

Praktikum Rechnerarchitektur

Einführung und Tipps Projektphase

Vincent Bode

Lehrstuhl für Rechnerarchitektur und Parallele Systeme
Fakultät für Informatik
Technische Universität München

14. Juni 2021

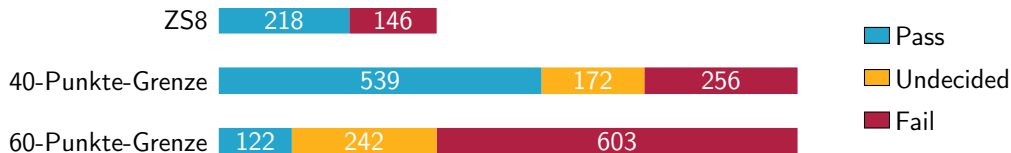
Materialien von Alexis Engelke & Vincent Bode

- ▶ Projektqualifizierung¹

¹ Nicht in Informatik immatrikuliert und noch keine Creditbestätigung? Unbedingt Übungsleitung kontaktieren

► Projektqualifizierung¹

Studis



¹ Nicht in Informatik immatrikuliert und noch keine Creditbestätigung? Unbedingt Übungsleitung kontaktieren

Organisatorisches

- ▶ Übungsbetrieb:
 - ▶ Woche 8: <https://gra.caps.in.tum.de/doc/ZusatztutorienWoche8>
 - ▶ Woche 9: Letzte Tutorübung, ohne Übungsblatt

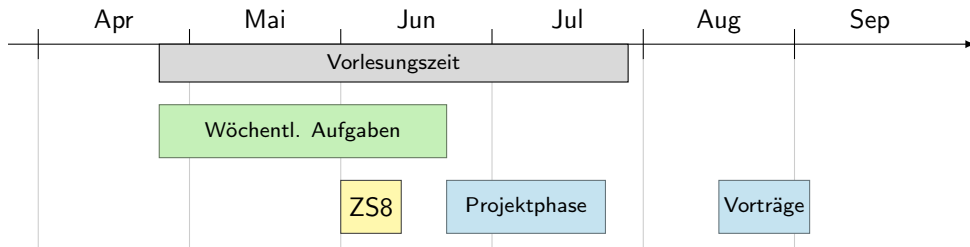
Organisatorisches

- ▶ Übungsbetrieb:
 - ▶ Woche 8: <https://gra.caps.in.tum.de/doc/ZusatztutorienWoche8>
 - ▶ Woche 9: Letzte Tutorübung, ohne Übungsblatt
- ▶ Prüfungsanmeldung auf TUMOnline
 - ▶ Wird von uns gemacht, sobald Prüfung in TUMOnline gefixed
 - ▶ Dann auch individuelle Benachrichtigung zu Bewertungsstatus

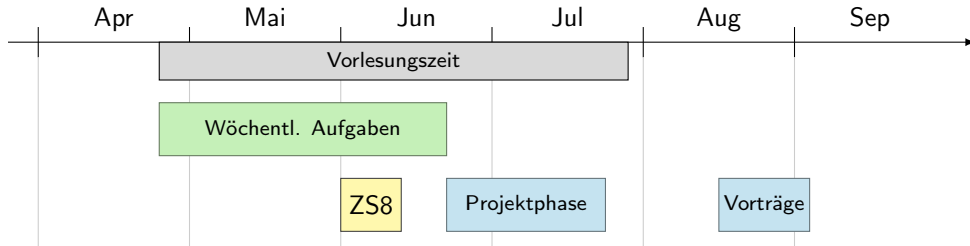
Organisatorisches

- ▶ Übungsbetrieb:
 - ▶ Woche 8: <https://gra.caps.in.tum.de/doc/ZusatztutorienWoche8>
 - ▶ Woche 9: Letzte Tutorübung, ohne Übungsblatt
- ▶ Prüfungsanmeldung auf TUMOnline
 - ▶ Wird von uns gemacht, sobald Prüfung in TUMOnline gefixed
 - ▶ Dann auch individuelle Benachrichtigung zu Bewertungsstatus
- ▶ Anmeldung zu Projektphase
 - ▶ 14.06.2022 (heute) 15:00 – 19.06.2022 23:59
 - ▶ Teamwünsche im Formular eintragen (Matrikelnummer)

