Structures in C

By: Yehia M. Abu Eita

Outlines

- Introduction
- Creating new structure type
- Declaring a structure variable
- Structure variables in memory
- Accessing a structure variable members

Introduction

- Structure is a **non-primitive** data type that stores **different types of data**.
- Structures are used to **group different types of data together**.
- You have to **create** the **new structure** type **first**, and **then** you can **declare** structure variables.
- Structure **size differs** according to the **sizes of its members**.
- Using structures will enhance your code organization.

Creating new structure type

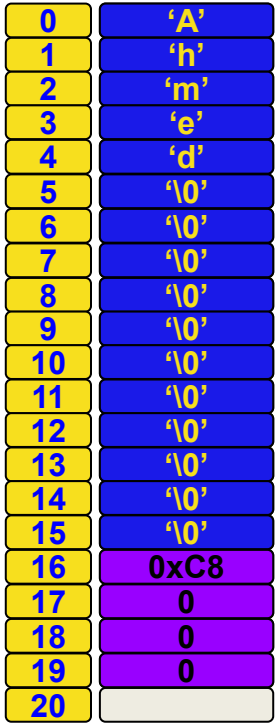
- A creation of the new structure data type must occur first, **written first in the .c** file, before any declaration occurs.
- **Creation** of a data type **doesn't mean** that we **allocate** anything from the **memory**.
- **Creation is just informing the compiler**.

```
struct student
{
    unsigned char name[16];
    unsigned int id;
};
```

Declaring a structure variable

- After creating a new structure data type, you can declare a variable.
- Declaration Example:
  - `struct student student1_data;`
- Also you can set initial values for the structure variable members.
- Definition Example:
  - `struct student student1_data = {"Ahmed", 200};`

Structure variables in memory



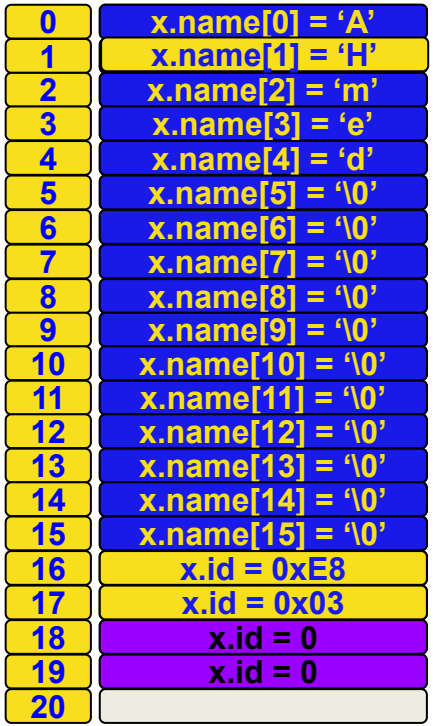
- Defining a structure variable:
  - `struct student student1_data = {"Ahmed", 200};`
- The size of this variable into the memory is **20 bytes**.
  - **16 bytes for the array of characters "name".**
  - **And 4 bytes for the integer member "id".**

```
struct student
{
      unsigned char name[16];
      unsigned int id;
};
```

Accessing a structure variable members

- Use the **‘ . ’ operator** to access the structure members.
- Access structure members is for reading and writing.
- Examples:
  - `struct student x = {“Ahmed”, 200};`
  - `x.name[1] = ‘H’;`
  - `x.id = 1000;  // 0x03E8`

```
struct student
{
      unsigned char name[16];
      unsigned int id;
};
```



Summary

- Now you are familiar with structures in C.
- You can create, declare and manipulate structures.
- You have learned that structure size depends on its members' sizes
- Accessing structures using the ' . ' operator.
- Remember that initial values given to structure members must be in the same order of the members created into the structure.