



2023.03.22.(수)

마이크로프로세서

2402 김나영

이중배열과 포인터

```
mic02 > C test01.c > main()
1  #include<stdio.h>
2  int main(){
3      int a[3][4]={
4          {1,2,3,4},
5          {5,6,7,8},
6          {9,10,11,12}
7      };
8      int (*p)[4] = a;
9      printf("%p\n", *p);
10     printf("%p\n", *a);
11     printf("%d\n", p[2][1]);
12     printf("%ld\n", sizeof(a));
13     printf("%ld\n", sizeof(p));
14     return 0;
15 }
```

-2차원 배열은 행단위로 주소 증가함

-1만큼 이동시 주소 8씩 이동함

$a[0][0]$
 $== (*a)[0]$
 $== *(a[0]+0)$

함수 포인터와 함수 호출

```
mic02 > C test02.c > main()
1  #include <stdio.h>
2  int add(int a, int b){
3      return a+b;
4  }
5  int main(){
6      int (*pfunc)(int, int);
7      pfunc = add;
8      printf("결과값 : %d\n", pfunc(1,2));
9      printf("결과값 : %d\n", add(1,2));
10     return 0;
11 }
```

- pfunc를 add 함수를 가리키도록 설정함

함수 포인터와 함수 호출

```
1  #include <stdio.h>
2  int add(int a, int b){
3      return a + b;
4  }
5  int sub(int a, int b){
6      return a - b;
7  }
8  int main(){
9      int (*fp)(int, int);
10     fp = add;
11     printf("결과값 : %d\n", fp(10,20));
12     fp = sub;
13     printf("결과값 : %d\n", fp(10,20));
14 }
```

- fp를 add 함수를 가리키도록 설정 -> 출력

- fp를 sub 함수를 가리키도록 설정 -> 출력

구조체

```
mic02 > C test04.c > ...
1  #include <stdio.h>
2  struct score{
3      int mic;
4      int lin;
5      int pro;
6  };
7  int main(){
8      struct score s1;
9      s1.mic = 60;
10     s1.lin = 70;
11     s1.pro = 80;
12     printf("각 점수 출력 : %5d %5d %5d\n", s1.mic, s1.lin, s1.pro);
13     return 0;
14 }
```

- 구조체 변수 **s1**을 선언
- **s1.mic = 60, s1.lin = 70, s1.pro = 80**와 같이 각 멤버에 값을 할당함
→ 출력

구조체 배열(1)

```
mic02 > C test05.c > main()
1  #include <stdio.h>
2  struct score
3  {
4      char name[20];
5      int mic, lin, pro;
6  };
7  int main(){
8      struct score s[2] = {0};
9      for (int i=0; i<2; i++){
10         scanf("%s", s[i].name);
11         scanf("%d", &s[i].mic);
12         scanf("%d", &s[i].lin);
13         scanf("%d", &s[i].pro);
14     }
15     for(int i=0; i<2; i++){
16         printf("각 점수 출력 : %5d %5d %5d\n", s[i].mic, s[i].lin, s[i].pro);
17     }
18 }
```

- 구조체 변수 s[2]을 선언함
- for 문을 사용하여 두 명의 학생의 이름과 세 과목 점수를 입력, 구조체 배열의 각 요소에 저장 -> 출력

구조체 배열(2)

- 구조체에 별칭을 지정한 후, 포인터를 이용하여 구조체 멤버에 접근함

함수 ShowData:
Person의 포인터를 매개변수로 받아서 구조체의 멤버들을 출력
함수main:

Person의 변수 ps를 생성하고, 포인터 ptr을 이용하여 ps의 주소를 가리킴->구조체 Person의 멤버들 출력

```
mic02 > C test06.c > main(void)
1  #include <stdio.h>
2  typedef struct person
3  {
4      char name[20];
5      int num;
6      unsigned int age;
7  }Person;
8
9  void ShowData(Person* ptr){
10     printf("이름: %s \n", (*ptr).name);
11     printf("번호: %d \n", (*ptr).num);
12     printf("나이: %d \n", (*ptr).age);
13 }
14
15 int main(void){
16     Person ps = {"이순신", 10, 24};
17     Person * ptr = &ps;
18     ShowData(ptr);
19     return 0;
20 }
```

구조체 변수 길이

```
mic02 > C test07.c > main()
1  #include <stdio.h>
2  typedef struct test1{
3      char ch;
4      int num;
5      short sh;
6  }Test1;
7  typedef struct test2{
8      char ch;
9      short sh;
10     int num;
11 }Test2;
12
13 int main(){
14     Test1 t1;
15     Test2 t2;
16     printf("%ld %ld\n", sizeof(t1), sizeof(t2));
17 }
```

함수 main:

**Test1과 Test2 구조체의 변수
t1, t2를 생성하고, sizeof 연산
자를 이용하여 각 구조체의 크기를
출력**

**Test1크기: 12byte
Test2 크기: 8byte**

유니언(같은 메모리 공간에 다양한 데이터 타입으로 접근)

mic02 > C test08.c > main(void)

```
1  #include <stdio.h>
2  typedef union num{
3      char a;
4      short b;
5      int c;
6  }Num;
7
8  int main(void){
9      Num n;
10     n.c = 0x12345678;
11     printf("%x\n", n.a);
12     printf("%x\n", n.b);
13     printf("%x\n", n.c);
14     return 0;
15 }
```

함수 main:

유니언 변수 n을 선언하고,
int형 데이터 0x12345678
을 n.c에 저장

n.a: 78(16진수로 0x78)
n.b: 5678(16진수로 0x5678)
n.c: 12345678(16진수로
0x12345678)