Ablauf: Praktikum



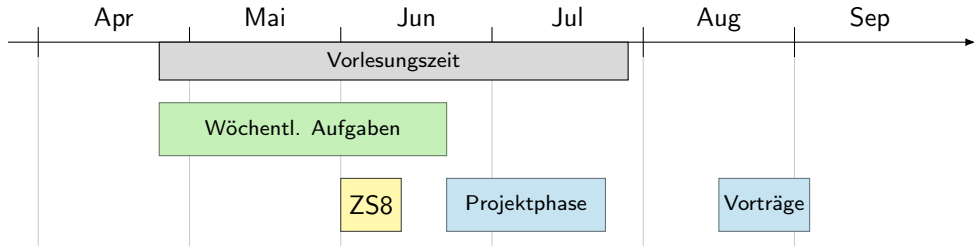
Ablauf: Praktikum



Projektphase

- ▶ Bearbeitung von Implementierung, Ausarbeitung, Folien
- ▶ Bis So. 24. Juli 2022, 23:59 Uhr (CEST)

Ablauf: Praktikum



Projektphase

- ▶ Bearbeitung von Implementierung, Ausarbeitung, Folien
- ▶ Bis So. 24. Juli 2022, 23:59 Uhr (CEST)

Vorträge: Anmeldung und Terminwünsche

- ▶ Nach Gruppenzuteilung
- ▶ Über Webseite (FCFS)
- ▶ Starttermin per E-Mail

Anforderungen

- ▶ Anforderungen = Praktikumsordnung + Aufgabenstellung
 - ▶ **Genau, gründlich, ggf. mehrfach** lesen!
 - ▶ Praktikumsordnung besser kennen → bessere Note

Anforderungen

- ▶ Anforderungen = Praktikumsordnung + Aufgabenstellung
 - ▶ **Genau, gründlich, ggf. mehrfach** lesen!
 - ▶ Praktikumsordnung besser kennen → bessere Note
- ▶ Offen gestellte Aufgabe
 - ▶ (Nicht-)Beispiele: Sortieren, Numerische Quadratur, ...
 - ▶ Meist keine genaueren Vorgaben zur Umsetzung

Anforderungen

- ▶ Anforderungen = Praktikumsordnung + Aufgabenstellung
 - ▶ **Genau, gründlich, ggf. mehrfach** lesen!
 - ▶ Praktikumsordnung besser kennen → bessere Note
- ▶ Offen gestellte Aufgabe
 - ▶ (Nicht-)Beispiele: Sortieren, Numerische Quadratur, ...
 - ▶ Meist keine genaueren Vorgaben zur Umsetzung
- ▶ Bewertung:
 - ▶ Einhaltung Aufgabenstellung und Formalitäten
 - ▶ Technische Korrektheit

Anforderungen II

- ▶ Ausarbeitung
 - ▶ Beschreibung Problemstellung und Lösungs**ansatz**
 - ▶ Evaluierung des Ansatzes und dessen Implementierung

Anforderungen II

- ▶ Ausarbeitung
 - ▶ Beschreibung Problemstellung und Lösungsansatz
 - ▶ Evaluierung des Ansatzes und dessen Implementierung
- ▶ Implementierung
 - ▶ Mittel zum Zweck: notwendig zur Evaluierung
 - ▶ *Assembler-Implementierung “nur” formale Anforderung²*
 - ▶ **Kein Assembler-Projekt** – eher: C-Projekt mit optionalem Assembler-Teil

²Errata: Assembler ist keine Anforderung

Anforderungen II

- ▶ Ausarbeitung
 - ▶ Beschreibung Problemstellung und Lösungsansatz
 - ▶ Evaluierung des Ansatzes und dessen Implementierung
- ▶ Implementierung
 - ▶ Mittel zum Zweck: notwendig zur Evaluierung
 - ▶ *Assembler-Implementierung “nur” formale Anforderung²*
 - ▶ **Kein Assembler-Projekt** – eher: C-Projekt mit optionalem Assembler-Teil
- ▶ Vortrag
 - ▶ Vorstellung Problemstellung, Ansatz, Ergebnisse

²Errata: Assembler ist keine Anforderung

Anforderungen II

- ▶ Ausarbeitung ← *wichtigster Teil*
 - ▶ Beschreibung Problemstellung und Lösungsansatz
 - ▶ Evaluierung des Ansatzes und dessen Implementierung
- ▶ Implementierung
 - ▶ Mittel zum Zweck: notwendig zur Evaluierung
 - ▶ Assembler-Implementierung “nur” formale Anforderung²
 - ▶ **Kein Assembler-Projekt** – eher: C-Projekt mit optionalem Assembler-Teil
- ▶ Vortrag
 - ▶ Vorstellung Problemstellung, Ansatz, Ergebnisse

