

CT2 Praktikum

Coding Style in C

1 Einleitung

In diesem Praktikum lernen Sie Coding Guidelines gemäss der Linux Kernel Entwicklung kennen.



Ziele der Einhaltung von Coding Guidelines sind:

- Reduktion der Anzahl Bugs: Der Code wird fehlerfreier, robuster und besser wartbar.
- Vereinfachung der Zusammenarbeit / Codelesbarkeit im Team: Es wird weniger Zeit benötigt den Code eines anderen Teammitglieds zu lesen.
- Erhöhung der Portabilität: Standardisierung vereinfacht die Wiederverwendung des Codes auf anderen HW Plattformen.

2 Lernziele

- Sie können die Vorteile von Coding Guidelines erklären.
- Sie kennen den vorgestellten Coding Styles.
- Sie können eigene C Programme nach diesen Coding Guildelines korrigieren und / oder schreiben.
- Sie können erläutern, was ein Beautifier ist und wie Sie ihn nutzen.

3 Aufgaben

3.1 Erarbeitung der Coding Guidelines

Studieren Sie das beigelegte File „coding_guidelines_handout.pdf“. Diskutieren Sie die verschiedenen Punkte mit Ihrem Nachbarn:

- Was ist der Hintergrund der einzelnen Regeln?
- Warum könnte eine Regel sinnvoll sein?
- Wo stimmen Sie (nicht) zu und warum?

3.2 Einarbeitung in die Anwendung

Im vorgegebenen Programmrahmen finden Sie ein Programm, welches sich compilieren lässt und läuft. Es wurden jedoch (absichtlich) einige Regeln ignoriert.

Was ist die Funktion des Programms (Stichworte)?

3.3 Coding Style im Source Code

Im vorgegebenen Programmrahmen finden Sie ein Programm, welches sich zwar compilieren lässt und läuft. Es wurden jedoch (absichtlich) einige Regeln ignoriert. Finden Sie mindestens 10 Stellen im Code, an denen die Coding Guidelines nicht eingehalten wurden. Kennzeichnen Sie die Stellen im Sourcecode und notieren Sie in der untenstehenden Tabelle, was falsch gemacht wurde und wie Sie sie korrigieren würden.

Zeile	Beschreibung	Korrekturvorschlag

3.4 Sourcecode korrigieren

Erstellen Sie eine Kopie des original Source Codes. Korrigieren Sie darin die gefundenen Fehler in der Kopie. Machen Sie ein Vergleich der beiden Dateien, um die Korrekturen hervorzuheben. Dazu können Sie Tools wie *meld* (Linux im Repository, Windows: <http://meldmerge.org/>) oder einen Online-Vergleich (z.B. <https://www.diffchecker.com/>) nutzen.

3.5 Beautifier installieren und nutzen

Für die C-Praktika stellen wir Ihnen ein Tool zur Verfügung, welches des Sourcecode automatisch auf (einen Teil der) Coding Guidelines anpassen kann, einen sogenannten Beautifier. Installieren Sie den Beautifier Uncrustify, wie in der Dokumentation auf dem Wiki (<https://ennis.zhaw.ch/>) beschrieben ist.

Machen Sie eine weitere Kopie des original Source Codes und bearbeiten Sie diese mit dem Beautifier. Vergleichen Sie das Resultat mit dem Original und mit Ihrer Korrektur.

Welche Stellen hat Uncrustify gefunden, welche nicht? Markieren Sie diese in der Tabelle zu Aufgabe 3.3.

4 Bewertung

Bewertungskriterien	Gewichtung
Tabelle zu Aufgabe 3.3 ausgefüllt (Min. 10 Positionen)	1/2
Quelltext korrigiert und Änderungen in der Tabelle markiert gemäss Aufgaben 3.4 und 3.5	1/2