

Programmierkurs C

Josef Strasser-Leitner

Stefan Kirchmair

Thomas Kluckner

Rene Plattner

Christian Posch

Mo. 13-16 Uhr - RR 15 + 19;

Di. - Do. vorm.: 8.30-12.00 - RR15+19;

Nachm.: Di. 13-15 Uhr - RR 15, 20 +21;

Mi. 14-16 Uhr – RR 15, 19 + 20;

Do. 12-14 Uhr – RR 15, 19 + 21;

Fr. 14-16 Uhr – RR 15, 19 + 20;

Programmierkurs C - Wochenübersicht

1. Tag: "Hallo Welt"; Input/Output;
2. Tag: if-Statement; Funktionen; modulo-Funktion
3. Tag: Schleifen; Rekursionen
4. Tag: Arrays u.a. für Zeichenketten
5. Tag: Zeiger, File I/O

Programmierkurs C/C++ - Geschichte

- ~1970 Dennis RITCHIE u. Brian KERNIGHAM in den Bell Labs C entwickelt
- UNIX, TCP/IP wurden in C programmiert
- Mitte der 80er: von Bjarne Stroustrup C++ vorgestellt. C++ ist um OOP (Objektorientierte Programmierung) erweitertes C.
- Alle "Befehle" von C sind auch in C++ enthalten.

Programmierkurs C – ANSI C

- ANSI - American National Standard Institute
- ISO - International Standards Organization
- C89-Standard (ANSI Standard X3.159-1989).
Teil des ANSI C Standards sind die Software Bibliotheken (besser bekannt unter dem Begriff ANSI C standard library).
- C99 – Standard (ISO/IEC 9899:1999)
Revision 1995 u. 1999 – u.a. Ergänzung um weitere Header-Dateien

Programmierkurs C - Organisatorisches

- Linux-Account beantragen:
<http://www2.uibk.ac.at/zid/antrag/index.html>
dann LINUX-Benutzung beantragen
- Kriterien für Zeugnis:
 - Anwesenheitspflicht
 - Programmierübungen erstellen und beim Tutor abgeben
 - Morgendlichen Tests und Abschlusstest Fr. Nachm.

Programmierkurs C

- Konsolen-Fenster in Linux öffnen:

Befehl ausführen:

`xterm (oder xterm -ls)`

- oder:
`konsole`

Programmierkurs C – Linux-Befehle

- `cd` – Wechsel in Home-Verzeichnis
`cd ..` - ‚Ebene‘ höher wechseln
`cd Verzeichnis/Pfad` – Wechseln in ein bestimmtes Verzeichnis
- `pwd` – Angabe des aktuellen Verzeichnisses
- `dir`, `ls` oder `ll` – Auflisten des Inhaltes eines Verzeichnisses
- `mkdir Verzeichnis` - zum Erzeugen eines Verzeichnisses
- `rm` zum Löschen von Dateien
- `man` Befehl – Erläutert den angegebenen Befehl

Programmierkurs C – Dateiname in Linux

- Achtung: In Linux wird zwischen Groß- und Kleinschreibung unterschieden!

Programmierkurs C - Kommentar

- **Kommentare**

// Das ist ein Kommentar

/* Das ist ein mehrzeiliger
Kommentar */

Den mehrzeiligen Kommentar bitte nicht verwenden – siehe
Programmierrichtlinien. Wenn dann eher so:

// Das ist ein
// mehrzeiliger Kommentar

Programmierkurs C - Struktur

- Programmstruktur:

```
int main()    //Funktionsname
{
    //Beginn des Anweisungsblock
    ...      //Deklarationen
    ...      //Anweisungen, jeweils mit ; abgeschlossen
    return 0; //Rückgabewert der Funktion - beendet Funkt.
}           //Ende des Anweisungsblocks der Funkt. main()
```