²Errata: Assembler ist keine Anforderung

Mögliche Vorgehensweise (Implementierung)

<https://tweedback.de/est5>

Mögliche Vorgehensweise (Implementierung)

<https://tweedback.de/est5>

- ▶ Zunächst: einfachen Ansatz in C implementieren
 - ▶ Vollständig funktionsfähiges (einfaches) Rahmenprogramm
 - ▶ Erste Referenzimplementierung zum Testen von Korrektheit und Performanz

Mögliche Vorgehensweise (Implementierung)

<https://tweedback.de/est5>

- ▶ Zunächst: einfachen Ansatz in C implementieren
 - ▶ Vollständig funktionsfähiges (einfaches) Rahmenprogramm
 - ▶ Erste Referenzimplementierung zum Testen von Korrektheit und Performanz
- ▶ Optimierte Ansätze auch erst in C implementieren
 - ▶ Deutlich einfacher zu Debuggen als Assembler

Optimierung

- ▶ Konzeptionelle Optimierungen wichtiger als Mikrooptimierung
 - ▶ Aber: kein unnötig ineffizienter Code

Optimierung

- ▶ Konzeptionelle Optimierungen wichtiger als Mikrooptimierung
 - ▶ Aber: kein unnötig ineffizienter Code
- ▶ Assembler-Impl. muss nicht besser als C-Impl. sein
 - ▶ Wichtiger: Unterschiede (korrekt) **begründen**

Optimierung

- ▶ Konzeptionelle Optimierungen wichtiger als Mikrooptimierung
 - ▶ Aber: kein unnötig ineffizienter Code
- ▶ Assembler-Impl. muss nicht besser als C-Impl. sein
 - ▶ Wichtiger: Unterschiede (korrekt) **begründen**
- ▶ Auch C-Code ist zu optimieren

Implementierung: Sonstiges

► Keine Segmentation Faults

- Eigentlich: kein Undefined Behavior; aber schwer umsetzbar ☹
- Überprüft euren Code mit Analysetools

Implementierung: Sonstiges

- ▶ **Keine Segmentation Faults**
 - ▶ Eigentlich: kein Undefined Behavior; aber schwer umsetzbar ☹
 - ▶ Überprüft euren Code mit Analysetools
- ▶ Keep it Simple: Over-Engineering und unnötige Features vermeiden
 - ▶ Weniger Code \approx weniger Fehler

Implementierung: Sonstiges

- ▶ **Keine Segmentation Faults**
 - ▶ Eigentlich: kein Undefined Behavior; aber schwer umsetzbar ☹
 - ▶ Überprüft euren Code mit Analysetools
- ▶ Keep it Simple: Over-Engineering und unnötige Features vermeiden
 - ▶ Weniger Code \approx weniger Fehler
- ▶ Sinnvolle & hilfreiche Fehlermeldungen

Implementierung: Sonstiges

- ▶ **Keine Segmentation Faults**
 - ▶ Eigentlich: kein Undefined Behavior; aber schwer umsetzbar ☹
 - ▶ Überprüft euren Code mit Analysetools
- ▶ Keep it Simple: Over-Engineering und unnötige Features vermeiden
 - ▶ Weniger Code \approx weniger Fehler
- ▶ Sinnvolle & hilfreiche Fehlermeldungen
- ▶ Git nutzen

Implementierung: Sonstiges

- ▶ **Keine Segmentation Faults**

- ▶ Eigentlich: kein Undefined Behavior; aber schwer umsetzbar ☹
- ▶ Überprüft euren Code mit Analysetools

- ▶ Keep it Simple: Over-Engineering und unnötige Features vermeiden

- ▶ Weniger Code \approx weniger Fehler

- ▶ Sinnvolle & hilfreiche Fehlermeldungen

- ▶ Git nutzen

- ▶ *Read the Error Message*

Implementierung: Sonstiges

- ▶ **Keine Segmentation Faults**
 - ▶ Eigentlich: kein Undefined Behavior; aber schwer umsetzbar ☹
 - ▶ Überprüft euren Code mit Analysetools
- ▶ Keep it Simple: Over-Engineering und unnötige Features vermeiden
 - ▶ Weniger Code \approx weniger Fehler
- ▶ Sinnvolle & hilfreiche Fehlermeldungen
- ▶ Git nutzen
- ▶ *Read the Error Message*

