

Grundkurs Compilerbau

» Hier geht's
direkt
zum Buch

DAS VORWORT

Vorwort

*Von allen Welten, die der Mensch erschaffen hat, ist die der Bücher die Gewaltigste.
– Heinrich Heine*

Lieber Leser, liebe Leserin, lassen Sie uns eintauchen in dieses Buch über Compilerbau.

Compilerbau ist eine der ältesten Disziplinen der Informatik, und ohne Compilerbau gäbe es keine höheren Programmiersprachen, keine großartigen Frameworks und Werkzeuge, keine tollen Anwendungen.

Viele Bücher sind über Compilerbau geschrieben worden – dieses Buch entstand aus einer Veranstaltung im Grundstudium Informatik an der Technischen Hochschule Mittelhessen, einer Hochschule für angewandte Wissenschaften. Meine Kollegen Hellwig Geisse und Michael Jäger haben diese Veranstaltung über viele Jahre entwickelt, ich habe sie etwas weiterentwickelt.

Dem Anspruch einer Hochschule für angewandte Wissenschaften entsprechend haben Michael Jäger und Hellwig Geisse zusätzlich zu der üblichen Vorlesung und den Übungen ein Praktikum entwickelt, bei dem die Studierenden innerhalb eines Semesters für eine kleine Programmiersprache einen vollständigen Compiler vom Einlesen des Quelltextes bis zur Ausgabe des Assembler-Codes entwickeln.

Dieses Buch greift daher diesen Faden auf und enthält von Anfang bis Ende immer wieder Code-Fragmente, die Sie nutzen können, um selbst einen Compiler zu schreiben. Natürlich bleibt noch einiges an Code übrig, den Sie selbst entwickeln müssen. Ich hoffe, dass Ihnen die Anstöße helfen, um zu verstehen, wie ein Compiler praktisch aufgebaut ist und wie man selbst daraus einen vollständigen Compiler entwickeln kann.

Beweise, zum Beispiel für die Korrektheit der Algorithmen, werden Sie in diesem Buch *nicht* finden. Stattdessen werden Sie viele Beispiele finden, um die Algorithmen zu erklären und die getroffenen Aussagen zu belegen.

Die am häufigsten an Hochschulen gelehrt Programmiersprache ist Java, und wir werden uns sowohl bei der Diskussion über Elemente von Programmiersprachen, die kompiliert werden müssen, als auch bei den Code-Beispielen für unseren Compiler oft an Java orientieren. Zusätzlich werden wir aber auch C als weitere sehr populäre Sprache

betrachten – als ein Exemplar von Programmiersprachen, das alle wesentlichen Bestandteile prozeduraler Sprachen enthält.

Ich möchte an dieser Stelle zuerst meinen beiden Kollegen Michael Jäger und Hellwig Geisse dafür danken, dass sie mir Zugang zu ihrer Compilerbau-Veranstaltung gewährt haben und mir immer mit gutem Rat zur Seite standen. Für den Anstoß und die Ermunterung zu diesem Buch bin ich ihnen sehr dankbar.

Bei der Weiterentwicklung hatte ich Unterstützung von studentischen Hilfskräften, von denen ich Lukas Gail und Niklas Deworetzki besonders hervorheben möchte.

Großer Dank geht auch an meinem Lektor im Rheinwerk Verlag, Herrn Christoph Meister, der mich bei der Konzeption und im langen Prozess des Schreibens immer unterstützt hat.

Besonders möchte ich mich nochmals bei meinem Kollegen Michael Jäger bedanken, der die mühevollen Arbeit des Fachgutachtens auf sich genommen hat und viele hilfreiche Hinweise und Verbesserungsvorschläge hatte.

Für die mühsame Arbeit des Korrekturlesens geht großer Dank an (in alphabetischer Reihenfolge): Friederike Daenecke, Lukas Gail, David Meyer und Björn Pfarr.

Zuletzt und am wichtigsten geht mein besonderer Dank an meine Frau Renate, die die Arbeit an diesem Buch oft an Wochenenden, im Urlaub und spät abends nicht nur toleriert, sondern immer auch ermutigt hat.

Vorwort zur zweiten Auflage

Liebe Leserinnen, liebe Leser,

nach dem großen Interesse an der ersten Auflage dieses Buches folgt nach kurzer Zeit nun schon eine zweite Auflage, die zum einen einige Fehler korrigiert, die sich leider eingeschlichen hatten, und die zum anderen neue Features der Sprache Java ab der Version 21 benutzt, die es Ihnen ermöglichen eine alternative Vorgehensweise zum Visitor-Pattern zu nutzen. Des Weiteren betrachten wir eine Methode zur Darstellung von Zwischencode – das sogenannte Static-Single-Assignment –, das in vielen modernen Compiler wie unter anderem auch im Java HotSpot-Compiler verwendet wird.

Ich danke den Leserinnen und Lesern, die über die verschiedenen Kanäle Kommentare zu dem Buch abgegeben haben und natürlich denen, die mich auf Ungenauigkeiten und Fehler hingewiesen haben.