

Script generated by TTT

Title: Baumgarten: GBS (31.01.2014)

Date: Fri Jan 31 08:31:50 CET 2014

Duration: 67:01 min

Pages: 18

Grundlagen: Betriebssysteme und Systemsoftware (GBS)

Uwe Baumgarten

kurs-in-der-programmiersprache-c.pdf — Kurs in der Programmiersprache C

TUM-GBS-Fol-2013-ALLE-1-237.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 1 (1 von 237) Auf Seitenbreite einpassen

Vorschaubilder

Technische Universität München

Grundlagen: Betriebssysteme und Systemsoftware (GBS)

Uwe Baumgarten

Menu gbs_c... TUM... kurs-i... Fr. 31. Jan. 08:33

kurs-in-der-programmiersprache-c.pdf — Kurs in der Programmiersprache C

TUM-GBS-Fol-2013-ALLE-1-237.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 37 (37 von 237) Auf Seitenbreite einpassen

Vorschaubilder

Technische Universität München

Inhalte

1. Übersicht	[38]
2. Einführung	[45]
3. Parallele Systeme – Modellierung, Strukturen	[71]
4. Prozess- und Prozessorverwaltung	[115]
5. Speicherverwaltung	[131]
6. Prozesskommunikation	[153]
7. Dateisysteme	[168]
8. Ein-/Ausgabe	[177]
9. Sicherheit in Rechengesystemen	[186]
10. Entwurf von Betriebssystemen	[199]
11. Exkurse Strukturierung, Aufrufe, Beispiele Virtualisierung, Ein-/Ausgabe mit SCSE, C-Programmierung	[212] [233]
12. Zusammenfassung	[]

Menu gbs_c... TUM... kurs-i... Fr. 31. Jan. 08:34

WDH: Systemnahe Programmierung in C

- Darstellung von Daten/Zeichen/Zahlen
- Speicher
 - Speicherbelegung
 - Belegen von Speicher
- Adressierung
 - Zeiger

WDH: Systemnahe Programmierung in C

- Darstellung von Daten/Zeichen/Zahlen
- Speicher
 - Speicherbelegung
 - Belegen von Speicher
- Adressierung
 - Zeiger

bookboon.com

Kurs in der Programmiersprache C

Thomas Theis

```
float a=1.5f, b=2.5f;
float c[10];
float *fp1, *fp2;

c[5]=23.87f;
fp1=&a;
b=*fp1;
*fp1=0.0f;
fp1=&c[5];
printf("%8.4f %8.4f %8.4f\n",a,b,*fp1);

fp2=fp1;
b=*fp2+1;
*fp1+=5;
a=*fp1;
printf("%8.4f %8.4f %8.4f\n",a,b,*fp1);
}
```

eBooks kostenlos heruntergeladen auf bookboon.com

kurs-in-der-programmiersprache-c.pdf — Kurs in der Programmiersprache C

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 85 (85 von 218) Auf Seitenbreite einpassen

Kurs in der Programmiersprache C Zeiger

- Mit "float *fp1, *fp2" werden zwei Zeiger-Variablen vom Typ "Zeiger auf float-Variable" deklariert.
- Mit "fp1=&a" bekommt die Zeiger-Variablen fp1 als Wert die Adresse von a zugewiesen. "&" nennt man den Adressoperator. Man sagt auch: fp1 zeigt auf die Variable a.
- Mit "b=*fp1" bekommt die Variable b den Wert der Variablen, auf die der Zeiger fp1 zeigt, zugewiesen. "*" nennt man im Zusammenhang mit Zeigern den Inhaltsoperator. b hat nun also den Wert von a.
- Mit "fp1=&c[5]" bekommt die Zeiger-Variablen fp1 eine neuen Wert, die Adresse des Feldelementes 5 des Arrays c.
- Mit "fp2=fp1" bekommt die Zeiger-Variablen fp2 den gleichen Wert wie fp1, also die Adresse von c[5] zugewiesen. fp2 zeigt jetzt auch auf c[5].
- Mit "b=*fp2+1" bekommt die Variable b den Wert der Variablen, auf die der Zeiger fp2 zeigt, erhöht um 1, zugewiesen. b hat nun den Wert 24.87.
- Mit "*fp1+=5" wird die Variable, auf die fp1 zeigt (das ist c[5]), um 5 erhöht. c[5] hat nun den Wert 28.87.
- Mit "a=*fp1" bekommt die Variable a den Wert der Variablen, auf die der Zeiger fp1 zeigt, zugewiesen. a hat nun also den Wert von c[5].

