Exercise 7-4: struct_pointer2.c

Maximilian Fernaldy - C2TB1702

Note: some links and other HTML-related objects may not work in pdf form. Consider reading the webpage format of the report here.

Instead of displaying information of all members, struct_pointer2.c will only display the selected member's information, using their "ID".

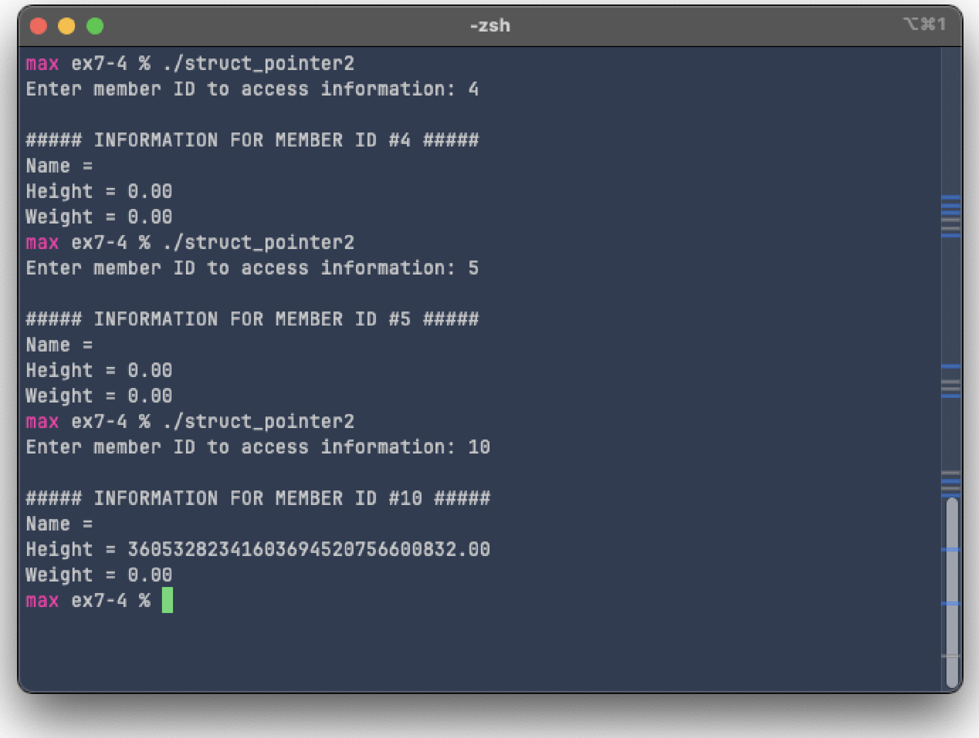
To do this, we can ask the user for the intended user's ID, then pass the pointer to the struct and the member ID to a function:

```
int main() {
    // variable declaration
    st_person member[MEMBER_NUM] =
    {
        {"Ichiro", 160, 50},
        {"Jiro", 170, 60},
        {"Saburo", 180, 70}
    };

    int member_ID = -1;  // Initialize member ID as something invalid
    printf("Enter member ID to access information: ");
    scanf("%d", &member_ID);

    print_person(member, member_ID);
    // Pass memory address of the right member to print function

    return 0;
}
```

Then, for the function itself, we can use pointer arithmetic to access the correct element, then the arrow operator to access the members:

```
void print_person(st_person pointer_member[], int member_ID) {
    printf("\n##### INFORMATION FOR MEMBER ID #%d #####\n", member_ID);
    printf("Name = %s\n", (pointer_member+member_ID)->name);
    printf("Height = %.2f\n", (pointer_member+member_ID)->height);
    printf("Weight = %.2f\n", (pointer_member+member_ID)->weight);
}
```

This program works, and there is **no error returned** by the compiler. However, there is a glaring mistake in *memory safety*. When we enter numbers lower than 0 or higher than 2, the program still runs, but there's something wrong with the values displayed:

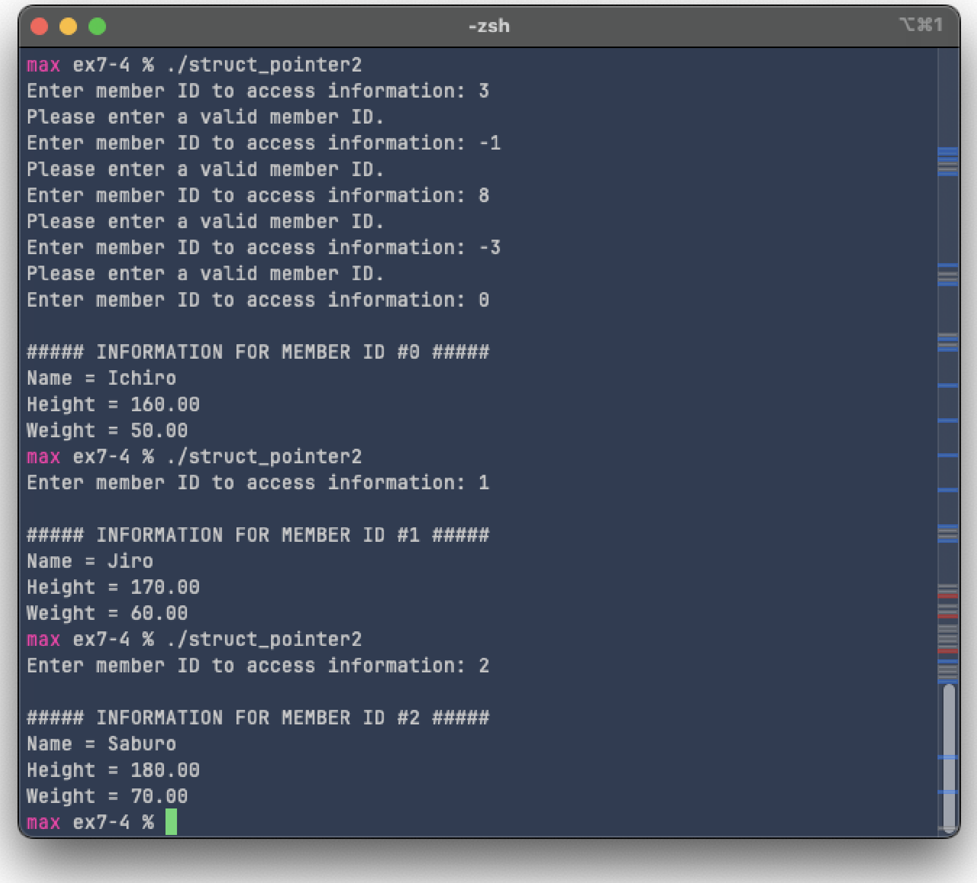


```
max ex7-4 % ./struct_pointer2
Enter member ID to access information: 4

##### INFORMATION FOR MEMBER ID #4 #####
Name =
Height = 0.00
Weight = 0.00
max ex7-4 % ./struct_pointer2
Enter member ID to access information: 5

##### INFORMATION FOR MEMBER ID #5 #####
Name =
Height = 0.00
Weight = 0.00
max ex7-4 % ./struct_pointer2
Enter member ID to access information: 10

##### INFORMATION FOR MEMBER ID #10 #####
Name =
Height = 36053282341603694520756600832.00
Weight = 0.00
max ex7-4 %
```

Either the values are "blank" or they are outlandishly big. This is because the program is accessing information that it's not supposed to access. Our array `members` only contains 3 elements with indices 0 to 2, but we're accessing numbers outside that boundary. As a consequence, the program accesses memory locations that are either uninitialized or used by **other programs** on our system. In this case, we encounter some gibberish, because the program may be trying to interpret data stored in another data type. However, if some malicious actor designs a program to correctly interpret this data, and the program is given access to the memory of our system, we can imagine how dangerous that may be. It's even more dangerous when we give the user access to writing and erasing data to and from memory. This program currently contains **unsafe code**, because a **segmentation fault or memory access violation** is possible. To fix this, we need to confine the user to the valid indices:

```c
int main() {
    // variable declaration
    st_person member[MEMBER_NUM] =
    {
        {"Ichiro", 160, 50},
        {"Jiro", 170, 60},
        {"Saburo", 180, 70}
    };

    int member_ID = -1;  // Initialize member ID as something invalid
    while (member_ID < 0 || member_ID >= MEMBER_NUM) {
        printf("Enter member ID to access information: ");
        scanf("%d", &member_ID);
        if (member_ID < 0 || member_ID >= MEMBER_NUM) {
            printf("Please enter a valid member ID.\n");
        }
    }

    print_person(member, member_ID);
    // Pass memory address of the right member to print function

    return 0;
}
```

The while loop will keep asking the user for a member ID while throwing an alert that the entered one is invalid. Then when a valid member ID is entered, the program can proceed to accessing the memory.



```
max ex7-4 % ./struct_pointer2
Enter member ID to access information: 3
Please enter a valid member ID.
Enter member ID to access information: -1
Please enter a valid member ID.
Enter member ID to access information: 8
Please enter a valid member ID.
Enter member ID to access information: -3
Please enter a valid member ID.
Enter member ID to access information: 0

##### INFORMATION FOR MEMBER ID #0 #####
Name = Ichiro
Height = 160.00
Weight = 50.00
max ex7-4 % ./struct_pointer2
Enter member ID to access information: 1

##### INFORMATION FOR MEMBER ID #1 #####
Name = Jiro
Height = 170.00
Weight = 60.00
max ex7-4 % ./struct_pointer2
Enter member ID to access information: 2

##### INFORMATION FOR MEMBER ID #2 #####
Name = Saburo
Height = 180.00
Weight = 70.00
max ex7-4 %
```