

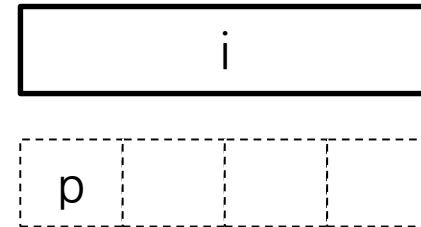
실습 구조체 (1 1/3)

실습 0 - 뒷 페이지의 코드 보고 이해하기

- ◆ 0~255 사이의 숫자 4개를 저장하려 한다. 그런데 우리가 가지고 있는 변수는 int형 변수 1개 뿐이다. int형 변수 1개에 숫자 4개를 저장할 수 있을까?
 - 힌트 : 0~255 사이의 숫자 : unsigned char형, 1바이트
 - 따라서 4개의 unsigned int형을 int형 변수에 담을 수 있다.
- ◆ 방법 1. 포인터를 이용하기
- ◆ 방법 2. 공용체를 이용하기

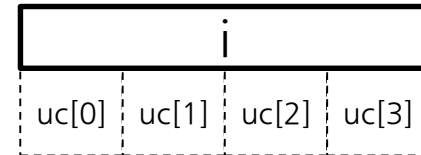
실습 0 - 포인터 이용하기

```
int main(void) {  
    int i;  
    unsigned char* p;  
  
    p = (unsigned char*)&i;  
  
    *p = 10;  
    *(p + 1) = 20;  
    *(p + 2) = 30;  
    *(p + 3) = 40;  
  
    printf("%d %d %d %d\n", *p, *(p + 1), *(p + 2), *(p + 3));  
}
```



실습 0 - 공용체 사용하기

```
union _U {  
    int    i;  
    unsigned char uc[4];  
};
```



```
int main(void) {  
    union _U u;  
  
    printf("%zd\\n", sizeof(u));  
  
    u.uc[0] = 10;  
    u.uc[1] = 20;  
    u.uc[2] = 30;  
    u.uc[3] = 40;  
  
    printf("%d %d %d %d\\n", u.uc[0], u.uc[1], u.uc[2], u.uc[3] );  
}
```

실습 1.

- ◆ 0~7 사이의 숫자 4개를 저장하려 한다. 그런데 우리가 가지고 있는 변수는 short형 변수 1개 뿐이다. short형 변수 1개에 숫자 4개를 저장할 수 있을까?
 - 힌트 : 0~7 사이의 숫자 : 3비트 사용
 - 3비트 * 4개 = 12비트, short형 하나는 2바이트=16비트. 따라서 가능
- ◆ 방법 2. 비트구조체 + 공용체를 이용하기

실습 1.

```
union _U {
    short    i;
    struct _B {
        unsigned char c1 : 3;
        unsigned char c2 : 3;
        unsigned char c3 : 3;
        unsigned char c4 : 3;
    } byte;
};

int main(void) {
    union _U u;

    printf("%zd\n", sizeof(u));

    u.byte.c1 = 1;
    u.byte.c2 = 2;
    u.byte.c3 = 3;
    u.byte.c4 = 4;

    printf("%d %d %d %d\n", u.byte.c1, u.byte.c2, u.byte.c3, u.byte.c4 );
}
```

실습 2.

- ◆ 책은 번호, 제목, 출판사로 구성된다. 출판사는 출판사 이름, 전화번호, 지역으로 구성된다. 이 구조의 변수를 하나 만든 후, 사용자로부터 입력받은 정보를 넣어라. 단, 구조체는 중첩된 구조체로 만들어야 한다.
- ◆ 구조체 변수에 다음의 값을 넣은 후 화면에 표시하라.
 - 출판사 정보 : 중앙출판사, 010-1111-2222, 서울
 - 책 정보 : 1, 프로그래밍, 중앙출판사

