

# Proseminar „C - Grundlagen und Konzepte“

## Übungsaufgaben zum Thema „Debugging mit GDB“

### AUFGABE 1

Schreibe ein kleines Programm, mit drei eigenen Funktionen, die an beliebiger Stelle ausgeführt werden. Wenn möglich, nutze auch Rekursion. Kompiliere das Ganze nun für GDB (das `-g` Flag beachten) und führe es in diesem aus (`gdb Binärdatei`). Setze dir zuerst einen Breakpoint bei einer der Funktionen (`break my_func`). Wenn GDB nun diese Funktion erreicht, verfolge ab diesem Punkt den Programmablauf mit `next`. Versuche auch, in einen weiteren Funktionsaufruf hineinzuspringen, indem du `step` verwendest.

### AUFGABE 2

Führe dein Programm aus der vorherigen Aufgabe noch einmal mit GDB aus. Wenn GDB nun deinen Breakpoint erreicht, schau dir einmal einen `backtrace` an. Mit `up` und `down` kannst du einen Frame nach oben oder unten springen. Lass dir auch einen Variablenwert mit `print` ausgeben.

### AUFGABE 3

Kompiliere folgenden Code (`-g` beachten):

```
1.  #include <stdio.h>
2.
3.  int fac(int x)
4.  {
5.      if (x < 0) {
6.          return 1;
7.      } else {
8.          return x * fac(x - 1);
9.      }
10. }
11.
12. int main(int argc, char *argv[])
13. {
14.     int fac5 = fac(5);
15.     printf("Fakultät von 5: %d\n", fac5);
16.
17.     return 0;
18. }
```

Dieser Code soll uns die Fakultät von 5 ausrechnen. Leider liefert er als Ergebnis immer nur 0. Führe das Programm in GDB aus und versuche, den Fehler zu finden. Da hier auch Rekursion auftritt, solltest du dich dabei mit `step` in die Stackframes vorarbeiten.