

# Praktikum Programmiersprache C - Preprozessor

---

Zur Fehlersuche an Programmen benutzt man sehr häufig Debugger. Viele IDEs stellen sehr komfortable Debugger bereit, unter Linux gibt es den kdbg und ddd, die beide auf dem kommandozeilenorientierten gdb basieren. Mitunter ist der Einsatz von Debuggern jedoch schlecht möglich, etwa bei Echtzeitanforderungen, parallelen Anwendungen oder bei häufiger Rekursion. In solchen Fällen ist es oft sinnvoll, Informationen in eine log-Datei zu schreiben und im Anschluss auszuwerten. Dieses Logging sollte mit bedingter Übersetzung programmiert werden, damit nicht in der Auslieferungsversion noch Testausgaben, die dem Anwender verborgen bleiben sollten, geschrieben oder ausgegeben werden. Macros bieten sich für diese Aufgabe an.

Stellen Sie ein Headerfile zusammen, in dem Macros definiert werden, die bedingt übersetzt werden, wenn das Symbol DEBUG definiert ist. Die Macros sollen Strings, Integerwerte, Zeichen und long-Werte in eine Datei ausgeben, die auch nach einem Programmabsturz möglichst viele Informationen enthält. Vereinbaren Sie in dem Headerfile eine Filevariable mit

```
FILE *pProtocoll;
```

über diese Datei soll das Logging erfolgen. Es sollen Macros der Form

```
#define DEBUG_INT(i)      ...
#define DEBUG_STR(String)...
#define DEBUG_CHR(c)      ...
...
```

bereitgestellt werden. Nach jedem Schreibvorgang ist die Protokolldatei zu schließen, um den Pufferinhalt in der Datei zu speichern. Testen Sie das Headerfile, indem Sie es in ein beliebiges C-Programm einbinden und bei der Übersetzung des C-Programms das Symbol DEBUG setzen.

```
gcc -DDEBUG myprogr.c
```

Werten Sie die erzeugte log-datei aus.

Erzeugen Sie die precompilierte Zwischendatei und suchen Sie darin die expandierten Macros mit und ohne definiertes Symbol DEBUG.

```
gcc -DDEBUG -E myprogr.c > myprog.e
```