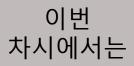




목차

- 1) 포인터 개요
- 2) 포인터 선언과 사용
- 3) 배열과 포인터
- 4) 포인터 연신





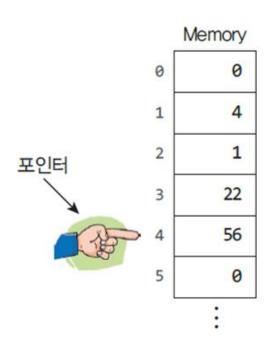
- •변수의 주소
- •포인터의 선언



포인터란?

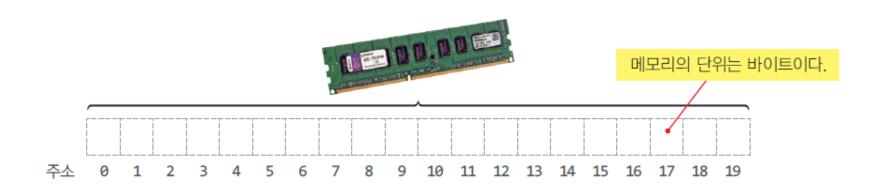
□ *포인터(pointer):* 주소를 가지고 있는 변수





메모리의 구조

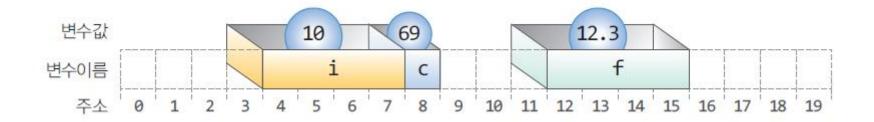
- □ 변수는 메모리에 저장된다.
- □ 메모리는 바이트 단위로 액세스된다.
 - □ 첫번째 바이트의 주소는 0, 두번째 바이트는 1,…



변수와 메모리

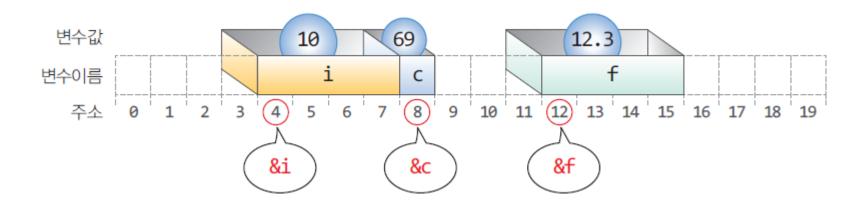
- □ 변수의 크기에 따라서 차지하는 메모리 공간이 달라진다.
- □ char형 변수: 1바이트, int형 변수: 4바이트,…

```
int main(void)
{
  int i = 10;
  char c = 69;
  float f = 12.3;
}
```



변수의 주소

- □ 변수의 주소를 계산하는 연산자: &
- □ 변수 i의 주소: &i



변수의 주소

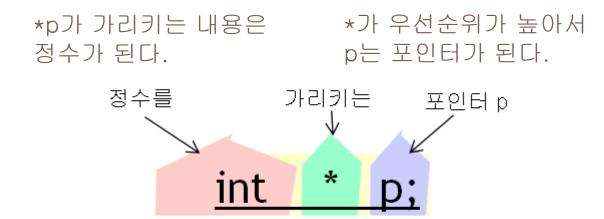
```
int main(void)
{
    int i = 10;
    char c = 69;
    float f = 12.3;

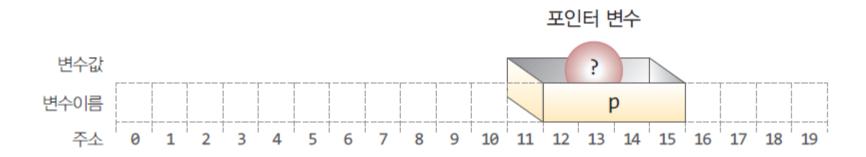
    printf("i의 주소: %u\n", &i);
    printf("c의 주소: %u\n", &c);
    printf("f의 주소: %u\n", &f);
    return 0;
}
```

```
[의 주소: 3209736
c의 주소: 3209727
f의 주소: 3209708
계속하려면 아무 키나 누르십시오 . . . . .
```

포인터의 선언

□ 포인터: 변수의 주소를 가지고 있는 변수



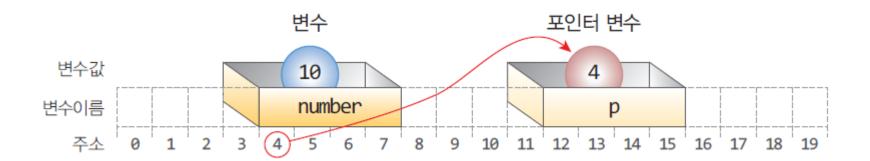


포인터와 변수의 연결

```
      int number = 10; // 정수형 변수 number 선언

      int *p; // 포인터 변수 p 선언

      p = &number; // 변수 number의 주소가 포인터 p로 대입
```



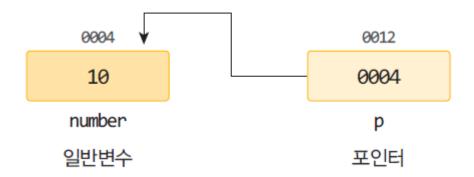
포인터와 변수

□ 포인터 p가 변수 number를 가리킨다.

```
      int number = 10; // 정수형 변수 number 선언

      int *p; // 포인터 변수 p 선언

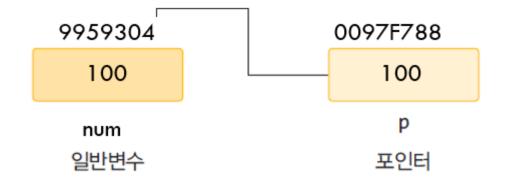
      p = &number; // 변수 number의 주소가 포인터 p로 대입
```



[문제]

□ 정수형 변 num에 100을 기억, 포인터 변수 선언 후, 아래와 같이 주소를 각각 출력해보자.

```
int num = 100; // 정수형 변수 num 선언
```



정수형 변수 : 100 포인터 변수 : 100

정수형 변수 주소 : 9959304 <u>포인터</u> 변수 주소 : 0097F788

감사합니다.

