

Proseminar - Zeiger in C

Thomas Mechelke

April 2011

Gliederung

1 Grundlagen

- Allgemeines
- Syntax

2 Anwendung

- Zeiger und Funktionen
- Zeiger und Arrays
- Weitere Anwendungen

3 Zusammenfassung

4 Quellen

Gliederung

1 Grundlagen

- Allgemeines
- Syntax

2 Anwendung

- Zeiger und Funktionen
- Zeiger und Arrays
- Weitere Anwendungen

3 Zusammenfassung

4 Quellen

Definition

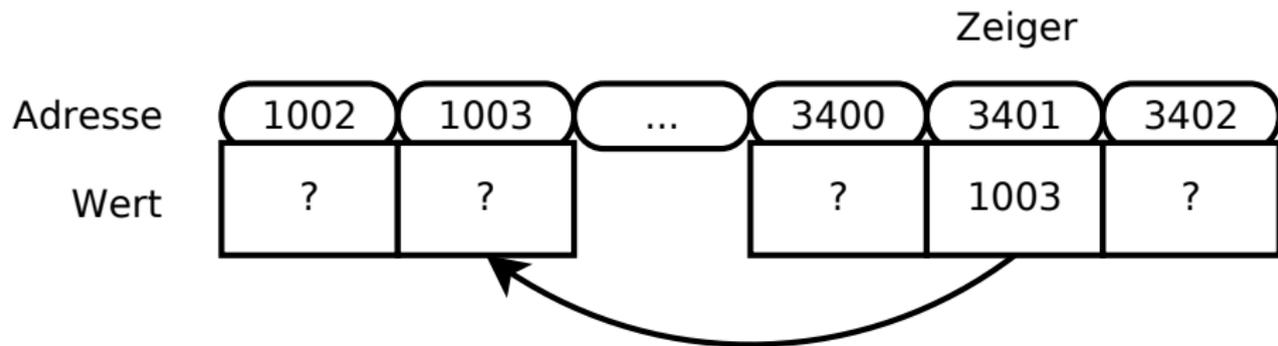
Was ist ein Zeiger?

Ein Zeiger ist eine Variable, die eine Speicheradresse enthält.

Definition

Was ist ein Zeiger?

Ein Zeiger ist eine Variable, die eine Speicheradresse enthält.



Motivation

Warum sollte man sich mit Zeigern auskennen?

- Keine andere Möglichkeit der Implementation
- Performance
- Zeiger hinter den Kulissen
- Halbwissen ist eine Fehlerquelle

1 Grundlagen

- Allgemeines
- Syntax

2 Anwendung

- Zeiger und Funktionen
- Zeiger und Arrays
- Weitere Anwendungen

3 Zusammenfassung

4 Quellen

Syntax

- Deklaration

```
int eineZahl = 0;  
int *einZeiger;
```

Syntax

- Deklaration

```
int eineZahl = 0;  
int *einZeiger;
```

- Adresszuweisung

```
einZeiger = &eineZahl;
```

Syntax

- Deklaration

```
int  eineZahl = 0;  
int  *einZeiger;
```

- Adresszuweisung

```
einZeiger = &eineZahl;
```

- Dereferenzierung

```
*einZeiger = 7;
```

Fehlerquellen

„Writing in C or C++ is like running a chain saw with all the safety guards removed,“ – Bob Gray

Fehlerquellen

„Writing in C or C++ is like running a chain saw with all the safety guards removed,“ – Bob Gray

Fehlerquelle

- Initialisierung vergessen

Fehlerquellen

„Writing in C or C++ is like running a chain saw with all the safety guards removed,“ – Bob Gray

Fehlerquelle

- Initialisierung vergessen

Mögliche Folgen

- Laufzeitfehler
- Ungewollte Änderung von Daten

Fehlerquellen

„Writing in C or C++ is like running a chain saw with all the safety guards removed,“ – Bob Gray