Weitere Details: siehe Handreichung zur Projektphase

Wahl der Inhalte

Ausarbeitung

Vortrag

Wahl der Inhalte

Ausarbeitung

- ▶ Für den Leser schreiben
- ▶ Relevante Information über:
 - ▶ Problemstellung
 - ▶ Lösungsansätze
 - ▶ Evaluierung der Ansätze
- ▶ Ergebnisse analysieren

Vortrag

Wahl der Inhalte

Ausarbeitung

- ▶ Für den Leser schreiben
- ▶ Relevante Information über:
 - ▶ Problemstellung
 - ▶ Lösungsansätze
 - ▶ Evaluierung der Ansätze
- ▶ Ergebnisse analysieren
- ▶ Nicht: was haben wir alles gemacht

Vortrag

Wahl der Inhalte

Ausarbeitung

- ▶ Für den Leser schreiben
- ▶ Relevante Information über:
 - ▶ Problemstellung
 - ▶ Lösungsansätze
 - ▶ Evaluierung der Ansätze
- ▶ Ergebnisse analysieren
- ▶ Nicht: was haben wir alles gemacht

Vortrag

- ▶ Am Zuhörer ausrichten
- ▶ Verständliche Darstellung von:
 - ▶ Problemstellung (Beispiele)
 - ▶ Ansätzen (Grafiken)
 - ▶ Ergebnissen (nur Wichtiges)
- ▶ Auf relevante Inhalte fokussieren

Wahl der Inhalte

Ausarbeitung

- ▶ Für den Leser schreiben
- ▶ Relevante Information über:
 - ▶ Problemstellung
 - ▶ Lösungsansätze
 - ▶ Evaluierung der Ansätze
- ▶ Ergebnisse analysieren
- ▶ Nicht: was haben wir alles gemacht

Vortrag

- ▶ Am Zuhörer ausrichten
- ▶ Verständliche Darstellung von:
 - ▶ Problemstellung (Beispiele)
 - ▶ Ansätzen (Grafiken)
 - ▶ Ergebnissen (nur Wichtiges)
- ▶ Auf relevante Inhalte fokussieren
- ▶ Nicht: 30 Zeilen Asm zeigen

Wahl der Inhalte

Ausarbeitung

- ▶ Für den Leser schreiben
- ▶ Relevante Information über:
 - ▶ Problemstellung
 - ▶ Lösungsansätze
 - ▶ Evaluierung der Ansätze
- ▶ Ergebnisse analysieren
- ▶ Nicht: was haben wir alles gemacht

Vortrag

- ▶ Am Zuhörer ausrichten
- ▶ Verständliche Darstellung von:
 - ▶ Problemstellung (Beispiele)
 - ▶ Ansätzen (Grafiken)
 - ▶ Ergebnissen (nur Wichtiges)
- ▶ Auf relevante Inhalte fokussieren
- ▶ Nicht: 30 Zeilen Asm zeigen

Weitere Details: siehe Handreichung zur Projektphase

Fragen nach dem Vortrag

Fragen nach dem Vortrag

- ▶ Fragen zum Projekt
 - ▶ Wieso, weshalb, warum?
 - ▶ Zu: Performanzergebnissen, Entscheidungen, möglichen Alternativen, aber auch: Details in Implementierung/Ausarbeitung
 - ▶ Jeder Prüfling zu allen Projektteilen

³Errata: Diese Frage ist nicht mehr im Fragenkatalog.

Fragen nach dem Vortrag

- ▶ Fragen zum Projekt
 - ▶ Wieso, weshalb, warum?
 - ▶ Zu: Performanzergebnissen, Entscheidungen, möglichen Alternativen, aber auch: Details in Implementierung/Ausarbeitung
 - ▶ Jeder Prüfling zu allen Projektteilen
- ▶ Fragen zum Inhalt des Praktikums *tendenziell* eher zu Konzepten

³Errata: Diese Frage ist nicht mehr im Fragenkatalog.

Fragen nach dem Vortrag

- ▶ Fragen zum Projekt
 - ▶ Wieso, weshalb, warum?
 - ▶ Zu: Performanzergebnissen, Entscheidungen, möglichen Alternativen, aber auch: Details in Implementierung/Ausarbeitung
 - ▶ Jeder Prüfling zu allen Projektteilen
- ▶ Fragen zum Inhalt des Praktikums *tendenziell* eher zu Konzepten
 - ▶ *“Mit welcher Instruktion lässt sich eine Division durch 3 effizient realisieren?”³*
 - ▶ *“In welchen Registern werden Integer-Parameter übergeben?”*

³Errata: Diese Frage ist nicht mehr im Fragenkatalog.

