Structures in C

Shivam Srivastava

Indian Institute of Technology, Madras

shivam.srivastava511@gmail.com

August 11, 2014

Overview

1. Motivation
    - Why Structures?
2. Structures
    - What are they?
3. How do they look?
    - Declaration
4. Structure Variables
    - How do you make them?

Motivation

How will you represent a person or a book or time with just one primitive variable?

Arrays? Can't handle multiple types of data.

Hence, the need for structures in C.

Structures

- A collection of variables under a single name.
- These variables can be of different types. Can even be arrays or structures of other types.
- Each has a name which is used to select it from the structure.
- A convenient way of grouping several pieces of related information together.

How does a Structure look?

General case

```
struct [structure tag]
{
        member definition;
        member definition;
         ...
        member definition;
} [zero or more structure variables];
```

Example

```
struct Books
{
        int price;
        char title[50];
        char author[50];
};
```

Structure Variables

Without typedef: Example1

```
struct Books book1;
```

'book1' is a structure variable.

Without typedef: Example2

```
struct Books
{
        int price;
        char title[50];
        char author[50];
}book1, book2;
```

'book1' and 'book2' both are structure variables.

Structure Variables

With typedef: General Example

'typedef' can be used to redefine any data type in C.

Example1: typedef int INTEGER;

Example2: typedef char* str_ptr;

With typedef: Structure example

```
typedef struct Books_
{
        int price;
        char title[50];
        char author[50];
}Book;
Book book1;//'book1' is a structure variable
```

Array elements are accessed using the subscript variable, similarly structure members are accessed using the dot[.] operator. Let's look at an example.

Questions

Any questions?

Today's lab

Hands-on session on Complex Numbers