Programmierregeln

(aus Bollow, Homann, Köhn; C und C++ für embedded Systems, mitp-Verlag 2009)

Wichitge Regelwerke:

MISRA - Motor Industry Software Reliability Association

Fb 812 – Bundesanstalt für Arbeitsschutz und Arbeitmedizin

Eisenbahn Bundesamt

CENELEC – European Commitee for Electrotechnical Standardization

GNU-Coding Standars

Google Coding Standards

…

Warum:

- Lesbarkeit
- sicher, robust(unerwatete Fehler)
- Portierbarkeit
- Überschaubarkeit(Module)
- Vermeidung von Fehlern

Was:

1. Regeln für die Programmierung
   - Einschränkungen der Verwendeten Sparche
   - Art der Implementierung
   - Schreibweisen
   - Namen
   - Coderichtilinien
   - Kommentierung
2. Regeln für die Beschreibung von Schnittstellen und deren Verwendung
3. Regeln für den Einsatz von Methoden

Regeln (Angelehnt an Fb 812)

Allgemein

A1 – Programmentwurf mit Hilfe von Werkzeugen

Es sollen Werkzeuge verwendet werden, die schon bei der Spezifikation eine Validierung der Anforderungen erlauben(Früherkennung von Fehlern).

A2 – Einheitliche Programmierregeln

In einem Projekt sind einheitliche Regeln für:

- Programmierstil
- Dokumentation
- Programmlayout

Überprüfung möglichst durch Werkzeuge. (pc lint, Splint, Artistic Style, ...)

A3 – Kein Wechsel der Programmiersprache im Modul

A4 – Quelldateien nicht größer als 500 Zeilen

A5 – Es ist ausschließlich der ASCII Zeichensatz zu verwenden(Steuerzeichen 0 – 30, Zeichen 31 – 127)

Regeln für C

C1 – Standarddatentypen

Vermeidung von Datentypen, die Plattformabhängig sind(z.B. Int).

Eindeutig sind: uint8, uint16, unint32, int8, int16, int32, …

C2 – Operatoren

Um Schreibfehler zu vermeiden sollten einige Operatoren mit eindeutigen Namen versehen werden.

z.B. == → eq != → not_eq

% → mod && → and & → bitand, ….

C3 – Kodierung von Grundstrukturen

Es sind nur folgende Elemente zugelassen:

- einzelne Verarbeitungsschritte
- Unterprogramm
- Einfache- und Mehrfachverzweigungen (switch kann auch leere case enthalten, jeder mögliche Fall muß berücksichtigt werden, default muß vorhanden sein)
- Bedingte Scheifen(while ist do..while vorzuziehen)
- break und continue sind möglichst zu vermeiden, return nur einmal pro Funktion, und exit nur im Fehlerfall

C4 – Präprozessorbefehle

Folgende sind zugelassen:

- #include < >, “ “
- #ifdef, #ifndef, #elif, #else, #endif
- #pragma mit Vorsicht, bei Compilerwechsel Funktion überprüfen!

C5 – Schlüsselwörter und Namen der Standardbibliothek dürfen nicht umdefiniert werden

C6 – Include-Dateien sind gegen Mehrfacheinbindung zu schützen#

#ifndef _*Name*_H

#define _*Name*_H

Anweisungen

#endif

C7 – Funktionsprototypen ausschließlich in Headerdateien

C8 – Fehlermeldungen und Warnungen

Es müssen alle Fehlermeldungen und Warnungen eingeschaltet sein. Warnungen sind als Fehler zu behandeln. Es ist eine statische Codeanalyse mit einem anderen Werkzeug(z.B. pclint) oder Compiler durchzuführen.

C9 – Für Komplexe Datenstrukturen sind typedef, struct und union(mit Vorsicht) zugelassen

C10 – Kommentare nur mit / .. */, keine Verschachtelung*

Kommentare sollen übersichtlich sein, offensichtliches ist nicht zu kommentieren. Mit Intuition das Wesentliche kommentieren (interessant, informativ und nicht langweilig)

C11 – Bezeichner

- A … Z, a ... z, 0 … 9, "_"
- Erstes Zeichen nur A...Z, a...z
- Bezeichner mit unterschiedlicher Bedeutung dürfen sich nicht nur durch Groß-/Kleinschreibung unterscheiden.
- Bezeichner müssen unterscheidbar und prägnant(sprechend) sein.
- Variablen dürfen nicht für unterschiedliche Inhalte verwendet werden
- Bezeichner für Makros müssen mit Großbuchstaben geschrieben werden