Fehlerquelle

- Initialisierung vergessen

Mögliche Folgen

- Laufzeitfehler
- Ungewollte Änderung von Daten
- Nullzeiger

1 Grundlagen

- Allgemeines
- Syntax

2 Anwendung

- Zeiger und Funktionen
- Zeiger und Arrays
- Weitere Anwendungen

3 Zusammenfassung

4 Quellen

- 1 Grundlagen
 - Allgemeines
 - Syntax
- 2 Anwendung
 - **Zeiger und Funktionen**
 - Zeiger und Arrays
 - Weitere Anwendungen
- 3 Zusammenfassung
- 4 Quellen

Funktionen allgemein

Stackframe

- Parameter
- Lokale Variablen

Funktionen allgemein

Stackframe

- Parameter
- Lokale Variablen
- Call by value

Zeiger als Parameter

- Deklaration

```
void eineFunktion(double *einZeiger){  
    // ...  
}
```

Zeiger als Parameter

- Deklaration

```
void eineFunktion(double *einZeiger){  
    // ...  
}
```

- Call by reference

```
double eineZahl = 0;  
eineFunktion(&eineZahl);
```

Zeiger als Parameter

- Deklaration

```
void eineFunktion(double *einZeiger){  
    // ...  
}
```

- Call by reference

```
double eineZahl = 0;  
eineFunktion(&eineZahl);
```

Beispiele

- Zusätzliche Rückgabewerte
- Sortierfunktion
- Strukturen

Zeiger als Rückgabewerte

- Signatur

```
double *eineFunktion ();
```

Zeiger als Rückgabewerte

- Signatur

```
double *eineFunktion ();
```

Fehlerquelle

- Zeiger wird ungültig

Zeiger als Rückgabewerte

- Signatur

```
double *eineFunktion ();
```

Fehlerquelle

- Zeiger wird ungültig

Thema

- Dynamische Speicherverwaltung

- 1 Grundlagen
 - Allgemeines
 - Syntax
- 2 Anwendung
 - Zeiger und Funktionen
 - **Zeiger und Arrays**
 - Weitere Anwendungen
- 3 Zusammenfassung
- 4 Quellen

Zeiger und Arrays I

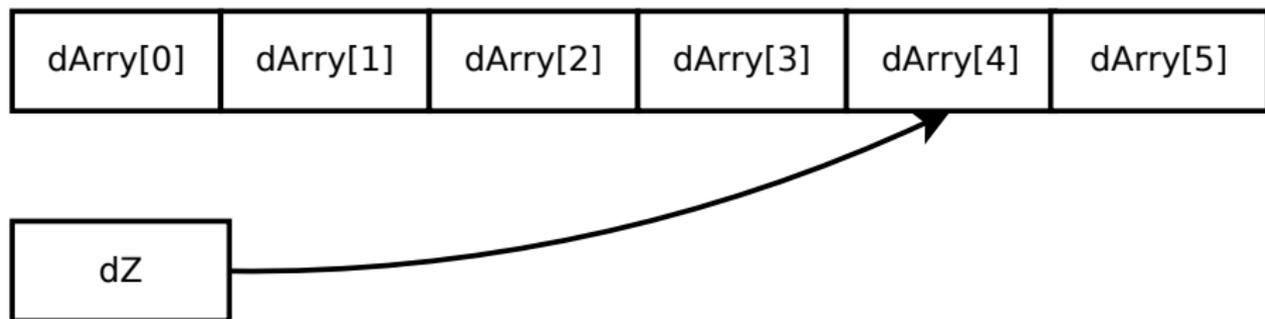
- Syntax

```
double *dZ, dArray[6];  
dZ = &dArray[4];
```

