

Programmiertipps 'Numerische Methoden der Physik' SS 2014

Bastian Knippschild, Christian Jost und Mitarbeiter

Programmieren zu lernen ist teilweise schwer und oft gibt es keinen einfachen Startpunkt. Hier sind einige gesammelte Tipps, wie man sich verbessern kann. Diese Auflistung ist bei weitem nicht vollständig, Anmerkungen sind daher herzlich willkommen.

Programmierstil und Kommentare: Das Aussehen des Codes wird häufig unterschätzt. Ein Grund dafür ist, dass die meisten Leute davon ausgehen, dass nur sie selbst mit dem Code arbeiten. Allerdings hat diese Sichtweise zwei große Fehler. Einerseits kann es sein, dass der Code nach einigen Monaten oder Jahren weiter verwendet wird. Auch dann ist es einfacher klar strukturierten und kommentierten Code zu verstehen. Andererseits wird aus einigen Projekten früher oder später ein Gruppenprojekt. Es ist dann einfacher den Code weiter zu entwickeln, wenn alle den gleichen Stil verwenden. Der Stil wird meistens durch einen *Styleguide* vorgegeben. Darin ist zum Beispiel festgelegt, aus wievielen Leerzeichen ein TAB besteht und wie lang eine Codezeile maximal sein darf. Darüber hinaus ist auch festgelegt, wie mit auftretenden Fehlern umgegangen wird. Zum Beispiel verwendet C++ *exceptions*, während in C die Fehlercodes durch Rückgabe eines Zahlenwertes behandelt werden. Man kann seinen eigenen Styleguide entwickeln, oder sich andere aneignen. Ein bekannter style guide ist der Google C++ Styleguide. Auf der Webseite sind auch alle Entscheidungen zu diesem Styleguide begründet.

10-Finger schreiben: Es hilft beim Programmieren ungemein blind schreiben zu können. Es gibt einige Webseiten, auf denen man das Blindschreiben üben kann, zum Beispiel TypingWeb.

Text Editoren: Ähnlich hilfreich wie das blinde schreiben ist das Kennen des Editors, den man benutzt. Es ist egal welchen Editor man benutzt, allerdings sollte man sich mit dem gewählten Editor gut auskennen. Unter Linux sind vim und emacs sehr verbreitet. Diese Editoren können vollständig über die Tastatur gesteuert werden. Außerdem können sie sehr stark an eigene Bedürfnisse angepasst werden. Auch Autovervollständigung und automatische Formatierung des Codes können eingestellt werden. Das vollständige Erlernen eines Editors benötigt allerdings Zeit und Geduld, lohnt sich jedoch durch kürzere Entwicklungszeiten am Ende.

Compiler: Der Compiler ist das wichtigste Werkzeug eines Programmierers. Er übersetzt den Code in Maschinensprache. Daher ist es wichtig zu wissen, was der Compiler kann. Um Programme zu beschleunigen, können, zum Beispiel für den GNU Compiler, die Optionen '-O2 -mtune=native -march=native' benutzt werden. So können Programme teilweise um einen Faktor 5 oder mehr beschleunigt werden. Außerdem haben moderne Compiler, wie der GNU Compiler, viele Funktionen um die Entwicklung und das Debuggen von Programmen zu unterstützen und zu beschleunigen. Wichtig ist vor allem, dass die Compiler Optionen bekannt sind, und nicht, welchen Compiler man tatsächlich benutzt.

Externe Bibliotheken: Externe Bibliotheken haben meistens einige Vorteile gegenüber eigenem Code. Vor allem werden sie ständig getestet und aktualisiert. Daher sind sie meistens sehr stabil

und einfach zu benutzen. Dadurch sinkt auch die Entwicklungszeit für eigene Programme. Gerade wissenschaftliche Bibliotheken sind häufig stark optimiert um kurze Laufzeiten zu erreichen. Einige Beispiele sind FFTW für Schnelle Fourier Transformationen, Eigen und Armadillo für Matrixoperationen mit dicht besetzten Matrizen oder PETSc und SLEPc für Matrixoperationen mit dünn besetzten Matrizen. Zwei allgemeinere Bibliotheken, die viele verschiedene Funktionen haben, sind die Gnu Scientific Library (GSL) und die Boost C++ library. Die GSL hat verschiedenste wissenschaftliche Funktionen, von der Berechnung von Mittelwerten bis hin zu Zufallszahlengeneratoren. Die BOOST Bibliothek umfasst mehr allgemeine Funktionen wie Container Klassen, reguläre Ausdrücke für C++ und eine Bibliothek für Kommandozeilenoptionen.

Bücher und andere Wissensquellen: Bücher und Webseiten können bei spezifischen Problemen sehr hilfreich sein, aber auch allgemein beim Programmieren helfen. Eines der besten Bücher über das Programmieren im Allgemeinen ist 'Der Pragmatische Programmierer'. Das Buch gibt eine Einführung in allgemeine Programmiertechniken und ist unabhängig von einer Programmiersprache. Außer Google gibt es Webseiten wie Stackoverflow, auf denen viele hilfreiche Beiträge und Lösungen zu verschiedensten Problemen gefunden werden können. Dort gibt es auch eine Auflistung der besten C++ Bücher.