Fragen nach dem Vortrag

- ▶ Fragen zum Projekt
 - ▶ Wieso, weshalb, warum?
 - ▶ Zu: Performanzergebnissen, Entscheidungen, möglichen Alternativen, aber auch: Details in Implementierung/Ausarbeitung
 - ▶ Jeder Prüfling zu allen Projektteilen
- ▶ Fragen zum Inhalt des Praktikums *tendenziell* eher zu Konzepten
 - ▶ *“Mit welcher Instruktion lässt sich eine Division durch 3 effizient realisieren?”*³
 - ▶ *“In welchen Registern werden Integer-Parameter übergeben?”*
- ▶ **Don't panic!**

³Errata: Diese Frage ist nicht mehr im Fragenkatalog.

Zeitplanung

- ▶ **Frühzeitig anfangen**

Zeitplanung

- ▶ **Frühzeitig anfangen**
- ▶ Genug Zeit für Ausarbeitung und Vortrag einplanen
 - ▶ Zeit für gute Ausarbeitung \approx Zeit für Implementierung

Zeitplanung

- ▶ **Frühzeitig anfangen**
- ▶ Genug Zeit für Ausarbeitung und Vortrag einplanen
 - ▶ Zeit für gute Ausarbeitung \approx Zeit für Implementierung
- ▶ Vor der Deadline: schauen, dass alles ist, wie es soll
 - ▶ Dateinamen, Ausarbeitung, etc.
 - ▶ Sauberer Clone: Implementierung funktioniert auf `1xhalle`

Was tun bei Problemen?

Was tun bei Problemen?

► Fragen zum Praktikumsinhalt

↪ Selbst herausfinden

Was tun bei Problemen?

- ▶ Fragen zum Praktikumsinhalt
- ▶ Programm kompiliert nicht
- ▶ Programm funktioniert nicht (richtig)

- ↪ Selbst herausfinden
- ↪ Fehlermeldung lesen
 - ↪ Debuggen

Was tun bei Problemen?

- ▶ Fragen zum Praktikumsinhalt
 - ▶ Programm kompiliert nicht
 - ▶ Programm funktioniert nicht (richtig)
 - ▶ Unklarheit Praktikumsordnung
 - ▶ Unklarheit Aufgabenstellung
- ~> Selbst herausfinden
 - ~> Fehlermeldung lesen
 - ~> Debuggen
 - ~> Tutor oder öff. Zulip
 - ~> Tutor

Was tun bei Problemen?

- ▶ Fragen zum Praktikumsinhalt
 - ↪ Selbst herausfinden
- ▶ Programm kompiliert nicht
 - ↪ Fehlermeldung lesen
 - ↪ Debuggen
- ▶ Programm funktioniert nicht (richtig)
 - ↪ Tutor oder öff. Zulip
- ▶ Unklarheit Praktikumsordnung
 - ↪ Tutor
- ▶ Unklarheit Aufgabenstellung
 - ↪ Tutor
- ▶ Gruppenmitglied springt ab
 - ▶ Aufgabenverringerung nur nach Anfrage
- ▶ Gruppenmitglied arbeitet nicht/meldet sich nicht
 - ↪ Tutor

Was tun bei Problemen?

- ▶ Fragen zum Praktikumsinhalt
 - ↪ Selbst herausfinden
- ▶ Programm kompiliert nicht
 - ↪ Fehlermeldung lesen
 - ↪ Debuggen
- ▶ Programm funktioniert nicht (richtig)
 - ↪ Tutor oder öff. Zulip
 - ↪ Tutor
 - ↪ Tutor
- ▶ Unklarheit Praktikumsordnung
- ▶ Unklarheit Aufgabenstellung
- ▶ Gruppenmitglied springt ab
 - ▶ Aufgabenverringerung nur nach Anfrage
- ▶ Gruppenmitglied arbeitet nicht/meldet sich nicht
 - ↪ Tutor
- ▶ Ausnahmeanträge
 - ↪ ÜL
- ▶ Alles andere
 - ↪ Erstmal zum Tutor

Fragen

<https://tweedback.de/est5>

Fragen

<https://tweedback.de/est5>

Viel Erfolg und Freude beim Projekt!