Die Ausgabe des Programms:

Fr. 31. Jan. 08:45

kurs-in-der-programmiersprache-c.pdf — Kurs in der Programmiersprache C

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 88 (88 von 218) Auf Seitenbreite einpassen

Zeiger und Felder sind in C sehr eng miteinander verwandt. Der Name eines Feldes ist gleichzeitig ein Zeiger auf das erste Element des Feldes. Zur Übergabe an und Rückgabe von Werten zu Funktionen spielen sie eine wichtige Rolle. Außerdem kann man mit Zeigern auch rechnen, dabei wird der Datentyp des Zeigers immer beachtet.

15.1 Aufbau

Der Aufbau soll an folgendem Programm verdeutlicht werden (Datei U1501.c):

```

/* Datei U1501.c */
#include <stdio.h>
void main(void)
{
    int a[5]={3,5,7,9,11}, *pa;
    printf("\n%i",a[0]);
    printf("\n%i",*a);
    printf("\n%i",a);
    pa=a;
    printf("\n%i",*pa);
    pa=pa+1;
}

```

01001001 01010100
00100000 01010100

Sie sehen hier mehr als Nullen und Einsen?

Wir sehen Ihre Karriere im IT-Trainee-

Fr. 31. Jan. 08:59

kurs-in-der-programmiersprache-c.pdf — Kurs in der Programmiersprache C

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 89 (89 von 218) Auf Seitenbreite einpassen

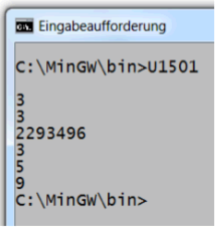
```

/* a=a+1 */
}

```

- Ein Feld von int-Zahlen wird deklariert und initialisiert
- Das erste Element wird wie gewohnt ausgegeben
- Das erste Element wird über "Inhalt von Zeiger" ausgegeben
- Die Adresse des Feldes wird ausgegeben
- Der Zeiger pa bekommt die Adresse des Feldes, anschließend wird der Inhalt von pa ausgegeben, also das erste Feldelement
- Der Zeiger pa wird um eins erhöht. Da er vorher auf das erste Feldelement gezeigt hat, zeigt er nun auf das zweite Feldelement
- Zur Adresse des Feldes wird 3 addiert, damit zeigt pa auf das Element 3 des Feldes.
- Es ist allerdings nicht möglich, einen Zeiger zu verändern, der als Feldname dient.

Die Ausgabe des Programms:



Fr. 31. Jan. 09:00

kurs-in-der-programmiersprache-c.pdf — Kurs in der Programmiersprache C

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Vorherige Nächste 107 (107 von 218) Auf Seitenbreite einpassen

Kurs in der Programmiersprache C Strukturen

17 Strukturen

Strukturen in C dienen dazu, mehrere Variablen mit unterschiedlichen Datentypen zu einer Einheit mit einem einzigen Namen zusammenzufassen.

17.1 Aufbau

Mit der Anweisung:

```

struct adr
{
    char strasse[30];
    int hausnr;
    long int plz;
    char ort[20];
};

```

wird der Struktur-Datentyp adr definiert. Objekte von diesem Typ besitzen vier Komponenten: zwei Zeichenketten und zwei int-Variablen.

