C 프로그래밍 및 실습

[14주차 1강]
포인터(1)

학습 내용

9.1 포인터 개요

학습 목표

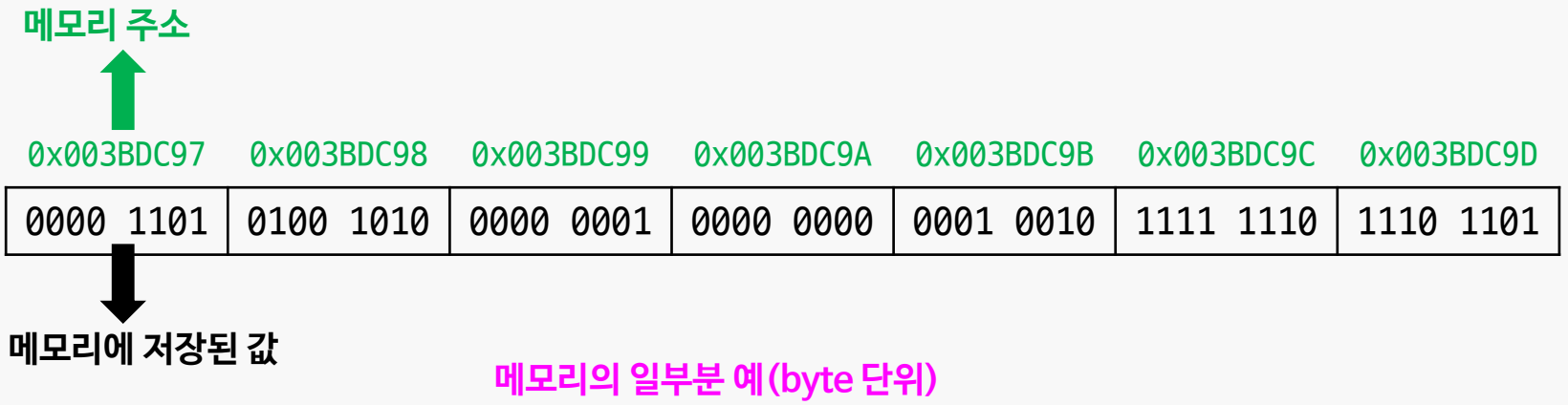
9.1 포인터란 무엇인지 이해한다.

C 프로그래밍 및 실습

9.1 포인터 개요

메모리

- 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)

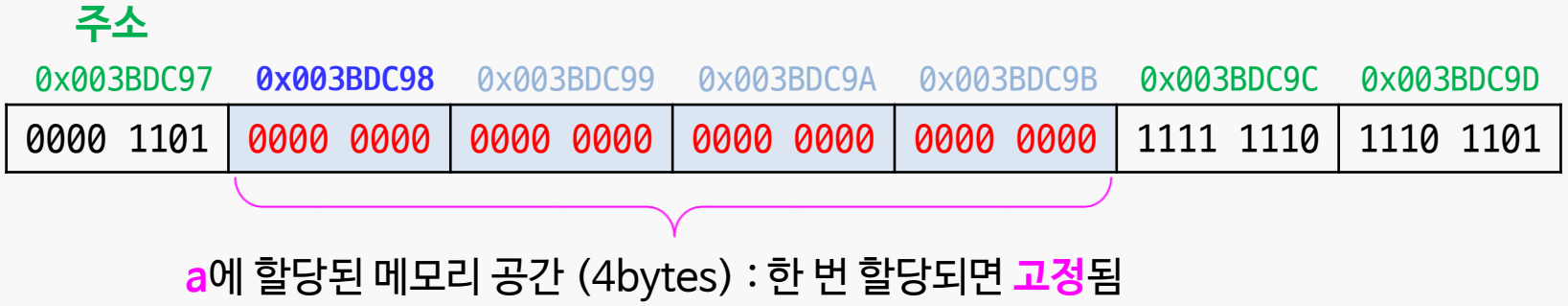
세종대학교

변수와 메모리의 관계

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

```
int a = 0;
printf("%d, %p", a, &a);  // %p: 주소를 16진수로 출력

결과:
0, 003BDC98                    ⇨ 주소 값은 다르게 나올 수 있음
```

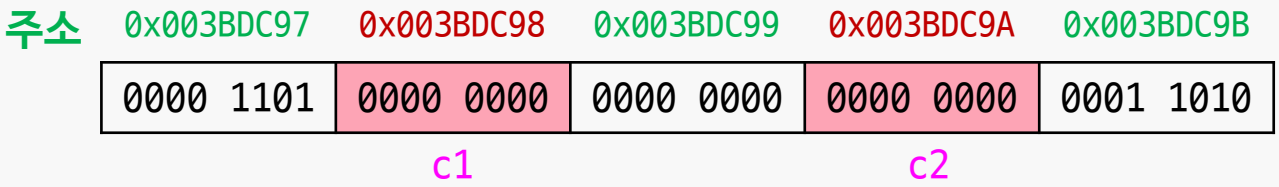


a에 할당된 메모리 공간 (4bytes) : 한 번 할당되면 **고정**됨

C프로그램에서 변수의 의미는 (2가지)

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

```
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
```



실습 하기

[예제 9.1]
아래와 같이 선언된 변수들의 주소를 출력하고, 출력된 주소를 보고 메모리에 변수가 할당된 모습을 그림으로 그려보자.

