Bedeutung und Korrektheit von C Programmen
Vorlesung vom 15.04.2008: Der C-Standard

Christoph Lüth & Lutz Schröder

SS 08

Fahrplan Teil I

- Teil I: C
  - Einführung in den Standard
- Teil II
- Teil III

Fahrplan

- Einführung in C
- Einführung in den C Standard
  - Geschichte
  - Charakter
  - Nomenklatur
  - Gliederung

C Pop-Quiz!

- Schreiben Sie ein Programm `permute`, welches beliebig viele Zeilen aus einer in der Kommandozeile angegeben Datei liest, und diese in zufällig permutierter Reihenfolge wieder ausgibt.
- Beispiel:

```
% cat test
aaa
bbbb
cc
% ./permute test
cc
aaa
bbbb
```


C: Meilensteine

- 1965– 69: BCPL, B; Unix, PDP-7, PDP-11
- 1972: Early C; Unix
- 1976– 79: K& R C
- 1983– 89: ANSI C
- 1990– : ISO C
  - 1985: C++

Geschichte des Standards

- 1978: Kernighan & Ritchie: The C Programming Language
- 1980: zunehmende Verbreitung, neue Architekturen (80x86), neue Dialekte
- 1983: Gründung C Arbeitsgruppe (ANSI)
- 1989 (Dez): Verabschiedung des Standards
- 1990: ISO übernimmt Standard (kleine Änderungen)
- 1999: Überarbeitung des Standards (ISO IEC 9899: 1999)

Nomenklatur

- "Implementation": Compiler und Laufzeitumgebung
- "Implementation-defined": Unspezifiziert, Compiler bestimmt, dokumentiert
  - Bsp: MSB bei signed shift right
- "Unspecified": Verhalten mehrdeutig (aber definiert)
  - Bsp: Reihenfolge der Auswertung der Argumente einer Funktion
- "Undefined": Undefiniertes Verhalten
  - Bsp: Integer overflow
- "Constraint": Einschränkung (syntaktisch/semantisch) der Gültigkeit
- Mehr: §3, "Terms, Definitions, Symbols"

Gliederung

- §3: Terms, Definitions, Symbols — 4 S.
- §4: Conformance — 3 S.
- §5: Environment — 20 S.
  - Übersetzungsumgebung, Laufzeitumgebung
- §6: Language — 165 S.
  - Die Sprache — Lexikalik, Syntax, Semantik
  - Präprozessor
  - Future language directions
- §7: Library — 238 S.
- Anhänge
  - Language syntax summary; Library summary; Sequence Points; Identifiers; Implementation limits; Arithmetic; Warnings; Portability — 112 S.

Eine Sprachkritik

- Philosophie: The Programmer is always right.
  - Wenig Laufzeitprüfungen
  - Kürze vor Klarheit
  - Geschwindigkeit ist (fast) alles
- Schlechte Sprachfeatures:
  - Fall-through bei `switch`, String concatenation, Sichtbarkeit
- Verwirrende Sprachfeatures:
  - Überladene und mehrfach benutzte Symbole (`*`, `()`[1]), Operatorpräzedenzen

[1] 7 Bedeutungen.

Zusammenfassung

- Kurze Reise durch den Standard
  - Begriffsklärung
  - Überblick
- Morgen: erstes Übungsblatt
- Wer kennt `yacc` und LALR(1)-Parsergeneratoren?