

Programmieren in C++

SS 2010

Vorlesung 2, Mittwoch 28. April 2010
(Compiler und Linker, .h und .cpp Dateien)

Prof. Dr. Hannah Bast
Lehrstuhl für Algorithmen und Datenstrukturen
Institut für Informatik
Universität Freiburg

Blick über die Vorlesung heute

■ Organisatorisches

- Feedback zu Ihren Erfahrungen
- Fristverlängerung für das 1. Übungsblatt / Termine im TF-Pool

■ Compiler und Linker

- Was macht der Compiler?
- Was macht der Linker?
- Warum die Trennung in `.h` und `.cpp` Dateien?
- Konsequenzen für das Makefile

■ Continuous Build System

- Was ist das? Wie hilft uns das?

■ Editor

- Eine kleine Einführung in `Vim` (am Ende, nur wer will)

Ihre Erfahrungen bisher

■ Zusammenfassung / Auszüge

- das Drumherum hat am meisten Zeit gekostet
 - Linux, Gtest, SVN, Vim, ...
- das Programmieren ging eigentlich (wegen Vorlage)
- viel auf dem Web nachguckt / auf dem Wiki gefragt
- warum keine IDE (integrated development environment)?
- Lint ist cool
- Lint ist doof
- Lint ist zu pingelig
- Probleme mit Lint bei ... } else { ...
- Vim ist ganz schön kompliziert

- Fristverlängerung für das 1. Übungsblatt
 - Um eine Woche = Sie haben also jetzt für das 1. + 2. Übungsblatt zusammen bis zum 5. Mai um 16 Uhr Zeit
 - Bis dahin sollten Sie das Drumherum und das Drumherum um das Drumherum im Griff haben
- Termine im TF-Pool
 - Donnerstag, 16 – 18 Uhr, Gebäude 082 – SR 00 029
 - Freitag, 16 – 18 Uhr, Gebäude 082 – SR 00 028
 - Dienstag, 16 – 18 Uhr, Gebäude 082 – SR 00 028
 - Es gibt auch noch ein paar Termine von 9 – 11 Uhr
 - will die jemand?

- Es gab "Beschwerden" zu cpplint.py
 - manche fanden manche Regeln nicht toll
 - mag sein, aber später für die Zusammenarbeit mit anderen ist es essentiell sich an [ein](#) Stylesheet zu halten
 - manche fanden Lint zu pingelig
 - je länger man programmiert, desto mehr versteht man den Grund hinter dieser und jener Pingeligkeit (sie haben alle einen Grund, und zwar einen guten)
 - manche hatten Probleme mit dem [else](#)
 - das war ein Bug den wir inzwischen gefixt haben, mit [svn update](#) bekommen Sie die neueste Version

- Das ist der Editor den ich benutze
 - Sie können aber irgendeinen Editor benutzen
 - Es sollte aber einer sein der viel kann
 - Wenn Sie noch keine Präferenz haben und / oder aus anderen Gründen gerne Vim benutzen wollen, gebe ich am Ende dieser Vorlesung eine kleine Einführung
 - Da brauchen dann nur die bleiben, die das auch interessiert
 - Ich werde dann im Anschluss auch noch für allgemeine Fragen zu Details zur Verfügung stehen
(SVN, gtest, make, lint, oder was immer Sie beschäftigt)

■ Der Compiler

- ... übersetzt den Programmcode für die einzelnen Dateien in die Maschinensprache
 - Eingabe: eine `.cpp` Datei
 - Ausgabe: eine `.o` Datei
 - Syntax : `g++ -c MyCode.cpp`
 - Insbesondere hat man dann also in der Regel viele `.o` Dateien
- Dabei ist es ok, wenn in einer `.cpp` Datei Funktionen benutzt werden, die in anderen `.cpp` Dateien definiert sind

■ Der Linker

- Macht aus den ganzen `.o` Dateien ein Programm
- Syntax: `g++ -o MyExecutable MyCode1.o MyCode2.o ...`
- In einer der Dateien muss die `main` Funktion definiert sein
- Wenn in einer der `.o` Dateien Funktionen definiert werden, die in keiner der anderen `.o` Dateien definiert sind gibt es eine Fehlermeldung (die häufigste vom Linker):
 - `Undefined reference ...`
 - Das bedeutet entweder ...
 - ... dass Sie die entsprechende Funktion nirgends implementiert haben
 - ... oder dass Sie dem Linker die entsprechende `.o` Datei nicht übergeben haben

- Was ist eine Bibliothek (library) in C++ ?
 - im Prinzip dasselbe wie eine `.o` Datei: kompilierter Code für eine Reihe von Funktionen
 - mit einem Index dabei, damit man (bzw. der Linker) schnell den Code für eine bestimmte Funktion findet
 - Syntax z.B. `g++ -o MyMain MyMain.o -lgtest`
 - der Linker sucht dann nach einer Datei `libgtest.a` (statisch) oder `libgtest.so.0` o.ä. (dynamisch)
 - er sucht in ein paar Standardverzeichnissen, z.B. `/usr/lib`, `/usr/local/lib`, usw.
 - mit der Option `-L<my directory>` kann man sagen, wo er noch suchen soll

Continuous Build System (Hudson)

■ Features

- Läuft auf einem separaten Rechner
- "Baut" Ihr Programm periodisch oder wann immer Sie etwas im SVN ändern
 - `make build`
 - `make test`
 - `make lint`
- Über ein Web Interface können Sie sich die einzelnen "Builds" bequem anschauen

Wir arbeiten gerade noch dran, aber bis Ende der Woche können auch Sie es benutzen, wir sagen über den Wiki Bescheid wenn es soweit ist

■ Compiler und Linker

- Online Manual zum g++ Version 4.5

<http://gcc.gnu.org/onlinedocs/gcc-4.5.0/gcc/>

- Linker Optionen von eben diesem

<http://gcc.gnu.org/onlinedocs/gcc-4.5.0/gcc/Link-Options.html#Link-Options>

- Wikipedias Erklärung zu Compiler und Linker

<http://en.wikipedia.org/wiki/Compiler>

[http://en.wikipedia.org/wiki/Linker_\(computing\)](http://en.wikipedia.org/wiki/Linker_(computing))

- Statische und dynamische Bibliotheken

[http://en.wikipedia.org/wiki/Library_\(computing\)](http://en.wikipedia.org/wiki/Library_(computing))

