

Programmieren in C++

SS 2018

Vorlesung 12, Dienstag 17. Juli 2018
(Evaluation, Projekt, Optimierung, Lehrstuhl)

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

Blick über die Vorlesung heute

■ Organisatorisches

- Erfahrungen mit dem Ü11 .h Dateien für Projekt
- **Letzte Vorlesung heute**

■ Inhalt

- Auswertung Evaluation Zusammenfassung + Ausblick
- Projekt Haben Sie noch Fragen?
- Optimierung Maschinencode, Option -O3
- Vorstellung Lehrstuhl Arbeitsweise, Projekte, Vorlesungen

- **Deadline Projekt: Dienstag 18. September um 12:00 Uhr**

(Empfehlung: machen Sie es innerhalb der nächsten beiden Wochen, dann sind sie noch "drin" und viel schneller fertig)

■ Zusammenfassung / Auszüge

- Interessant, nur die .h Dateien zu schreiben
- Aber auch schwierig, viele waren sich unsicher, ob das so auch nachher mit ihrem Code zusammenpassen wird

Bei den meisten wahrscheinlich nicht, aber kein Problem, Sie können das Gerüst noch beliebig abändern

Wenn Ihr Gerüst halbwegs Sinn macht und kompiliert, es vernünftige Doku und einen funktionierenden Test gibt und auf Jenkins alles fehlerfrei durchläuft, bekommen Sie schon die vollen 20 Punkte

- Sind die 210 Testinstanzen einsehbar? Klar, auf dem Wiki!

■ Teilnahme

– Noch aktive Teilnehmer*innen : **170**

– An der Evaluation teilgenommen : **144**

101 x Info, 24 x ESE, 14 x Mathe+Physik, 2 x Sonstige

Einige (wenige) haben die EvaSys Mail nicht bekommen

Die BOK-Kurs Leute machen eine separate Evaluation

– Nominierungen für Lehrpreis : **102 ... DANKE!**

– Im Folgenden, eine Zusammenfassung des Feedbacks

Die vollständigen Ergebnisse, inklusive aller Statistiken
und aller Freitextkommentare, finden Sie auf dem Wiki

Evaluationsergebnisse 2/8

■ Art und Weise

- **Viel gelernt:** 64% trifft voll zu, 30% trifft zu, 5% ok SS 2018
70% trifft voll zu, 23% trifft zu, 7% ok SS 2016
- **Verständlich:** 74% trifft voll zu, 19% trifft zu, 6% ok
77% trifft voll zu, 18% trifft zu, 5% ok
- **Niveau:** 51% angemessen, 47% hoch, 1% tief
61% angemessen, 37% hoch, 2% tief
- **Qualität insg:** 67% sehr gut, 28% gut, 6% geht so
73% sehr gut, 24% gut, 3% geht so

■ Übungsblätter

- **Aufwand relativ zu ECTS** ... 1 = sehr hoch, 5 = sehr gering

19% x 1 32% x 2 44% x 3 5% x 4 0% x 5 C++ SS 2018

22% x 1 30% x 2 43% x 3 5% x 4 1% x 5 C++ SS 2016

14% x 1 30% x 2 52% x 3 3% x 4 1% x 5 Ø INF, SS 2017

- Sehr gute Abstimmung zwischen Vorlesung und Übung;
interessante + realitätsnahe Aufgaben; einige (wenige)
fanden den Anstieg des Schwierigkeitsgrades zu stark
- Viel Lob für das ganze Team und die Tutor[^]*: Daniel
Bindemann, Patrick Brosi, Yannick Bühler, Panajiotis Christo-
foridis, Lukas Gebhard, Johanna Götz, Johannes Kalmbach,
Claudius Korzen, Niklas Schnelle, Daniel Tischner

■ Materialien / Online Support

- **Hilfreich:** 71% trifft voll zu, 22% trifft zu, 6% geht so
- **In welcher Form** Veranstaltung vor allem konsumiert:
 - 17% Anwes, **48%** Aufz, 18% teils teils, 14% Folien C++ 2018
 - 21% Anwes, **33%** Aufz, 32% teils teils, 14% Folien C++ 2016
 - 36% Anwes, **16%** Aufz, 18% teils teils, 18% Folien Ø INF, SS 2017
- Aufzeichnungen sehr gut und schnell verfügbar ... Dank an:
Frank Dal-Ri (Technik) & **Alexander Monneret** (Schnitt)
- Viel Lob für den Support auf dem Forum, insbesondere:
Axel Lehmann (Forum, Musterlösungen, Daphne, Alles)
- Viel Lob für Livestreaming und Qualität der Aufzeichnungen