실습 2.

```
struct PUBLISHER {
    char name[12];
    char phone[14];
    char area[10];
};

struct book {
    int number;
    char bookname[20];
    struct PUBLISHER pub;
};

int main(void) {
    // 사용자로부터 입력받지 않고 바로 변수를 초기화한 경우
    struct book mybook = { 1, "프로그래밍", {"중앙출판사", "010-1111-2222",
"서울"}};

    printf("%d %s %s %s %s\n", mybook.number, mybook.bookname,
mybook.pub.name, mybook.pub.phone, mybook.pub.area );
}
```

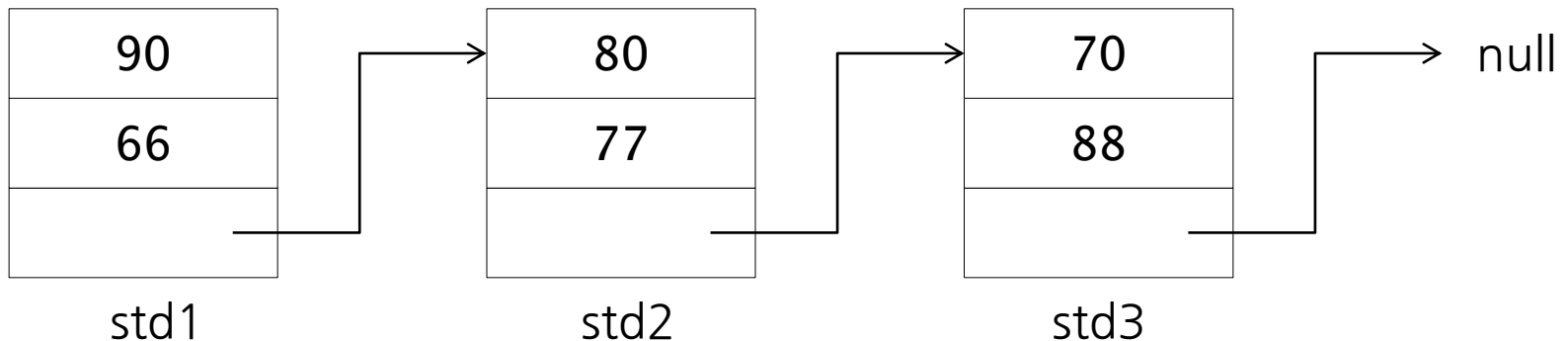

실습 3

- ◆ 다음 구조체를 정의하자.

```
struct STUDENT {  
    int kor;  
    int eng;  
    struct STUDENT *mystd;  
};
```

다음 페이지의 코드를 보자.

- ◆ 세 개의 변수를 만들고 다음과 같이 데이터를 채운 후 연결하자.
(std1.mystd = &std2; std2.mystd = &std3; std3.mystd = NULL))



실습 3

```
struct STUDENT {  
    int kor;  
    int eng;  
    struct STUDENT* mystd;  
};  
  
int main(void) {  
    struct STUDENT std3 = { 70,88, NULL }, std2 = { 80,77,&std3 },  
        std1 = {90,66,&std2};  
}
```

실습 3-1

◆ 하나의 포인터 변수를 선언한 후 첫 번째 데이터를 가리키자.

- `struct STUDENT *pstd;`
- `pstd = &std1;`

◆ for 문과 pstd를 이용하여 데이터를 모두 출력하라.

- 출력양식)
90 66
80 77
70 88

◆ while 문을 이용하여 동일한 출력을 하라.

연결 구조로 이루어져 있으므로 첫 변수만 알면 뒤로 따라갈 수 있다.

실습 3-1

```
int main(void) {
    struct STUDENT std3 = { 70,88, NULL }, std2 = { 80,77,&std3 }, std1 = { 90,66,&std2 };
    int i;
    struct STUDENT* pstd = &std1;

    for (i = 0; i < 3; i++) {
        printf("%d %d\n", pstd->kor, pstd->eng);
        pstd = pstd->mystd;
    }

    while (pstd) {          // 의미를 생각해 보자.
        printf("%d %d\n", pstd->kor, pstd->eng);
        pstd = pstd->mystd;
    }
}
```