Typedefs

By: Yehia M. Abu Eita

Outlines

- Introduction
- Famous typedefs
- Typedef for complex declarations

Introduction

- Typedefs are used to define a **new type** from an already **existing data type**.
- Typedefs must be **written first** in the .c files **before any declarations**.
- Using typedefs properly will give you a good **indication** of the used **type** and its **size**.
- Using typedefs make your code more **readable**.
- Using typedefs make it **easier** for you to write **complex declarations**.

Famous typedefs

- Example on famous typedefs:
  - `typedef unsigned char uint8_t;`
  - `typedef signed char int8_t;`
  - `typedef unsigned short uint16_t;`
  - `typedef signed short int16_t;`
  - `typedef unsigned int uint32_t;`
  - `typedef signed int int32_t;`
  - `typedef unsigned long long uint64_t;`
  - `typedef signed long long int64_t;`

Typedef for complex declarations

- **Three steps** to make a typedef for a complex declaration:
  - Write a **normal declaration** of a variable.
  - Write **typedef before** this declaration.
  - Change **variable name** to the **new type name**.
- **Example 1:**
  - `typedef unsigned char arrOfFiveChars_t[5];`
  - `arrOfFiveChars_t x; // x is an array of 5 characters`
- **Example 2:**
  - `typedef struct student {uint8_t name[50]; int32_t id;} ST_student_t;`
  - `ST_student_t student1; // student1 is a structure of type struct student`

Summary

- Now you are familiar with typedefs
- Now you are able to make a typedef for both simple and complex declarations.
- Remember, typedefs must be written first in the .c file.
- Remember, give a meaningful names to the new types.