Fr. 31. Jan. 09:14

Kurs in der Programmiersprache C

117 (117 von 218)

19 Software

Kurs in der Programmiersprache C Software

In diesem Kapitel werden Download, Installation und Benutzung der Software beschreiben, die für das Entwickeln von C-Programmen unter Windows oder Ubuntu Linux notwendig sind.

19.1 Download und Installation unter Windows

Ein häufig genutzter, frei verfügbarer C-Compiler für Windows ist MinGW. Die Abkürzung steht für Minimalist GNU for Windows. Die Installationsdatei ist zurzeit (Dezember 2011) mingw-get-inst-20111118. Sie hat eine Größe von ca. 0,6 MB und kann von der folgenden Website geladen werden:

<http://sourceforge.net/projects/mingw/files/Installer/mingw-get-inst>

Nach dem Download wird die Installationsdatei aufgerufen. Es bietet sich die Möglichkeit zur Installation verschiedener Compiler. Zumindest der C-Compiler muss installiert werden, weitere werden für diesen Kurs auch nicht benötigt. Nach der Auswahl des Compilers werden die gewünschten Elemente geladen und installiert. Das Standardverzeichnis ist C:\MinGW, die installierte Größe ca. 130 MB. Es wird kein Startmenüeintrag erzeugt (und auch keiner benötigt).

Kurs in der Programmiersprache C

114 (114 von 218)

Dynamische Speicherverwaltung

Kurs in der Programmiersprache C

```

/* Datei U1801.c */

#include <stdio.h>
#include <stdlib.h>
void main(void)
{
    /* Zeiger soll auf Beginn des angeforderten Speicherplatzes zeigen */
    int *zahl;

    /* Speicherplatz-Anforderung für 15 int-Zahlen */
    zahl = (int *) malloc(15 * sizeof(int));

    /* Speicherplatz-Anforderung erfolgreich ? */
    if (zahl != NULL)
        printf("Speicherplatz-Anforderung erfolgreich!\n");
    else
        printf("Speicherplatz-Anforderung nicht erfolgreich!\n");
}

```

Dabei bedeuten:

`malloc(15 * sizeof(int))`: die Reservierung eines Bereiches für 15 int-Zahlen. Rückgabewert von `malloc` ist

WDH: Systemnahe Programmierung in C

- Darstellung von Daten/Zeichen/Zahlen
- Speicher
 - Speicherbelegung
 - Belegen von Speicher
- Adressierung
 - Zeiger

12. Zusammenfassung und Ausblick

- Behandelte Aspekte
 - Modellierung von Prozesse, Petrinetze, Synchronisation
 - Verwaltung von Prozessen, Scheduling
 - Arbeitsspeicherverwaltung, Virtueller Speicher
 - Persistenter Speicher
 - Prozesskommunikation
 - Verwaltung der Geräte
 - Sicherheit in Rechensystemen

12. Zusammenfassung und Ausblick

- Rückblick
 - Hatten wir das Wichtigste?
 - Was fehlte?

- Prüfung

- ...

- Kritisches Feedback

- ...

- **Ausblick**

- Was geschieht in der Zukunft?

C für

BSc

Quelle: [JS12] Kap. 11

Ende

Vielen Dank!

**Auf
Wiedersehen!?!**

kurs-in-der-programmiersprache-c.pdf — Kurs in der Programmiersprache C

TUM-GBS-Fol-2013-ALLE-1-237.pdf — TUM-GBS-Fol-2013.ppt [Kompatibilitätsmodus]

Technische Universität München

TUM

Ende

Vielen Dank!

Auf

```

printf("Speicherplatz-Anforderung erfolgreich!\n");
else
    printf("Speicherplatz-Anforderung nicht erfolgreich!\n");
}

```

Dabei bedeuten:

• malloc(15 * sizeof(int)): die Reservierung eines Bereiches für 15 int-Zahlen. Rückgabewert von malloc ist

Menu gbs_c... TUM... kurs... Fr. 31. Jan. 09:38