

# Praktikum Compilerbau

## Sitzung 11 – Abschlusspräsentation

**Prof. Dr.-Ing. Gregor Snelting**  
**Andreas Zwinkau**

IPD Snelting, Lehrstuhl für Programmierparadigmen



1. Letzte Woche

2. Präsentation

3. Optimier-Wettbewerb

4. Sonstiges

- Was waren die Probleme?
- Hat soweit alles geklappt?
- Ist der Compiler wirklich fertig?

1. Letzte Woche

**2. Präsentation**

3. Optimier-Wettbewerb

4. Sonstiges

# Abschlusspräsentation

Mittwoch 6.1. Raum 010 um 16 Uhr.

- Pro Gruppe ein Vortrag
- Vorträge dauern 20 Minuten
- Vorsicht: Bei Überziehen wird abgewürgt!

- Gruppe stellt sich vor
- Je Gruppe ein Thema detaillierter:
  1. Syntax und Semantik (Gruppe ?)
  2. Optimierungen (Gruppe ?)
  3. Codeerzeugung (Gruppe ?)
    - Kurze Eingliederung des Themas
    - Generelle Ansätze und Technologie vorstellen
    - Eigener Ansatz: Wie? Warum gewählt? Erfahrungen?
- Sonstige Besonderheiten des eigenen Compilers
- Erfahrungen aus Teamarbeit: Zeitaufwand, Arbeitsteilung, Kenntnisgewinn

**Zielpublikum:** Leute die grob Ahnung von Compilerbau haben. Nicht nur Teilnehmer des Praktikums!

- Denkweise: „*Wir stellen unser Produkt dem Auftraggeber vor*“
- Immer von allgemein nach spezifisch (nicht springen oder low level Einstieg)
- Nicht in Details verlieren
- Wenn es wirklich Sinn ergibt 1-2 Details ausbreiten ist ok
- Vorher testen ob die Zeit reicht
- Vortrag auch als PDF mitnehmen (falls der Laptop streikt)
- 15 Folien sollten reichen (→ Detail Falle...)

1. Letzte Woche

2. Präsentation

3. Optimier-Wettbewerb

4. Sonstiges



- Abgabe des Compilers bis Dienstag, 5.1.
- Wir lassen Tests laufen
- Ergebnisse werden nach der Präsentation bekannt gegeben

## 3 Kriterien

- Korrektheit (sehr wichtig)
- Laufzeit des erzeugten Codes (wichtig)
- Laufzeit des Compilers (unwichtig solange sie im Rahmen liegt)

Laufzeitmessungen als Kombination von realer Laufzeit sowie Zählen der ausgeführten x86-Instruktionen.

**Wann:** Bis zum Dienstag 5.1. um 23:59 Uhr (GMT+1)

**Wie:** Email Anhang, Download von Webseite oder vorbeibringen

Die Abgabe enthält:

- Byte- und Sourcecode des Compilers
- Evtl. verwendete jFirm Version mit abgeben
- Lauffähig! (mindestens unter Linux)
- Shellskript / Jar mit Main Klasse o.ä.:  
Aufruf: `<compiler> <MiniJava file>`
- README mit Beispiel Aufruf
- Ausgabe: Eine ausführbare `a.out` oder eine `a.s` Assembler-Datei
- Optional: bis zu 3 eigene Benchmarks

1. Letzte Woche

2. Präsentation

3. Optimier-Wettbewerb

4. Sonstiges

# Feedback! Fragen? Probleme?

- Anmerkungen?
- Probleme?
- Fragen?