Enumerations in C

By: Yehia M. Abu Eita

Outlines

- Introduction
- Creating new enum type
- Declaring an enum variable
- Enum variables in memory

Introduction

- Enumerations are **non-primitive** data types that **represent integers as text**.
- Enumerations allocate **fixed memory size, the largest integer defined**.
- You have to **create** the **new enum** type **first**, and **then** you can **declare** enum variables.
- Using enums will **make your code more readable**.

Creating new enum type

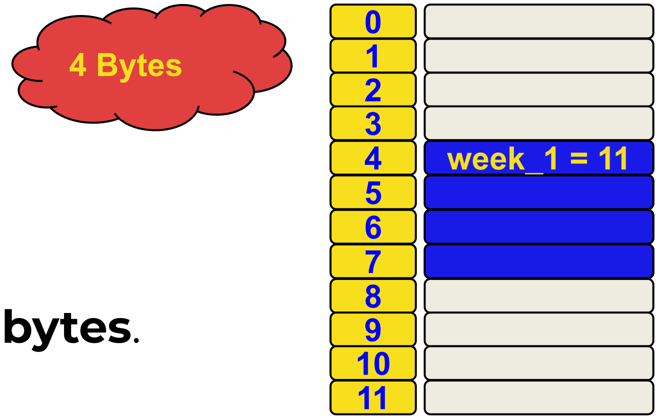
- A creation of the new enum data type must occur first, **written first in the .c** file, before any declaration occurs.
- **Enums start from 0 and increment by 1, if not defined.**
- **You can define different enums with the same value, but you can not do the opposite.**

```
enum week
{
    SAT=10,SUN,MON,TUES,WED,THU
    ,FRI
};
```

Declaring an enum variable

- After creating a new enum data type, you can declare a variable.
- Declaration Example:
  - `enum week week_1;`
- Also you can set **initial value** to **enum** variable **using the defined text**.
- Definition Example:
  - `enum week week_1 = SUN;  //week_1 = 11`

Enum variables in memory



- Defining an enum variable:
  - `enum week week_1 = SUN;`
- The size of this variable into the memory is **4 bytes**.

```
enum week
{
     SAT=10,SUN,MON,TUES,WE
     D,THU,FRI
};
```

Summary

- Now you are familiar with enums.
- You can create, declare enums.
- You have learned that enum size depends the largest integer defined and maximum size decided by the compiler.
- Remember that enum values can not be changed during run-time.
- You can define different enums with the same value, but you can not do the opposite.