



C Programmierkurs

9. Stunde: Der Compiler

Johannes Hayeß & Mirko Seibt

10. Januar 2019

Technische Universität Dresden

Gleitkommazahlen

Schlüsselwörter: `float`, `double`, `long double`

Schlüsselwörter: **float**, **double**, **long double**

Speicherung nach IEEE 754 Standard

pause Zur Darstellung wird statt dem Komma ein Punkt verwendet!

Beispiele: 2.9, 3.E+25, .00007

Man unterscheidet die Typen auch direkt in den Werten.

0.0F (Typ **float**), 0.0 (Typ **double**), 0.0L (Typ **long double**)

Schlüsselwörter: **float**, **double**, **long double**

Speicherung nach IEEE 754 Standard

pause Zur Darstellung wird statt dem Komma ein Punkt verwendet!

Beispiele: 2.9, 3.E+25, .00007

Man unterscheidet die Typen auch direkt in den Werten.

0.0F (Typ **float**), 0.0 (Typ **double**), 0.0L (Typ **long double**)

Gleitkommazahlen repräsentieren **nie** den exakten Wert, sondern nur eine präzisions-, rechnerabhängige Annäherung!

0.2 könnte im Rechner gespeichert sein als

0.20000000000000000111

Compiler-Optionen

```
$ gcc -O<x> main.c
```

- O0 Standardeinstellung
- O1 Aktiviert einfache Optimierungen für den generierten Code
- O2 Beinhaltet alle Optimierungen von -O1 und optimiert noch zusätzlich
- O3 Beinhaltet alle Optimierungen von -O2 und optimiert noch zusätzlich
- Os Optimiert generierten Code auf Größe

O3s Optimierungen sind in manchen Randfällen zu aggressiv und verringern die Leistung des kompilierten Programms.

Ob nun O2 oder O3 verwendet werden sollte gilt also ausgetestet und variiert von Programm zu Programm.

Beachte

Es gibt keine Optimierungsstufe höher als O3, auch wenn der Compiler sich nicht beschwert.

Warnungen gibt der Compiler aus, wenn dieser etwas im Code entdeckt, was wahrscheinlich nicht beabsichtigt ist, aber kein Kompilierfehler ist. Zum Beispiel:

- unbenutzte Variablen
- Automatisches Casten zwischen inkompatiblen Pointer-Typen
- ...

-Wall Aktiviere alle Warnungen

Warnungen gibt der Compiler aus, wenn dieser etwas im Code entdeckt, was wahrscheinlich nicht beabsichtigt ist, aber kein Kompilierfehler ist. Zum Beispiel:

- unbenutzte Variablen
- Automatisches Casten zwischen inkompatiblen Pointer-Typen
- ...

-Wall Aktiviere alle Warnungen

-Wextra Aktiviere noch zusätzliche Warnungen

Warnungen gibt der Compiler aus, wenn dieser etwas im Code entdeckt, was wahrscheinlich nicht beabsichtigt ist, aber kein Kompilierfehler ist. Zum Beispiel:

- unbenutzte Variablen
 - Automatisches Casten zwischen inkompatiblen Pointer-Typen
 - ...
-
- Wall** Aktiviere alle Warnungen
 - Wextra** Aktiviere noch zusätzliche Warnungen
 - Werror** Behandelt Warnungen wie Kompilierfehler

Bestimmte Funktionalitäten werden regelmäßig gebraucht.

Diese werden in sogenannte Bibliotheken ausgelagert. Diese müssen mit dem fertigen Programm dann gelinkt werden.

Beispiel log2:

```
double log_zn = log2(x * x + y * y) / 2;
```

`log2` ist in `math.h` definiert, wird allerdings nicht kompiliert, da die Implementierung nur in der bereits kompilierten Math-Bibliothek zu finden ist.

```
$ gcc -l<lib> main.c
```

In unserem Fall ist die Math-Bibliothek einfach nur **m**.

Also:

```
$ gcc -lm main.c
```

Fragen bisher?

Diese Präsentation ist lizenziert unter der **Creative Commons Attribution-ShareAlike 4.0 International** Lizenz.