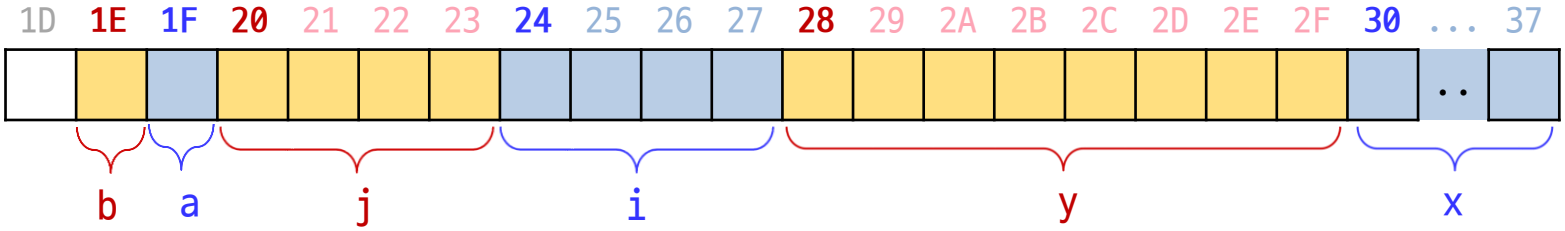
```
char a, b;
int i, j;
double x, y;
```

출력 예시

```
a: 0018F91F, b: 0018F91E
i: 0018F924, j: 0018F920
x: 0018F930, y: 0018F928
```

- 메모리에 변수가 할당된 모습
  - ✓ 연속으로 할당된다는 보장은 없음

주소 (마지막 두 자리만 표시)



실습하기

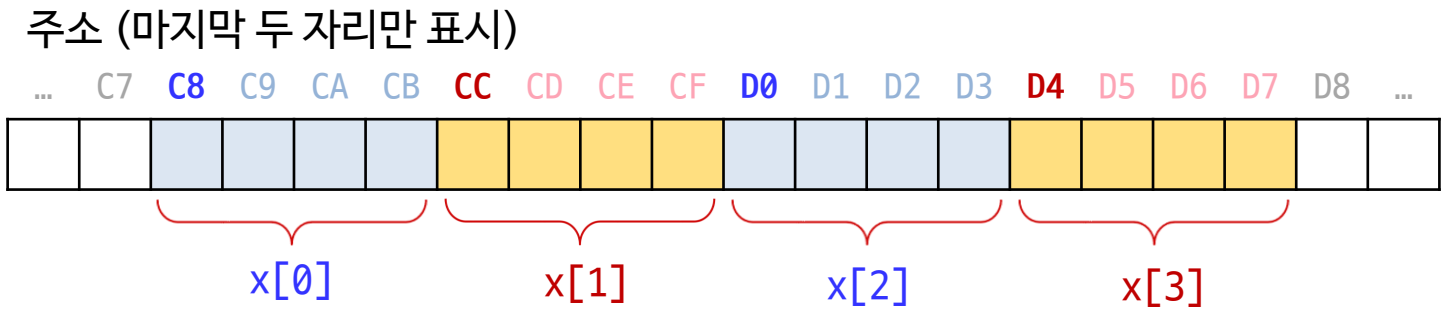
[예제 9.2]
아래와 같이 배열 원소들의 주소를 출력하고, 출력된 주소를 보고 메모리에 변수가 할당된 모습을 그림으로 그려보자.

```
int x[4];
```

출력 예시

```
x[0]: 001FFEC8
x[1]: 001FFECC
x[2]: 001FFED0
x[3]: 001FFED4
```

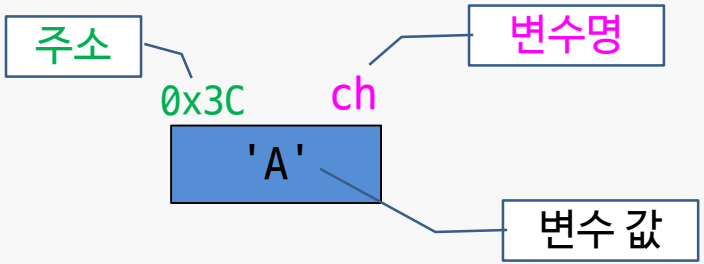
- 메모리에 배열이 할당된 모습
  - ✓ 배열은 항상 연속된 공간에 할당됨



세종대학교

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



세종대학교

포인터 맛보기 1

- **선언** : 변수 명 앞에 * (참조연산자) 사용
- **연결** : & (주소연산자)를 이용해 포인터 변수를 다른 변수에 연결
- **참조** : * (참조연산자)를 이용해 포인터에 연결된 변수에 접근

```
char ch;
char *pch;          // 포인터 변수 선언

pch = &ch;          // 포인터 대입(연결)

ch = 'A';

printf("ch = %c\n", ch);
printf("*pch = %c\n", *pch);  // 참조
```

실행 결과

```
ch = A
*pch = A
```

포인터 맛보기 2

```
int num = 3;
int *pnum;                    // 선언

pnum = &num;                  // 연결

printf("num = %d\n", num);
printf("*pnum = %d\n", *pnum);

*pnum = 5;                    // 참조

printf("num = %d\n", num);
printf("*pnum = %d\n", *pnum);
```

실행 결과

```
num=3
*pnum=3
num=5
*pnum=5
```

학습 정리

- **메모리**는 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