

# Programmieren in C

## Debuggen mit gdb

Dr. Adrian Koller, Büro E307, [adrian.koller@hslu.ch](mailto:adrian.koller@hslu.ch)

# Debuggen mit gdb

- Effiziente Analyse und Überprüfung von Programm-Code
- gdb – GNU Debugger
  - sehr mächtiges Werkzeug, häufig das Einzige
- Programm mit `-g` compilieren
  - Debug Information im Executable enthalten

Beispiel:

```
gcc -Wall -g mySourceFile.c -o myProgram
```

# gdb Befehle

- Starten

gdb myProgramm

(gdb)

→ Prompt

- Break Points:

- |                     |  |
|---------------------|--|
| • break main        | setzt einen break point beim Funktionseinstieg |
| • info break        | break points auflisten                         |
| • disable 1         | schaltet break point 1 aus (nicht gelöscht)    |
| • enable 1          | schaltet break point 1 ein                     |
| • delete 1 oder d 1 | löscht break point 1                           |

# gdb Befehle

- (gdb) list oder l      code in der Nähe der aktuellen Ausführung ansehen
- (gdb) run oder r      ausführen (bis zum ersten break point)  
Kommandozeilenargumente möglich
- (gdb) print x      Wert von x ausgeben (falls verfügbar)  
(gdb) print (char)x      x als Character ausgeben  
(gdb) print f(123)      Funktion aufrufen mit «123», Resultat ausgeben

# gdb Befehle

- (gdb) step oder s      Schritt für Schritt Programm ausführen -> geht in Funktionen hinein
- (gdb) next oder n      Schritt für Schritt Programm ausführen -> übersteigt Funktionen
- (gdb) backtrace oder bt  
Führt den Call Stack auf, hilfreich bei  
«Segmentation Fault»