

GCC, erste Schritte, C-Syntax und -Semantik

Ausgabe: 10.10.2013
Abgabe: KW42 (15./18.10.2013)
Punkte: 10+0

1. Aufgabe: Aufwärmen

2 Punkte

Machen Sie sich mit der Arbeitsumgebung unter Linux und den verwendeten Werkzeugen vertraut. Installieren Sie, falls noch nicht geschehen, die GNU-Compilercollection (mit `gcc` und `g++`) und eine für Sie passende IDE. Probieren Sie ruhig verschiedene IDEs aus, um die für Sie günstige herauszufinden.

1. Schreiben Sie ein einfaches C-Programm, welches einen beliebigen Text ausgibt und übersetzen Sie dieses manuell (auf der Kommandozeile). Wie wird das übersetzte Programm gestartet?
2. Was bedeuten die unterschiedlichen Compileroptionen (`-Wall`, `-Werror`, `-O3`)?

Ziel: Erster Kontakt mit C-Compiler

2. Aufgabe: Limits kennen

2 Punkte

Schreiben Sie ein C-Programm, welches die größtmögliche `unsigned int` Zahl auf Ihrem System berechnet und ausgibt. Verwenden Sie hierzu **nicht** die Kenntnis der systemintern verwendeten Bytes (`sizeof`, ...). Nutzen Sie auch nicht die Konstanten/Makros/Funktionen aus `limits.h` oder `float.h` oder anderen Headerdateien!

Ziel: Datentypen, Wertebereiche

3. Aufgabe: Sinus

2 Punkte

Schreiben Sie ein C-Programm, welches für alle Winkelwerte zwischen 0 und 360 Grad in 10-Grad-Schritten den Sinus berechnet und auf drei Stellen nach dem Komma genau ausgibt. Geben Sie jeweils auch den Winkel mit aus, beispielweise Winkel: 90 Grad => Sinus-Funktionswert: 1.000. Schreiben Sie zwei Varianten des Programms: Nutzen Sie einmal eine `for`- und einmal eine `while`-Schleife.

Hinweis: Überlegen Sie, wo Sie die Sinus-Funktion finden können.

Hinweis: Welchen Datentyp nutzen Sie sinnvollerweise für die Iteration? Welchen Datentyp erhält die Sinus-Funktion als Eingabe? Nutzt die Sinus-Funktion als Eingabe Grad oder Bogenmaß (Radian)?

Ziel: Schleifen, Umgang mit Funktionen aus der Standard-Bibliothek

4. Aufgabe: Bit-Operationen

2 Punkte

Schreiben Sie ein C-Programm, welches für eine vorgegebene Zahl zwischen 0 und 255 prüft, ob ein bestimmtes Bit gesetzt ist. Anschliessend soll das Bit gesetzt werden (Wert 1) und danach auf den Wert 0 gesetzt werden. Geben Sie jeweils nach der Operation/Prüfung das Ergebnis aus.

Hinweis: Das Bit und die Zahl können fest codiert werden. Sie können natürlich auch beide Zahlen im Hauptprogramm als Tastatureingabe einlesen.

Ziel: Bit-Operationen

5. Aufgabe: Rätsel?

2 Punkte

Was ist das Ergebnis folgender Programme? Begründen Sie Ihre Antwort.

```
1  #include <stdio.h>
2
3  int main() {
4      unsigned char x = 0;
5      unsigned char y = 3;
6
7
8      if (x=y)
9          printf("x:_%d\n", x);
10     else
11         printf("y:_%d\n", y);
12
13     return 0;
14 }
```

```
1  #include <stdio.h>
2
3  int main() {
4      short x = 0;
5      short y = 3;
6      short z = 2;
7
8      if (x<y<z)
9          printf("sortiert\n");
10     else
11         printf("unsortiert\n");
12
13     return 0;
14 }
```

Ziel: C-Semantik, „Falsche Freunde“ in C