

교학습내용



♀ 9.1 포인터 개요





C프로그래밍및실습



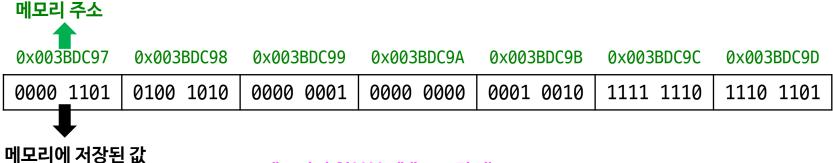


1. 포인터 개요

- 2. 포인터 선언과 사용
- 3. 배열과 포인터
- 4. 포인터 연산
- 5. 포인터 인자와 주소 반환
- 6. 포인터 배열
- 7. 다중 포인터



- 프로그램이 실행되기 위해 필요한 정보(값)을 저장하는 공간
- 1 byte (8 bits) 단위로 물리 주소가 부여 되어 있음
- 개념적으로, 메모리는 일렬로 연속되어 있는 크기가 1 byte 인 방들의 모음이라고 볼 수 있음
- 일반적으로 주소의 길이는 4 bytes이고, 주소는 16진수로 표현



메모리의 일부분 예(byte 단위)



변수와 메모리의 관계

- 변수는 선언될 때, 메모리에 그 변수를 위한 공간이 할당됨
 ❖ (주의) 변수에 할당되는 메모리의 주소는 시스템마다 다르다.
- <mark>주소연산자(&)</mark>: 변수에 할당된 메모리 공간의 시작 주소를 구해 줌

주소

0x003BDC97	0x003BDC98	0x003BDC99	0x003BDC9A	0x003BDC9B	0x003BDC9C	0x003BDC9D
0000 1101	0000 0000	0000 0000	0000 0000	0000 0000	1111 1110	1110 1101

a에 할당된 메모리 공간 (4bytes): 한번 할당되면 고정됨



■ C프로그램에서 변수의 의미는 (2가지)

- 1. 그 변수에 **할당된 공간**을 의미 (주소를 뜻하는 것은 아님)
 - ✓ 선언 or 대입문의 왼쪽 변수(I-value)로 사용될 때
- 2. 그 변수에 저장된 값을 의미
 - ✓ 대입문의 오른쪽 변수(r-value), 조건식, 함수의 인수로 사용될 때

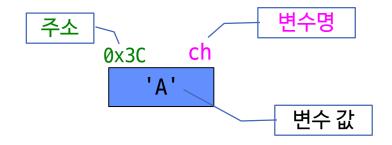
c1

```
char c1, c2; // c1, c2를 위한 공간을 메모리에 할당
c1 = c2; // c1에 c2를 저장 → c1의 공간에 c2에 저장된 값을 저장
if(c1 < c2) // c1이 c2보다 작으면
             // → c1에 저장된 값이 c2에 저장된 값보다 작으면
printf("%c",c1); // c1을 인수로 전달 → c1에 저장된 값을 전달
주소
    0x003BDC97
               0x003BDC98
                          0x003BDC99
                                    0x003BDC9A
                                               0x003BDC9B
     0000 1101
               0000 0000
                          0000 0000
                                    0000 0000
                                               0001 1010
```



포인터(자료형)

- **주소**를 나타내는 특수 자료형
- 주소는 기본적으로 양의 정수로 표현됨 하지만, int(정수형 자료형)와 구별되어 처리됨 (다른 자료형)
- 변수를 나타내는 메모리 그림







- 친구에게 동영상을 전달하려면(보여주려면)?
 - 1. 동영상을 복사해서 준다.
 - 2. 동영상이 저장된 장소만 알려준다.
 - ✔ 이름
 - ✓링크
- 프로그램에서 데이터를 전달하려면?
 - 1. 데이터를 복사해서 준다.
 - 2. 데이터가 저장된 장소만 알려준다.
 - **✓** 이름
 - ✓ 주소(포인터)
- C 언어에서는 주소를 직접적으로 조작하는 방법 제공



高い 학습 정리 **정리**

- 메모리는 프로그램이 실행되기 위해 필요한 정보(값)을 저장하는 공간으로, 1 byte (8 bits) 단위로 물리 주소가 부여 되어 있음
- 변수는 선언 시에 그 변수를 위한 메모리 공간이 할당되고, 할당된 위치는 변경되지 않음
- **포인터**는 메모리 주소를 나타내는 개념으로, 저장 공간을 가리키는 것이라는 의미임
- 변수의 주소는 주소 연산자(&)를 사용하여 얻을 수 있음

