

Übungen zur Vorlesung: Struktur und Implementierung von Programmiersprachen II

Blatt 3 (Typen, Pointer und Datenrepräsentationen)

Aufgabe 3 (Typumwandlungen bei Pointern)

Geben Sie Beispiele in C jeweils für die Umwandlung eines `int*`-Wertes in einen `int`-Wert und umgekehrt, mittels:

- (a) Conversion
- (b) Coercion (hier sollte der Compiler eine Warnung geben)
- (c) Cast

Aufgabe 4 (Gefahren bei Coercions)

Gegeben sei folgende Funktionsdefinition in C:

```
int sqr(int n) { return n*n; }
```

- (a) Was ist der Wert von `x` nach der Zuweisung:
`int x = 2.5*2.5;`
- (b) Was ist der Wert von `x` nach der Zuweisung:
`int x = sqr(2.5);`
- (c) Wird der C-Compiler bei (a) oder (b) irgendwelche Fehlermeldungen oder Warnungen geben? Warum/warum nicht?

Aufgabe 5 (Alignment)

Gegeben ist folgender `struct` in C:

```
struct S { char aC; double aD;  
           char bC; double bD;  
           char cC; double cD;  
           char dC; double dD; }
```

- (a) Wieviel Speicherplatz (in Bytes) belegt ein Element vom Typ `struct S`, wenn `double`-Werte an durch 4 teilbaren Adressen ausgerichtet sein müssen?
- (b) Wieviel Speicherplatz davon ist Nutzinformation?
- (c) Ordnen Sie die Elemente in `struct S` so um, dass kein Speicherplatz verschwendet wird.
- (d) Warum darf der C-Compiler diese Umordnung nicht automatisch durchführen?

Aufgabe 6 (Mehrdimensionale Arrays)

- (a) Welche Vorteile/Nachteile hat die Verwendung von Vektoren von Vektoren in C gegenüber echten zweidimensionalen Arrays?
- (b) Sie wollen Dreiecksmatrizen speichern. Brauchen Sie dazu Vektoren von Vektoren oder gibt es eine andere Lösung, bei der kein Speicherplatz verschwendet wird? Wenn ja, beschreiben Sie diese bitte.

Besprechung: in der Übung am Donnerstag, 16.11.2006. Schicken Sie bitte Ihre Lösung, am besten in Form einer einzelnen C-Datei mit Kommentaren bis zum 15.11.2006, 12:00 Uhr an herrmann@uni-passau.de mit Subject: SIPS.