

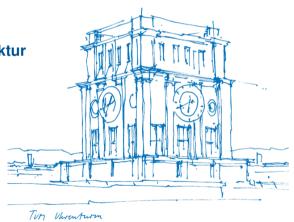
Übung 03: C Toolchain

Grundlagenpraktikum Rechnerarchitektur

#### Niklas Ladurner

School of Computation, Information and Technology Technische Universität München

3. Mai 2024





Keine Garantie für die Richtigkeit der Tutorfolien: Bei Unklarheiten/Unstimmigkeiten haben VL-Folien Recht!

## **GNU Debugger**



- pwndbg Verbesserungen für gdb
- ermöglicht step-by-step debugging von Programmen
- Kompilieren mit -g fügt Debugging-Symbole hinzu: gcc -g main.c
- wichtige Kommandos:
  - □ break label: Setze breakpoint an label
  - ☐ run/continue: Führe Programm bis zum nächsten breakpoint aus
  - next: Geht über Funktionen hinweg (step over)
  - step: Steigt in Funktionen hinein (step into)
  - finish: Steigt wieder heraus

# **Valgrind**



- eignet sich zum Debugging und Profiling (Performanzmessung) von Programmen
- Besteht aus verschiedenen Werkzeugen
- hilft bspw. sehr bei schwer debuggbaren Speicherfehlern, die nur unter bestimmten Umständen auftreten

### **GNU Make**



- Makefiles ermöglichen einen übersichtlicheren Buildvorgang
- Definition verschiedener Targets (zu kompilierenden Dateien) mit komplexen Abhängigkeiten
- make kompiliert Target nur neu, wenn sich Ausgangsdateien geändert haben!



# Fragen?

## Links



- Zulip: "GRA Tutorium Gruppe 20" bzw. "GRA Tutorium Gruppe 22"
- pwndbg
- Dokumentation zu Valgrind
- Dokumentation zu GNU Make



Übung 03: C Toolchain

Grundlagenpraktikum Rechnerarchitektur

#### Niklas Ladurner

School of Computation, Information and Technology Technische Universität München

3. Mai 2024

