C-Grundlagen

Einführung von Tronje Krabbe

Gliederung

- Hintergrund
- Geschichte
- Nutzungsgebiete
- C-Derivate
- Syntax
- Compiler
- Beispielcode

Was ist C?

- C ist eine imperative, kompilierte Programmiersprache
- Entwickelt von Dennis Ritchie zwischen 1969 und 1973
- Am weitesten verbreitete Programmiersprache

Hintergrund

- Multics (Multiplexed Information and Computing Service)
  - Kolaboration von MIT, General Electric und Bell Laboratories (AT&T)
  - Bell Labs verlässt das Projekt
  - Ken Thompson von Bell Labs startet eigenes Projekt

Hintergrund

- Unics
  - Verfügbare Hardware:
    - DEC PDP-7 (18 Bit Architektur mit 8kB RAM)
  - Ähnlich zu Multics
    - Simpler, aufgrund von Hardware
  - Wortspiel: *Uniplexed Information and Computing Services*

Hintergrund

- Original Unix in PDP-7 Assembler geschrieben
- Thompson entscheidet: Unix braucht eigene Systemprogrammiersprache
- Erste Versuche: Fortran, PL/I, Algol 68
  - Hauptspeicher zu klein
- Zweiter Versuch: BCPL-Derivat “B”

Hintergrund

- B
  - Entwickelt von Ken Thompson
  - “C ohne Typen”
  - “BCPL squeezed into 8K bytes of memory and filtered through Thompson's brain”

    [http://cm.bell-labs.com/cm/cs/who/dmr/chist.html]
  - 1969: Entwicklung von B-Compiler
  - Danach: B neu geschrieben in B

Hintergrund

- Compiler erzeugt “Zwischencode”
  - Dieser wird durch einen Parser auf einer Stack-Maschine ausgeführt
- 1970: Thompsons Team bekommt neuen Rechner:
  - PDP-11 mit 24 kB RAM
  - Unix wird in PDP-11 Assembler neu geschrieben

Hintergrund

- New B
  - Dennis Ritchie ändert/erweitert B und nennt es “NB”
  - Compiler übersetzt jetzt direkt in Maschienensprache
- C
  - NB wird in C umbenannt
  - Unix wird in C neugeschrieben
  - Ziel: Portabilität

Geschichte

- 1973: AT&T stell Quellcode von Unix zur Verfügung
- Zusammen mit Unix verbreitet sich auch C sehr schnell
- 1977-79: Diverse Änderungen, Publikation von “The C Programming Language”, a.k.a. “K&R”
  - Informeller Standard von C
- Ab 1983: ANSI X3J11 Komitee standardisiert C

Was soll C sein?

- C wurde zur Erstellung von Systemsoftware entwickelt
- Fokus: Funktionsweise von Rechnern, “Programmierer weiß, was er tut”
  - Schutz gegen Programmierfehler nicht ausgeprägt

Nutzungsgebiete

- Hauptsächlich Systemprogrammierung
- Andere Programmiersprachen:
  - Implementationen von Python, Perl 5 und PHP sind in C geschrieben
- C wurde auch viel für End-User Applikationen benutzt, aber heute werden eher neuere Sprachen genutzt

C-Derivate

- C++
  - Zuerst: “C with Classes”
  - Echte Erweiterung von C zur Objektorientierten Programmierung
- Objective-C
  - Ebenfalls eine echte Erweiterung zur Objektorientierten Programmierung

C-Derivate

- Außerdem:
  - C#, D, Go, Java, JavaScript, Limbo, LPC, Perl, PHP, Python
  - Einfluss syntaktisch, nicht technisch

Syntax

- Ähnlich zu Java
- Enthält:

  a-z, A-Z, _, 0-9

  ~ ! @ # % ^ & * ( ) - + = : ; " ' < > , . ? | / \ { } [ ]

  Whitespace: Leerzeichen, horizontaler Tab, vertikaler Tab, newline, form feed

Compiler

- Auch: Übersetzer, Kompilierer
- Programme, die eingegebene Quellsprachen in Zielsprachen übersetzen
  - z.B.: der C-Compiler übersetzt C-Code in ausführbaren Maschinencode
- Ziel: Optimierung bei gleichbleibender Funktionalität

Sinn des Compilers

- Cross-Platform Programmierung möglich
- Fehlererkennung
- Kompilierte Sprachen sind schneller als interpretierte Sprachen, da Maschinencode besser mit der Maschine arbeitet

Compilerphasen

- Analysephase
  - Prüfung des Codes auf Korrektheit
- Synthesephase
  - Das Programm wird optimiert der Maschinencode generiert

Beispielcode

```
/*
 * C-Beispiel
 */

void main(void)
{
  printf("Dies ist ein Beispielprogramm\n");

} /* Quelle: http://www.lernnetz-sh.de/kmLinux/doc/c-vorlesung/teil3/index.htm */
```

```
void main(void)
 {
   int lower, upper, step;
   float fahr, celsius;

   lower = 0;      /* untere Grenze der Temperaturtabelle */
   upper = 300;    /* obere Grenze */
   step = 20;      /* Schrittweite */
   fahr = lower;

   while (fahr <= upper) {
     celsius = (5.0/9.0) * (fahr-32.0);
     printf("%4.0f %6.1f\n", fahr, celsius);
     fahr = fahr + step;
   }

 } /* Quelle: http://www.lernnetz-sh.de/kmLinux/doc/c-vorlesung/teil3/index.htm */
```

Quellen

- http://cm.bell-labs.com/cm/cs/who/dmr/chist.html
- http://en.wikipedia.org/wiki/C_(programming_language)
- http://wr.informatik.uni-hamburg.de/_media/teaching/sommersemester_2011/cgk11-zuehlke-compiler-ausarbeitung.pdf