C12 – Konstanten

- Konstanten sind mit const oder enum zu vereinbaren, #define ist nicht zu verwenden.
- Konstanten müssen immer groß geschrieben werden

C13 - Namen

- Namen (Variablen, Funktionen und Felder) sind vor ihrer Verwendung explizit zu deklarieren.
- Variablen und Felder sind vor ihrer ersten Verwendung zu initialisieren. Für Schleifenvariablen der for – Schleife reicht die Initialisierung im Schleifenkopf.
- Keine Variablen, Funktionen und Felder, die nicht verwendet werden.
- Keine erneute Definition von Variablen mit gleichem Namen in untergeordneten Funktionen.
- Schlüsselwörter von C und C++ sowie von Bibliotheken sind nicht zu verwenden.
- Namen sollten sprechend sein(möglicherweise auch ein Typkennzeichen, ungarische Notation).

C14 – Pro Zeile nur eine Zuweisung, komplexe Zuweisungen auf mehrere Zeilen aufteilen

- Strukturierung des Quellcodes
- Einrückungen
- Position von geschweiften Klammern
- Aufteilung von langen Parameterlisten auf mehrere Zeilen
- Anordnung von Kommentaren

C15 – Zuweisungen nur in Anweisungen, nicht in Ausdrücken

- If(x = y) – verboten
- for(…; x == y + 1; …) - verboten, test auf Gleichheit problematisch
- x = y + 8z = 47 – verboten

C16 – Verwendung rekursiver Funktionen

Bei der Verwendung rekursiver Funktionen ist auf eine maximale Rekursionstiefe zu prüfen.

```
If ( rekursivcount < MAX_REKURSIV_COUNT)
{
        rekursivcount++;
        rekursivfunktion();
}
```

C17 – Pointer

- Zeiger sind vor dem ersten Gebrauch zu initialisieren (z.B. NULL)
- Es dürfen keine Zeiger auf Variablen verwendet werden, die nicht im Gültigkeitsbereich der Variablen liegen.
- Zeiger auf Funktionen sind zu vermeiden
- Eine Funktion darf keinen Zeiger auf eine lokale Variable zurückliefern.
- Zeiger vom Typ void* sind nur bei der dynamischen Speicherverwaltung zulässig.

Schnittstellen

S1 – Modulbeschreibungen

- Jedes Modul besteht aus einer Header- und einer Quellcodedatei
- Aufgabe des Moduls, Beschreibung der Algorithmen(allgemein)
- Handhabung des Moduls(Initialisierung, Verwendung)
- Beschreibung der Schnittstellen
- Autor, Version, Datum, Änderungen, Import, Export, Tests, …

S2 – Funktionsbeschreibung

- Aufgabe der Funktion
- Bedingungen für den Aufruf der Funktion
- Beschreibung der Parameterliste
- Autor, Version, Datum, …

Kodierregeln

- Die Verwendung von Tabulatoren im Code und in Kommentaren ist nicht zulässig.
- Alle geschwungenen Klammern stehen in einer Zeile ohne Code.
- Schleifenrümpfe und Zweiganweisungen werden mit den geschweiften Klammern eingerückt.
- Alle Einrückungen haben 3 Leerzeichen.
- Lange Zeilen werden in der Folgezeile rechtsbündig formatiert.
- Nur ein Statement pro Zeile.
- Eine Codezeile darf 55 Zeichen nicht überschreiten.
- Jede Datei enthält am Seitenanfang zwei Kommentarzeilen zur Verwendung des Programms.
- Operatoren werden zwischen Leerzeichen gesetzt.
- Jede Variable wird in einer eigenen Zeile deklariert.
- Variablennamen beginnen bei der Deklaration in der gleichen Spalte.
- Variablennamen beginnen mit einem Kleinbuchstaben. Trennungen sind durch Großbuchstaben oder _ möglich.
- Variablen sollen sprechende Namen haben, Ausnahmen sind Scheifenzähler mit begrenztem Wirkungsbereich(i,j,k,...).
- Schleifenzähler sollten ganzzahlig sein.
- Typen (auch struct, …) werden mit Großbuchstaben geschrieben.
- Casting ist explizit durchzuführen.
- Testtiefe des Compilers oder anderer Tools immer auf der höchsten Stufe.
- Ein Compilerdurchlauf darf keine Warnungen erzeugen.