Bedeutung und Korrektheit von C Programmen

Vorlesung vom 07.07.2008: Zusammenfassung & Ausblick

Christoph Lüth & Lutz Schröder

SS 08

Der C-Standard: 517 Seiten Spaß

- §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.

In der Vorlesung Nicht Behandelt

- §6.10: Der Präprozessor
- §7: Die Standardbücherei
  - Wichtige Büchereien:

`<assert.h>`	Zusicherung
`<ctype.h>`	Zeichenklassifikation und Konversion
`<errno.h>`	Fehlerwerte
`<math.h>`	mathematische Funktionen
`<stdio.h>`	Ein/Ausgabe
`<stdlib.h>`	Standardbücherei (e.g. `malloc`)
`<time.h>`	Zeit und Datum
`<setjmp.h>`	Nichtlokaler Sprung
`<signal.h>`	Signale (Plattformspezifisch)
`<string.h>`	Zeichenketten (gefährlich und umständlich)

Die Sprache

- 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 (`*`, `()`, Operatorpräzedenzen
- Deklarationen
  - Declaration resembles use $\Longrightarrow$ schwer lesbare Syntax
  - Präsent: Magic Decoder Ring
- Namensräume
  - Labels, Tags von `struct`, `enum` und `union`, Rest
  - Unterschiedliche kinds: Parameter vs. lokale Variablen

Highlights

Umfrage:

- Was ist euer Lieblingsprachfeature (aus dieser Veranstaltung)?
- Was ist euer Lieblingsfeature?

Semantik und Verifikation

- Typisierung, Formale Bedeutung (Semantik)
- Mathematisch präzise Definition der Sprache
- Erlaubt Verifikation durch Beweisregeln:

$$[\![-]\!] : Stmt \to Env \to \Sigma \to \Sigma$$
$$\Gamma \vdash \{P\}p\{Q\} \Longleftrightarrow \forall S.PS \longrightarrow Q([\![p]\!]S)$$

  - Korrektheit der Regeln kann bewiesen werden
  - Nützlich nur mit Computerunterstützung

Ausblick

- Nützung ähnlicher Techniken in Projekt SAMS (DFKI Bremen)

  `http://www.sams-projekt.de`

- Werkzeug mit dieser Technik: Caduceus/Why

  `http://caduceus.lri.fr`, `http://why.lri.fr/`

- Andere Programmiersprachen: Java, C++ ?

- Weiterführende Veranstaltungen: Diplomarbeiten