- Lob (überwiegend, siehe Wiki)
 - "Live-Programmierung mit 'Fehlern' ergänzt sehr hilfreich das Verständnis der Folien"
 - "Viel Geduld einzelne Fehler/Fragen genau zu erklären"
 - "Anspruchsvoller Stoff, verständliche motivierend vermittelt"
 - "Themen spannend und unterhaltsam rüber gebracht"
 - "Ich habe selten eine so gut organisierte Vorlesung besucht"
 - "Einzigste Professorin, welche mit dem Niveau von Prof. Schätz aus der Physik mithalten kann"
 - "Sie hat den Tag des Ausscheidens der deutschen Nationalmannschaft nach dem 1. Spieltag korrekt vorhergesagt"

- Kritik (vereinzelt, siehe Wiki)
 - "ECTS Punkte anpassen, 4 ist zu wenig" **Ab SS 2019: 6 ECTS**
 - "Forenantworten manchmal kryptisch"
 - "Die Beleuchtung für die Aufnahmen ist nicht wirklich notwendig und eine Stromverschwendung"
 - "Exkurse am Anfang interessant ... aber nicht selten werden Meinungen der Dozentin mehr oder weniger bewusst als Tatsachen verkauft, was ich nicht immer für angemessen halte"
 - "Die Programmieraufgaben könnten 'knobeliger' sein"
 - "Für ESE-Studierende besser mikrocontroller-orientiertes C"
 - "Studierende, die nicht die Norm erfüllen, mit Nachsicht behandeln und integrieren statt zu isolieren"

- Verbesserungen gg.über **letztem Mal** ... im SS 2016
 - Zu Beginn (V1, Ü1) noch klarer gemacht, was genau die Regelungen und die Anforderungen sind + diverse Maßnahmen zur konsistenteren Punktevergabe (schwierig!)
 - Live-Streaming + noch bessere Aufzeichnungen
 - Ncurses diesmal 100% standardkonform (ohne ANSI codes)
 - Bessere Ü3 – Ü5 (SS 2016: Tetris = interessant aber zu viel) + viele andere neue interessante Aufgaben mit Praxisbezug
 - Mehr Übungsblätter mit automatischen Tests
 - Zusatzaufgaben für die (ansonsten) Unterforderten
 - Editor und Konsole in zwei verschiedenen Fenstern
 - Zahlreiche Verbesserungen der einzelnen Vorlesungen

■ Geplante Verbesserungen für das **SS 2020**

- Die bisherige Qualität halten und weiter auf die vielen verschiedenen Lerntypen und Niveaus eingehen (schwierig!)
- **Noch** klarer machen, was als Plagiat gilt und noch besseren Umgang damit finden, wenn es passiert
- Eigene Seite zum Setup des Linux-Subsystems von Windows
- Mehr Unterstützung für Nicht-Deutsch-Muttersprachler
- Treffen mit Tutor/in eine Woche früher: es wird auch viel dazu genutzt, Verständnisprobleme und Fragen zu klären
- Pünktlicher Schluss machen (nicht leicht, wenn man live programmiert und auf Fragen aus dem Publikum eingeht)

■ Deadline

- Die Deadline für das Projekt ist Dienstag, der
18. September 2018 um 12:00 Uhr UTC+2 (MESZ)
- Sie können es aber auch schon **laaaange** vorher fertig machen, und das wäre auch meine Empfehlung
**Wenn man sich jetzt dran setzt (wo man noch "drin" ist),
kriegt man es gut in einer oder zwei Wochen fertig**
- Für Fragen gibt es das Unterforum "Projekt", kann aber sein, dass es urlaubsbedingt mal etwas länger dauert

■ Punkte

- Wie gesagt, für das Projekt gibt es **100 Punkte**
- Zum Bestehen braucht man davon **50 Punkte ...**
zusätzlich zu mindestens 100 Punkten aus Ü1 – Ü10
- Wenn Sie das **Ü11** ordentlich gemacht haben, haben Sie für das Projekt schon mal **20 Punkte**
- Wenn Sie die Evaluation gemacht haben, haben Sie nochmal **+10 Punkte**
(oder 20 Punkte, die das schlechteste von den Ü1 – Ü10 ersetzen, je nachdem, was für Sie günstiger ist)

Weglassen können Sie das Projekt aber nicht, auch nicht wenn Sie genug Punkte aus den Ü1 – Ü10 haben

■ FAQ zum Projekt

- Kann man das Projekt weglassen, wenn man mehr als 150 Punkte aus den Ü1 – U10 hat?

Nein

- Muss man das Projekt auch dann machen, wenn man in den Ü1 – Ü10 über die Hälfte von 300 Punkten hat?

Ja

- Muss man das Projekt in jedem Fall machen, auch wenn man Xyz studiert und Informatik nur als Nebenfach?