- Funktion ohne Rückgabewert:

```
void main()
{
    ...
}
```

Programmierkurs C - Ausgabe

- Ausgabe am Bildschirm

```
printf( "Text" );
```

- Neue Zeile

```
printf( "\n" );
```

`printf()` funktioniert nur mit der Standardbibliothek `stdio.h`. Diese Bibliothek wird am Anfang des Programmes eingebunden:

```
#include <stdio.h>
```

Programmierkurs C - Präprozessoranweisungen

- Präprozessoranweisungen beginnen mit # und haben – anders als bei Anweisungen - keinen abschließenden Strichpunkt (;)
- `#include` – bindet andere, separate Dateien in den Quelltext ein.
- `#define`
Die `#define`-Direktive wird auch als Makro oder „symbolische Konstante“ bezeichnet. Die allgemeine Syntax lautet: `#define BEZEICHNER ZU_ERSETZEN`

Der Präprozessor ersetzt jedes Vorkommen des Bezeichner-Makros im Quelltext durch den zu ersetzenden Text `ZU_ERSETZEN`.

Programmierkurs C - Variablen

- Variablen sind Platzhalter für Werte (und belegen Speicherplatz im Arbeitsspeicher)

- Datentypen für Variablen

`int` für Ganzzahlen

`double` für Kommazahlen

`char` für einzelne Zeichen

`unsigned int` - vorzeichenlose, also nur positive Ganzzahlen.

Programmierkurs C - Bezeichner

- Regeln für Bezeichner:

Keine Ziffer am Anfang

Keine Umlaute oder ß

Keine Sonderzeichen wie ? u.a.

Programmierkurs C - key-Words

- Reservierte Wörter (nicht für Bezeichner verwendbar):

auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

Programmierkurs C - Kompilierung

- Kompilierung

`cc HelloWorld.c`

erzeugt a.out als ausführbare Datei

`cc HelloWorld.c -o HelloWorld.out`

HelloWorld.out als ausführbare Datei

Programmierkurs C - Variablen

- Deklaration u. Initialisierung einer Variablen
 - Deklaration:
`int age;`
 - Initialisierung – mit Anfangswert belegen:
`age=0;`
 - Auch beides auf einmal ist möglich:
`int age = 0;`

Programmierkurs C – Ausgabe von Variablen

- `printf(Formatelemente, auszugebende Variablen)`

Formatelemente beginnen immer mit %

Beispiel:

```
printf("%d\t%d\n", fahr, celsius)
```

Programmierkurs C - Formatelemente

%d	als dezimale ganze Zahl ausgeben
%6d	als dezimale ganze Zahl ausgeben, min. 6 Zeichen breit
%f	als Gleitpunktzahl ausgeben
%6f	als Gleitpunktzahl ausgeben, mindestens 6 Zeichen breit
%.2f	als Gleitpunktzahl ausgeben, 2 Zeichen hinter dem Dezimalpunkt als
%6.2f	als Gleitpunktzahl ausgeben, mind. 6 Zeichen breit und 2 Zeichen hinter dem Dezimalpunkt
%c	ein Zeichen ausgeben
%s	einen String ausgeben
-	linksbündig ausgeben

Programmierkurs C - Tastaturwert einlesen

- Einlesen von Werten in eine Variable von der Tastatur:

```
int age=0;
```

```
...
```

```
printf("Wie alt bist Du: ");
```

```
scanf("%d",&age);
```

Programmierkurs C – Formatierte Eingabe

- `int scanf(format, &Variable1, &Variable2, ...)`

`scanf` liest Zeichen von der Standard-Eingabe (=Tastatur), interpretiert sie unter Kontrolle von `format` (in Anführungszeichen) und legt die Eingabe in den Variablen, die mit dem Adressoperator `&` versehen sind, ab.

Programmierkurs C - Operatoren

=	Zuweisungsoperator
==, <, >, <=, >=, !=	Vergleichsoperatoren
&&	Logisches UND
	Logisches ODER

Programmierkurs C - IF-Anweisung

- IF-Statement

```
if ( Bedingung )  
    {  
        ...  
    }  
else  
    {  
        ...  
    }
```

true oder false

true-Zweig

false-Zweig (optional)