Zeiger und Arrays I

- Syntax

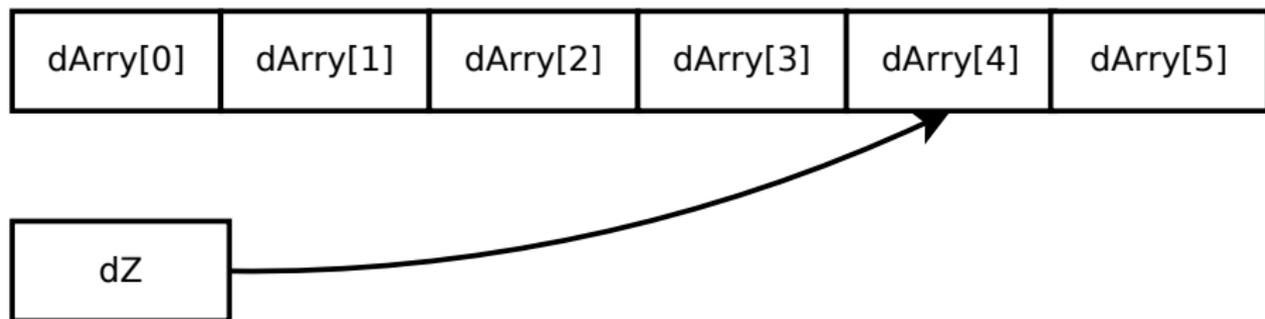
```
double *dZ, dArray [6];  
dZ = &dArray [4];
```



Zeiger und Arrays I

- Syntax

```
double *dZ, dArray [6];  
dZ = &dArray [4];
```



- Elemente nebeneinander
- Zeigerarithmetik

Zeiger und Arrays II

Beispiel

- for-Schleife

Zeiger und Arrays II

Beispiel

- for-Schleife

```
for (dZ = &dArray [0]; dZ <= &dArray [5]; dZ++)  
    *dZ = 0;
```

Zeiger und Arrays II

Beispiel

- for-Schleife

```
for (dZ = &dArray [0]; dZ <= &dArray [5]; dZ++)  
    *dZ = 0;
```

- Möglicherweise bessere Performance

Zeiger und Arrays II

Beispiel

- for-Schleife

```
for (dZ = &dArray [0]; dZ <= &dArray [5]; dZ++)  
    *dZ = 0;
```

- Möglicherweise bessere Performance

Fehlerquelle

- Arraygrenzen überschreiten

Zeichenketten

- char Array
- Nullterminiert

Zeichenketten

- char Array
- Nullterminiert

```
char text[20] = "fast";  
char *nochEinText = "am";  
char undNochEinText[5] = {'e', 'n', 'd', 'e', '\\0'};
```

Zeiger auf Zeiger

Beispiel

- Wie durchläuft man ein Array von Zeichenketten?

Zeiger auf Zeiger

Beispiel

- Wie durchläuft man ein Array von Zeichenketten?

```
char *zKArray [3] = { "ZK1", "ZK2", "ZK3" };
```

```
int i;
```

```
for (i = 0; i < 3; i++)  
    printf("%s\n", zKArray[i]);
```

```
char **zzc;
```

```
for (zzc = zKArray; zzc < &zKArray[3]; zzc++)  
    printf("%s\n", *zzc);
```

1 Grundlagen

- Allgemeines
- Syntax

2 Anwendung

- Zeiger und Funktionen
- Zeiger und Arrays
- Weitere Anwendungen

3 Zusammenfassung

4 Quellen

Weitere Anwendungsbeispiele

- Funktionszeiger
- Datenstrukturen (zB. linked list)

1 Grundlagen

- Allgemeines
- Syntax

2 Anwendung

- Zeiger und Funktionen
- Zeiger und Arrays
- Weitere Anwendungen

3 Zusammenfassung

4 Quellen

Zusammenfassung

Zeiger

- Enthalten Speicheradressen
- Sind essentieller Bestandteil von C
- Können Fehlerquelle sein
- Machen „Call by Reference“ möglich
- Sind (meist) hinter den Kulissen von Arrays

Dies war nur eine sehr knappe Einführung

- Empfehlung: „learning by doing“

- 1 Grundlagen
 - Allgemeines
 - Syntax
- 2 Anwendung
 - Zeiger und Funktionen
 - Zeiger und Arrays
 - Weitere Anwendungen
- 3 Zusammenfassung
- 4 Quellen

Quellen

- „The C Book“ - Banahan, Brady, Doran
http://publications.gbdirect.co.uk/c_book/ [05.05.11]
- „C Wikibook“ <http://de.wikibooks.org/wiki/C-Programmierung>
[05.05.11]
- „C Standart ISO/IEC 9899“
<http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1124.pdf>
[05.05.11]