이번 차시에서는

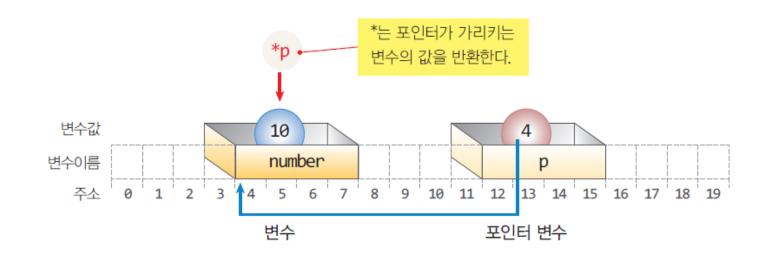




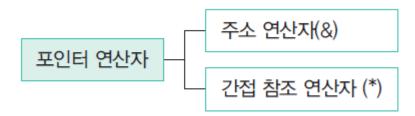
간접 참조 연산자

□ 간접 참조 연산자 *: 포인터가 가리키는 값을 가져오는 연산자

```
int i=10;
int *p;
p =&i;
printf("%d", *p):
```

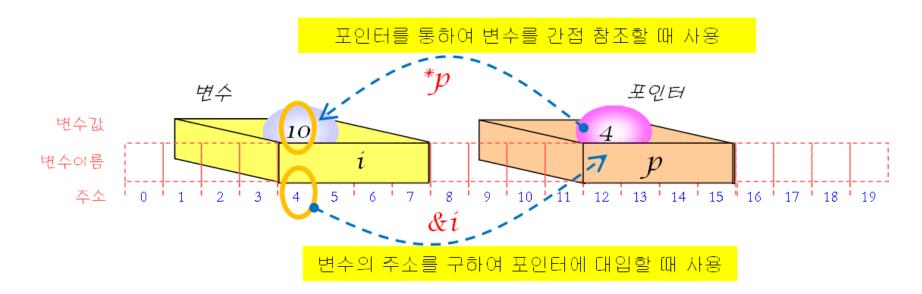


포인터 연산자



& 연산자와 * 연산자

- □ & 연산자: 변수의 주소를 반환한다
- □ * 연산자: 포인터가 가리키는 곳의 내용을 반환한다.



예제 #1

```
#include <stdio.h>
int main(void)
   int number = 10;
   int *p;
   p = & number;
   printf("변수 number의 주소 = %u\n", &number);
   printf("포인터의 값 = %u\n", p);
   printf("변수 number의 값 = %d\n", number);
   printf("포인터가 가리키는 값 = %d\n", *p);
   return 0;
```

```
© C:\Windows\system32\cmd.exe

변수 number의 주소 = 3276052
포인터의 값 = 3276052
변수 number의 값 = 10
포인터가 가리키는 값 = 10
계속하려면 아무 키나 누르십시오 . . . . .
```

예제 #2

```
#include <stdio.h>
int main(void)
        int number = 10;
        int *p;
         p = & number;
         printf("변수 number의 값 = %d\n", number);
         *p = 20;
         printf("변수 number의 값 = %d\n", number);
        return 0;
```

```
© C:#Windows#system32#cmd.exe
변수 number의 값 = 10
변수 number의 값 = 20
계속하려면 아무 키나 누르십시오 . . .
```

배열의 시작 주소

- □ 배열로 선언된 arr[5]는 순차적으로 메모리에 나열되어 있다.
- □ 즉, 배열의 첫 번째 항목인 arr[0] 항목의 시작 주소는 arr 변수의 전체 시작 주소와 동일한 구조이다.



□ 배열 시작 주소를 알기 위해서는 &arr[0]을 사용한다.

배열의 시작 주소

arr 배열의 시작 주소를 포인터 변수 p에 저장해서 arr 배열의 각 항목을 사용하고 싶다면 아래와 같이 코드를 작성한다.

```
int arr[5];
int *p;
p=&arr[0];
```

배열의 이름은 배열의 시작 주소 &arr[0]이다.

[문제]

배열에 $arr[5]=\{5,3,2,7,9\}$ 을 기억, 포인터 연산자를 통해 기억된 배열 값과 기억된 주소 값을 배열 명령을 통해, 포인터 변수를 통해 동일하게 출력되는지 출력해 보자.

```
**배열 값 출력 **
5 ---10156008
3 ---10156012
2 ---10156016
7 ---10156020
9 ---10156024

**포인터 연산 출력 **
5 ---10156008
3 ---10156012
2 ---10156016
7 ---10156020
```

9 --- 10156024

중간 점검

- □ 메모리는 어떤 단위를 기준으로 주소가 매겨지는가?
- □ 포인터도 변수인가?
- □ 변수의 주소를 추출하는데 사용되는 연산자는 무엇인가?
- □ 변수 x의 주소를 추출하여 변수 p에 대입하는 문장을 쓰시오.
- □ 정수형 포인터 p가 가리키는 위치에 25를 저장하는 문장을 쓰시오.



감사합니다.