

C für (Java-)Programmierer

Eine kleine Einleitung

Was hat C nicht

- Keine Haldenverwaltung
- Keine objektorientierten Sprachelemente
- Keine Namespaces
- Keine „erzwungene“ Typsicherheit
- Keine „Portabilität“
- Keine Zeichenketten
- Keine Freispeicherverwaltung
- Keine Ausnahmenbehandlung

Was C hat

- Zeigerarithmetik
- Maschinennahes Ausführungsmodell
- Goto
- Inline-Assembler
- Präprozessor
- Geschwindigkeit
- Maximale Flexibilität

Zeigerarithmetik

- Von allen Objekten im Speicher läßt sich deren Adresse holen
- Reihungen und Zeiger werden gleich behandelt
- Zeigerarithmetik erlaubt $p[3] == p+3$
- Aber: Zeiger können auch auf „ungültige“ Adressen zeigen

Zeichenketten

- Gibt es NICHT!
- Man verwendet Zeiger auf Zeichen
- Den Speicher für die Zeichen muß man selbst verwalten
- Zeichenketten sind üblicherweise mit einer 0 abgeschlossen
- Deshalb: nur Zeigerarithmetik möglich

Typsystem

- Verbundtypen, Vereinigungstypen und Aufzählungstypen
- Vereinigungstypen sind immer „anonym“
- Für alle Typen lassen sich Zeigertypen bilden, auch für Zeigertypen selbst
- Primitive Typen für vorzeichenbehaftete und vorzeichenlose Ganzzahlen sowie Gleitpunktzahlen existieren
- Funktionstypen für Funktionen

Speicherverwaltung

- Ist NICHT Aufgabe der Sprache, sondern muß von Bibliotheken erledigt werden
- Üblicherweise Verwendung von malloc() und free(),
- Allocation auf dem Keller mittels alloca()

Modifyer

- `static`, `const`, `restrict`, `register`, `auto`, `volatile`, `inline` regeln die Sichtbarkeit UND den Allokationsort von Variablen und Funktionen, steuern mögliche Optimierungen
- Aber: nicht alle Modifyer werden von allen C-Dialekten unterstützt

Dialekte

- Kernighan und Ritchie: der Urdialekt
- C89
- C90
- C99
- GNU-C
- Embedded-C
- ...