C 프로그래밍 1
Lecture Note #17

백윤철
ybaek@smu.ac.kr

내용

- 함수의 인자, 리턴값으로 구조체 사용
- 구조체의 크기
- 구조체의 이점
- 중첩된 구조체

함수의 인자로 전달되고 반환되는 구조체 변수

```
typedef struct {
  int xpos;
  int ypos;
} Point;
void ShowPosition(Point pos) {
  printf("[%d, %d]\n", pos.xpos, pos.ypos);
}
Point GetCurrentPosition(void) {
  Point cen;
  printf("Input current pos: ");
  scanf("%d %d", &cen.xpos, &cen.ypos);
  return cen;
}
```

구조체 변수 cen의 값이 통째로 반환된다.

함수의 인자로 전달되고 반환되는 구조체 변수

```
int main(void) {
   Point curPos = GetCurrentPosition();
   ShowPosition(curPos);
   return 0;
}
```

ShowPosition 함수의 매개변수에
curPos에 저장된 값이 통째로 복사된다.

실행결과

```
Input current pos: 2 4
[2, 4]
```

배열까지도 통째로 복사

```
typedef struct person {
  char name[20];
  char phoneNum[20];
  int age;
} Person;

void ShowPersonInfo(Person man) {
  printf("name: %s\n", man.name);
  printf("phone: %s\n", man.phoneNum);
  printf("age: %d\n", man.age);
}
```

구조체의 멤버로 배열이 선언된 경우
구조체 변수를 인자로 전달하거나 반환 시
배열까지도 통째로 복사가 이뤄진다.

배열까지도 통째로 복사

```
Person ReadPersonInfo(void) {
  Person man;
  printf("name? "); scanf("%s", man.name);
  printf("phone? "); scanf("%s", man.phoneNum);
  printf("age? "); scanf("%d", &man.age);
  return man;
}
int main(void) {
  Person man = ReadPersonInfo();
  ShowPersonInfo(man);
  return 0;
}
```

실행결과

```
name? Jung
phone? 010-12XX-34XX
age? 22
name: Jung
phone: 010-12XX-34XX
age: 22
```

포인터로 구조체 변수 전달하기

```
typedef struct point {
  int xpos;
  int ypos;
} Point;
void OrgSymTrans(Point* ptr) { /* 원점 대칭 */
  ptr->xpos = (ptr->xpos) * -1;
  ptr->ypos = (ptr->ypos) * -1;
}
void ShowPosition(Point pos) {
  printf("[%d, %d]\n", pos.xpos, pos.ypos);
}
```

함수 내부에서 값이 바뀌는 경우에 주의

포인터로 구조체 변수 전달하기

```
int main(void) {
  Point pos = { 7, -5 };
  OrgSymTrans(&pos); /* pos의 값을 원점 대칭 이동 */
  ShowPosition(pos);
  OrgSymTrans(&pos); /* pos의 값을 원점 대칭 이동 */
  ShowPosition(pos);
  return 0;
}
```

실행결과

```
[-7, 5]
[7, -5]
```

포인터로 구조체 변수 전달하기

```
typedef struct point {
  int xpos;
  int ypos;
} Point;
void OrgSymTrans(Point* ptr) { /* 원점 대칭 */
  ptr->xpos = (ptr->xpos) * -1;
  ptr->ypos = (ptr->ypos) * -1;
}
void ShowPosition(const Point* pos) {
  printf("[%d, %d]\n", pos->xpos, pos->ypos);
}
```

빠르게 처리됨

포인터로 구조체 변수 전달하기

```
int main(void) {
  Point pos = { 7, -5 };
  OrgSymTrans(&pos); /* pos의 값을 원점 대칭 이동 */
  ShowPosition(&pos);
  OrgSymTrans(&pos); /* pos의 값을 원점 대칭 이동 */
  ShowPosition(&pos);
  return 0;
}
```

실행결과

```
[-7, 5]
[7, -5]
```

구조체 변수를 대상으로 가능한 연산

```
typedef struct point {
  int xpos;
  int ypos;
} Point;
int main(void) {
  Point pos1 = { 1, 2 };
  Point pos2;
  pos2 = pos1; /*pos1의 내용이 pos2로 복사됨 */
  printf("크기: %d\n", sizeof(pos1);
  printf("[%d, %d]\n", pos1.xpos, pos1.ypos);
  printf("크기: %d\n", sizeof(pos2);
  printf("[%d, %d]\n", pos2.xpos, pos2.ypos);
  return 0; }
```

실행결과

```
크기: 8
[1, 2]
크기: 8
[1, 2]
```

구조체를 정의하는 이유

- 연관 있는 데이터를 하나로 묶을 수 있는 자료형 정의 가능
- 연관 있는 데이터를 묶으면 데이터의 표현 및 관리가 용이함
- 데이터의 표현 및 관리가 용이해지면 그만큼 합리적인 코드를 작성할 수 있음

```
typedef struct student {
  char name[20];
  char stdnum[20];
  char school[20];
  char major[20];
  int year;
} Student;
```

구조체를 정의하는 이유

인자 전달 시 용이하다.

```
void ShowStudentInfo(const Student* sptr) {
  printf("학생 이름: %s\n", sptr->name);
  printf("학생 고유번호: %s\n", sptr->stdnum);
  printf("학교 이름: %s\n", sptr->school);
  printf("선택 전공: %s\n", sptr->major);
  printf("학년: %d\n", sptr->year);
}

#define ARR_SIZE 3
int main(void) {
  Student arr[ARR_SIZE];
```

하나의 배열 선언으로
종류가 다른 데이터들을 한
개로 묶어 저장할 수 있다.

구조체를 정의하는 이유

```
  int i;
  for (i = 0; i < ARR_SIZE; i++) {
    printf("이름: "); scanf("%s", arr[i].name);
    printf("번호: "); scanf("%s", arr[i].stdnum);
    printf("학교: "); scanf("%s", arr[i].school);
    printf("전공: "); scanf("%s", arr[i].major);
    printf("학년: "); scanf("%d", &arr[i].year);
  }
  for (i = 0; i < ARR_SIZE; i++) {
    ShowStudentInfo(&arr[i]);
  }
  return 0;
}
```

중첩된 구조체의 정의와 변수의 선언

```
typedef struct point {
   int xpos;
   int ypos;
} Point;

typedef struct circle {
   Point cen;
   double rad;
} Circle;
```

앞서 정의한 구조체는 이후에 새로운 구조체를 선언하는데 있어서 기본 자료형의 이름과 마찬가지로 사용이 될 수 있다.

중첩된 구조체의 정의와 변수의 선언

```
void ShowCircleInfo(const Circle* cptr) {
  printf("[%d, %d]\n", (cptr->cen).xpos,
                       (cptr->cen).ypos);
  printf("radius: %g\n\n", cptr->rad);
}
int main(void) {
  Circle c1 = { { 1, 2 }, 3.5 };
  Circle c2 = { 2, 4, 3.9 };
  ShowCircleInfo(&c1);
  ShowCircleInfo(&c2);
  return 0;
}
```

실행결과

```
[1, 2]
radius: 3.5

[2, 4]
radius: 3.9
```

정리

- 함수의 인자, 리턴값으로 구조체 사용
- 구조체의 크기
- 구조체의 이점
- 중첩된 구조체