C 복습

포인터 및 파일 입출력

2016.09.20

황슬아

seula.hwang@cnu.ac.kr

목차

포인터

1. 포인터 변수

1) 값을 저장하는 변수가 아닌 변수의 주소를 저장하는 변수

2) 참조 연산자 : &

- ✓ &의 오른쪽에 있는 피연산자의 주소 값을 반환
- ✓ 예) &num : num의 주소를 반환

3) 포인터 *

- ✓ 포인터가 가리키는 메모리 공간에 접근할 때 사용
- ✓ 예) *pnum : pnum이 가리키는 메모리 공간의 값을 가져옴

```c
#include <stdio.h>

int main()
{
    int num = 6;
    int *pnum;
    pnum = &num;

    printf("num의 값 : %d \n", num);
    printf("num의 주소 : %p \n", &num);
    printf("pnum의 값 : %p \n", pnum);
    printf("pnum이 가리키는 값 : %d \n", *pnum);

    return 0;
}
```

```
num의 값 : 6
num의 주소 : 0x7fffe13aedd4
pnum의 값 : 0x7fffe13aedd4
pnum이 가리키는 값 : 6
```

포인터

2. Call by value

1) 인자로 넘기는 값을 복사해서 새로운 함수에게 넘겨주는 방식

2) 값을 복사해서 전달하기 때문에 원본의 값이 변경될 가능성이 없음

```c
#include <stdio.h>

void sum(int a, int b);

int main()
{
    int a = 3;
    int b = 4;

    sum(a, b);
    printf("==== main ===== \n");
    printf("a : %d, b : %d \n", a, b);
    return 0;
}
void sum(int a, int b)
{
    printf("call sum function... \n");
    a += b;
    printf("a : %d, b : %d \n", a ,b );
}
```

```
call sum function...
a : 7, b : 4
==== main =====
a : 3, b : 4
```

포인터

3. Call by reference

1) 주소 값을 인자로 새로운 함수에게 넘겨주는 방식

2) 원본의 값이 변경될 가능성이 있다.

```c
#include <stdio.h>

void sum(int *a, int *b);

int main()
{
    int a = 3;
    int b = 4;

    sum(&a, &b);
    printf("==== main ===== \n");
    printf("a : %d, b : %d \n", a, b);
    return 0;
}
void sum(int *a, int *b)
{
    printf("call sum function... \n");
    *a += *b;
    printf("a : %d, b : %d \n", *a , *b );
}
```

```
call sum function...
a : 7, b : 4
==== main =====
a : 7, b : 4
```

따라 해보기

1. swap 함수 작성하기

1) 포인터를 사용하지 않고 swap 함수를 작성한다.

- gcc를 통해 컴파일 한 뒤 실행 결과를 확인한다.

```c
#include <stdio.h>

void swap(int a,int b)
{
        int temp = a;
        a=b;
        b=temp;
}

int main()
{
        int a=3;
        int b=4;
        swap(a,b);
        printf(" a : %d , b : %d \n",a,b);
        return 0;
}
```

따라 해보기

2) 포인터를 사용하여 swap 함수를 작성한다.

- gcc를 통해 컴파일 한 뒤 실행 결과를 확인한다.
- 앞서 포인터를 사용하지 않은 swap 함수와 어떤 차이가 있는지 생각해본다.

```c
#include <stdio.h>

void swap(int *a,int *b)
{
        int temp = *a;
        *a=*b;
        *b=temp;
}

int main()
{
        int a=3;
        int b=4;
        swap(&a,&b);
        printf(" a : %d , b : %d \n",a,b);
        return 0;
}
```

따라 해보기

2. 배열 이름과 포인터의 관계

1) int 형 배열 array를 선언한다
2) array와 array[0]의 주소(&array[0])를 출력해본다
3) array+1과 array[1]의 주소(&array[1])를 출력해본다
4) 출력한 결과를 비교하고, array 배열의 이름(예. array[0], array[1]..) 과 포인터 (예. array, array_1…)의 관계를 생각해본다.

```c
#include <stdio.h>

int main()
{
    int array[5] = {1, 2, 3, 4, 5};

    printf("&array[0] : %p, array : %p \n", &array[0], array);

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int array[5] = {1, 2, 3, 4, 5};

    printf("&array[1] : %p, array : %p \n", &array[1], array+1);

    return 0;
}
```