 주소값을 교환하여 값을 변경하는 것을 해보았다.
그렇다면 변수 3개를 놓고, 이것에 대해서 무한 Loop를 돌면서 저글링을 해보자!

포인터 연습문제 1번

정수 2004016을 변수에 저장하고 이것을 char형 포인터로 받는다.
그리고 정수형은 총 4byte로 구성되므로 총 4개의 byte를 볼 수 있을것이다.
각 byte에 숫자가 어떻게 배치되었는지 확인해보자.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int num = 2004016;
6      char* p = &num;
7      int i = 0;
8      /*int형 데이터의 주소를 char 형(1바이트) 포인터로 받았다.
9       즉, sizeof(int)=4바이트를 4토막으로 나누어 sizeof(char)=바이트 단위로 접근한다.
10      따라서 num이라는 변수가 잡혀있는 주소를 한바이트씩 참조해보면 다음과 같다
11      */
12
13     for (i = 0; i < 4; i++)
14     {
15         printf("%p\n",p[i]);
16     }
17
18     return 0;
19 }
```

포인터 연습문제 2번

우리는 예제에서 주소값을 교환하여 값을 변경하는 것을 해보았다.
그렇다면 변수 3개를 놓고, 이것에 대해서 무한 Loop를 돌면서 저글링을 해보자!

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int num1 = 3, num2 = 7, num3 = 5;
6      int temp = NULL;
7      int *num1_p = &num1;
8      int *num2_p = &num2;
9      int *num3_p = &num3;
10     int **num_p_p = &num1_p;
11
12     printf("*num1_p=%d\n", *num1_p);
13     printf("*num2_p=%d\n", *num2_p);
14     printf("*num3_p=%d\n", *num3_p);
15     while (1)
16     {
17         temp = *num_p_p;
18         *num_p_p = num2_p;
19         num2_p = num3_p;
20         num3_p = temp;
21
22         printf("*num1_p=%d\n", *num1_p);
23         printf("*num2_p=%d\n", *num2_p);
24         printf("*num3_p=%d\n", *num3_p);
25     }
26     return 0;
27 }
```

추가문제

- 삼각형의 넓이 구하는 문제

case 1) 밑변, 높이

case 2) 밑변, 밑변과 다른 변이 이루는 각도

- 2 가지 경우로 모두 구현해본다.
(삼각함수 복습합시다 ^^)

추가문제 1번

삼각형의 넓이 구하는 문제

case 1) 밑변, 높이

```
roject (선택 범위) main(void)
1  #include <stdio.h>
2
3  int main(void)
4  {
5
6      int h = 3, len = 5;
7
8      printf("삼각형의 넓이: %f\n", h*len*0.5);
9
10     return 0;
11 }
```

추가문제 2번

삼각형의 넓이 구하는 문제

case 2) 밑변, 밑변과 다른 변이 이루는 각도

Quiz

1. 총 7개의 통장을 만들어서 100만원 단위로 최대 500만원까지 입금하였다. 이자율이 연 4%라고 할 때, 3년 후 각각의 총액을 구하시오.
2. 2 by 2 행렬의 곱셈을 계산할 수 있는 프로그램을 만드시오.

Quiz 1번

총 7개의 통장을 만들어서 100만원 단위로 최대 500만원까지 입금하였다. 이자율이 연 4%라고 할 때, 3년 후 각각의 총액을 구하시오.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int i, j = 0;
6      int size = 0;
7      int year = 3;
8      int length = 0;
9
10     double Arr1[7] = { 1000000, 2000000, 3000000, 4000000, 5000000, 4000000, 3000000 };
11     int Arr2[7] = { 0, };
12
13     /*배열요소의 개수를 조절할 수 있도록 length변수를 사용한다*/
14     length = sizeof(Arr1) / sizeof(double);
15
16     for (i = 0; i < length; i++)
17     {
18         for (j = 0; j < year; j++)
19         {
20             Arr2[i]=Arr1[i] * 1.04;
21         }
22         printf("3년 후 통장 %d의 총액: %7d원\n",i+1,Arr2[i]);
23     }
24     return 0;
25 }
```

Quiz 2번

2 by 2 행렬의 곱셈을 계산할 수 있는 프로그램을 만드시오.

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      // 2x2인 int형 이중배열선언 및 초기화
6      int Arr1[2][2] = {
7          {1,2},
8          {3,4}
9      };
10     int Arr2[2][2] = {
11         {5,6},
12         {7,8}
13     };
14
15     //행렬의 곱셈의 결과값을 저장하기 위한 3중배열을 선언 및 초기화
16     int Arr3[2][2][2] = {
17         {0,0,0},
18         {0,0,0}
19     };
20
21     int i=0, j= 0, k=0;
22     int result = 0;
23     // 반복문을 사용하여 행렬의 곱셈을 한 뒤 그 결과값을 3중 배열에 저장
24     for (i = 0; i < 2; i++)
25     {
26         for (j = 0; j < 2 ; j++)
27         {
```

```
25     {
26         for (j = 0; j < 2 ; j++)
27         {
28             for (k = 0; k < 2; k++)
29             {
30                 Arr3[i][j][k] = Arr1[i][j] * Arr2[j][k];
31             }
32         }
33     }
34
35     //행렬의 곱셈의 결과를 출력한다
36     for (i = 0; i < 2; i++)
37     {
38         for (k = 0; k < 2 ; k++)
39         {
40             result = 0;
41             for (j = 0; j < 2; j++)
42             {
43                 //printf("%5d", Arr3[i][j][k]);
44                 result+=Arr3[i][j][k];
45             }
46             printf("%5d",result);
47
48         }
49         printf("\n");
50     }
51     return 0;
```