4-2 Journal: Best Coding Practices

Joel Meza

Professor Ricky Sethi

January 26, 2022

CS 350 – Emerging Systems Architectures & Technologies

Southern New Hampshire University

There is a variety of categories to initialize and implement best coding practices for the Embedded C program system language. Every rule is required to be followed as a category to prevent a pitfall to happen in Embedded C, which are known to be data type, scope of variables, integral statements, methods/procedures, and white spaces. In the white-space category, the rule is followed as all spaces are required to be positioned and from local/global variables as struct, class, and assigned operators must be arranged accordingly. Those spaces that are left blanked should require to be apart from other statements. As for indenting in a method purpose, the required guideline must be followed as four characters from the beginning of each line.

Next, the module category requires for appropriate explicit naming, template, source files, and header files. The data types require the explicit naming as well, with proper library that will help interpret the implicit header file to the source file. Examples of these data type names are signed/unsigned integers, signed/unsigned characters, null pointers, booleans, strings, parameters/arguments, structures, instance/constant variable types, and unions. Methods/Procedures category needs to comply when naming explicit functions, interrupt service iterations, and executable threads. Statements categories need to be precise and compliant for the iterative loops like the switch, conditional, assertions for unit testing, and declared variables to be reliable for the code to work properly.

Although still there are some usual pitfalls that may accumulate when coding from Embedded C system programming language. To be precise one usual pitfall can be when positioning spaces in which they are not properly suitable or being in the state of denial when allocating these spaces to the corresponding position. This may lead to difficulties in discerning from blocks of code, unreliable code layout, undetectable bugs creations, etc. in trying to figure where or when it went wrong. Additionally, usual pitfall can also occur when not using module rules where it causes frequent issues like implicit declarations, redundant coupling, inaccurate behavior, etc. Now, if a software engineer doesn’t enforce their safe, secure, and reliable best coding practices or follow the code category rules as mentioned then they will acquire misbehavior outcomes. When following these rules, the concentration strength to avoid compiled errors can be minimized or preventable which makes debugging and testing simply better. As well when working with the statement category rule, it can help to maximize the code implementation and distribution from the building compiler to cross compiler before getting into the targeted compiler of an actuator or machine (device). Which may also increase the possibility in preventing security issues within the managed Embedded C’s categorized code rules.

Finally, it’s mandatory to engage in best coding practices when working in Embedded C system programming language to avoid vulnerabilities in security and issues with compiled errors. Eventually when following the best coding rules and practices, its guaranteed to help any software engineer to acknowledge the code and will also help them to easily review and write accordingly, thus makes it easy to find bugs and mistakes within the program’s code.