Ja

- Fragen zu Projekt 1 und 2
 - Detaillierte Beschreibung zu den beiden Projekten unter <http://ad-wiki.informatik.uni-freiburg.de/teaching/ProgrammierenCplusplusSS2018/Projekt>
 - **Gibt es dazu gerade noch Fragen?**

■ Motivation

- Der Compiler erzeugt aus Ihrem Quellcode (in der `.cpp` Datei) Maschinencode (in der `.o` Datei)

Der Linker fügt dann nur noch alles zusammen

- Denn kann man sich mit `g++` auch einfach in menschenlesbarer Assemblersprache anschauen:

`g++ -S Simple.cpp`

- Das erzeugt dann eine Datei `Simple.s` ... die kann man dann (trivial) so in Maschinencode übersetzen:

`g++ -c Simple.s`

■ Optimierter Code

- Ohne Optimierung erzeugt `g++` Maschinencode, der **eins zu eins** dem `C/C++` Code entspricht ... siehe Codebeispiel
- Mit Optimierung wird versucht, Code zu erzeugen, der schneller läuft, dabei gibt es verschiedene Stufen

`g++ -O1 ...`

Bringt schon **das meiste**

`g++ -O2 ...`

Noch ein paar Tricks

`g++ -O3 ...`

Fast **alle** Tricks, die der `g++` kennt

- Details zu was da genau optimiert wird: siehe Referenzen
- Bei `-O3` wird die ausführbare Datei **größer**, aber dafür (eventuell) noch **schneller** in der Ausführung

■ Optimierung und Debugging

- Geht nicht gut zusammen, weil bei `-O...` der Maschinencode nicht mehr **1 zu 1** dem Quellcode entspricht
- Man kann dann mit `gdb` nicht mehr so ohne Weiteres Schritt für Schritt durch den Quellcode gehen

`gdb` kann einem dann auch nicht mehr unbedingt sagen, in welcher Zeile im Quellcode ein seg fault aufgetreten ist

- Deswegen zum Debuggen Optimierung **ausschalten**

Wenn ein Bug nur mit Optimierung auftritt, ist das natürlich ein Problem, das ist aber sehr sehr selten

■ Wie wir arbeiten

- Wir lösen praktisch relevante Probleme

Routenplanung auf Google Maps, "Search As You Type",
Semantische Suche, "Question Answering"

- Wir machen unsere Software + Ergebnisse verfügbar

Dazu braucht man gute Software, gute Dokumentation,
gute Benutzerschnittstellen, usw.

- Theorie als Werkzeug, nicht um der Theorie willen

Aber wichtig: ohne theoretisches Verständnis beim Lösen
komplexer Problem nur "Rumgehacke" und "Raterei"

■ Betreuung

- Ähnlich wie in der Vorlesung

Sehr gute Infrastruktur, Systeme und Support

Abgesehen davon sollen und dürfen Sie unabhängig arbeiten und haben viel Freiheit

Gut geeignet für enthusiastische Leute, die gerne an praktischen Problemen arbeiten, so dass am Ende etwas dabei herauskommt, das funktioniert und nützlich ist

(und am besten auch noch schön 😊)

■ Aktuelle Projekte und Demos

- Routenplanung (Teil von Google Maps) [demo](#)
- Visualisierung des weltweiten ÖPNV (Travic) [demo](#)
- Automatisches Malen von ÖPNV-Karten (Loom) [demo](#)
- Automatische Linienverläufe (Pfaedle) [demo](#)
- Interaktive Semantische Suche (Broccoli) [demo](#)
- Large-Scale SPARQL+Text Suche (QLever) [demo](#)
- Question Answering (Aqqu) [demo](#)
- Text extraction from PDF (Icecite) [paper](#)

■ Die nächsten Vorlesungen

– **Information Retrieval...** im WS 18/19

Mit Aufzeichnungen und Live-Tutoraten, da ich ab dem WS 18/19 für 1 Jahr Dekanin der Technischen Fakultät bin

– **Algorithmen und Datenstrukturen** ... im SS 2019

Nach dem anfänglichen Schock des Dekaninnenamtes

– **Programmieren in C++** ... im SS 2020

Hoffentlich wieder in alter Frische

– **B.Sc. / M.Sc. Projekte und Abschlussarbeiten**

Ablauf und Angebote siehe Wiki ... wir sind sehr ausgelastet,
bei Interesse bitte nicht aufgeben und mehrmals fragen !

■ Spezifizierung der Projekte

- Auf dem Wiki:

<http://ad-wiki.informatik.uni-freiburg.de/teaching/ProgrammierenCplusplusSS2018/Projekt>

■ G++ Optimierungsstufen

- Siehe das g++ Manual (958 Seiten)

<https://gcc.gnu.org/onlinedocs/gcc-9.1.0/gcc.pdf>

■ Lehrstuhl

- Webseiten: <http://ad.informatik.uni-freiburg.de>
- Demos: <http://ad.informatik.uni-freiburg.de/demos>