

Howto: Anwendung des GDB Debuggers auf ein C++ Programm

Ziel: Finden eines Programmfehlers, wenn der Code zwar kompiliert, das resultierende Programm aber nicht wie gewünscht funktioniert.

Voraussetzung: Das Programm muss mit der Compiler Flag `-g` kompiliert worden sein. (`g++ -g -o PROGRAMMNAME DATEINAME.cc`)

Zunächst verwenden wir DDD, eine graphische Oberfläche des Debuggers GDB.

Befehlsübersicht:

- Starten das Programms mit dem Debugger GDB (im Terminal):
`ddd PROGRAMMNAME`
- Sind wir daran interessiert zu wissen, was das Programm an einer bestimmten Stelle tut, so setzen wir davor einen **Breakpoint**:
Rechtsklick an der betreffenden Stelle im Quellcode (*nicht innerhalb einer Anweisung!*) → **Set Breakpoint**.
- Programm bis zum *Breakpoint* laufen lassen:
Klicke Taste 'F2', dann klicke auf '**Run**'. (Beachte: Verlangt das Programm Argumente, so müssen diese vor 'Run' mitgegeben werden.)
- *Breakpoint* vorübergehend deaktivieren:
Rechtsklick auf *Breakpoint* (ein kleines Stoppschild), dann klicke auf '**Disable Breakpoint**'. (Stoppschild wird grau.)
- *Breakpoint* löschen:
Rechtsklick auf *Breakpoint* (ein kleines Stoppschild), dann klicke auf '**Delete Breakpoint**'. (Stoppschild verschwindet.)
- Nach dem Erreichen des *Breakpoints* wollen wir den Verlauf des Programms zeilenweise weiter verfolgen:
Klicke **Step** (F5): das Programm geht zur nächsten Zeile (auch wenn diese in einer anderen Funktion/Methode liegt)
Klicke **Next** (F6): das Programm geht zur nächsten Zeile, überspringt aber Zeilen innerhalb von weiteren Funktionsaufrufen.
- Wir wollen während des Programmdurchlaufs (step by step) die Größe einer bestimmten Variablen **VAR** verfolgen:
Rechtsklick auf die Variable → klicke **Display VAR**.
Die Grösse der Variable zum aktuellen Zeitpunkt wird jetzt im obrigen Fenster angezeigt (das gepunktete Fenster). Möglicherweise muss man hier noch nach unten scrollen.
Jetzt kann man die Veränderungen der Variable im Einzelnen verfolgen.

Debugging in C++ mit *GDB* ohne DDD:

Befehlsübersicht:

- Programm mit Compiler Flag **-g** kompilieren:
`g++ -g -o PROGRAMMNAME DATEINAME.cc`
- Starte Programm mit Debugger GDB: **gdb PROGRAMMNAME**
- *Breakpoint* im Programm in Zeile **n** setzen:
`break n`
- Programm bis zum *Breakpoint* laufen lassen:
`run`
- Programm bis zum nächsten Breakpoint laufen lassen:
`continue`
- Programm einen 'Schritt' (siehe oben) weiter laufen lassen:
`step`
- Programm **n**-viele 'Schritte' weiter laufen lassen:
`step n`
- Programm soll in die nächste 'Code-Zeile' (siehe oben) springen:
`next`
- *Breakpoint* Nummer **m** vorübergehend deaktivieren:
`disable m`
- *Breakpoint* Nummer **m** löschen:
`delete m`
- Wert einer Variable **var** verfolgen:
`display var`
- Veränderungen einer Variable **var** verfolgen:
`watch var`