

## Einführung Proseminar C-Grundlagen

### Was ist C?

C ist eine imperative, kompilierte Programmiersprache, entwickelt um effizient und universell benutzt werden zu können. C wurde zwischen 1969 und 1973 von Dennis Ritchie entwickelt und ist heute eine der am weitesten verbreiteten Programmiersprachen.

### Hintergrund/Geschichte

1964 begann ein Projekt zwischen dem MIT, General Electric und den Bell Laboratories von AT&T namens Multics (Multiplexed Information and Computing Service), ein Betriebssystem für Großrechner.

1969, im Glauben, dass Multics nicht profitabel werden würde, verließen die Bell Laboratories das Projekt. Bereits vorher hatte sich eine inoffizielle Gruppe unter der Leitung von Ken Thompson nach Alternativen umgesehen.

Thompsons Ziel war es, ein komfortables Betriebssystem nach seinen eigenen Vorstellungen zu schaffen, wobei seine Pläne viele der innovativen Eigenschaften von Multics beinhalteten. Während Multics jedoch hauptsächlich in einer Hochsprache (PL/I) geschrieben wurde, war der Nutzen einer solchen, vergleichsweise leicht zu schreibenden Sprache nicht eine von Thompsons Prioritäten. Die Hardware, die Thompson zur Verfügung hatte, war eher spartanisch, sogar für die damalige Zeit. Er arbeitete auf einem DEC PDP-7 mit 8 kB RAM und einer 18 bit Architektur.

Die erste Version von Unix, wie das System, an dem Thompson arbeitete später heißen würde, war in PDP-7 Assembler geschrieben. Tatsächlich hat Thompson nicht einmal auf dem PDP-7 selbst geschrieben, sondern auf einer GE-635-Maschine, welche Papierbänder generierte, die vom PDP-7 gelesen wurden. Als eine erste, primitive Version von Unix fertiggestellt war, wurde die Entwicklung auf der PDP-7-Maschine fortgesetzt.

Wenig später entschied Thompson, dass Unix eine eigene Systemprogrammiersprache bräuchte. Nachdem Versuche mit Fortran, PL/I und Algol 68 sich aufgrund des geringen Speichers seiner Hardware als fruchtlos erwiesen, begann Thompson seine eigene Sprache zu entwickeln. Er nannte sie „B“, wahrscheinlich nach BCPL, der Sprache auf der B basiert.

1969 war dann der erste B-Compiler für Unix fertiggestellt. Dieser erzeugte jedoch keinen Maschinencode, sondern eine Art „Zwischencode“, welcher dann durch einen Parser auf einer Stack-Maschine ausgeführt wurde. Thompson begann, B in sich selbst neu zu schreiben, stieß aber immer wieder auf Probleme mit dem Speicher seines PDP-7.

Ein Jahr später, da das Unix-Projekt vielversprechend erschien, war Thompsons Team in der Lage, einen neuen Rechner zu erwerben: den DEC PDP-11, mit 24 kB Hauptspeicher. Natürlich wurde Unix in PDP-11 Assembler neu geschrieben.

Da B auf dem PDP-11 einige Features nicht nutzen konnte, entschied sich Dennis Ritchie, ein Mitglied von Thompsons Team, B zu verbessern; er nannte es „NB“, für „New B“, und der Compiler übersetzte jetzt den Code direkt in Maschinensprache.

Bis 1973 wird NB alphabetisch sinnvoll in „C“ umbenannt, und Unix in C neu geschrieben. Das Ziel hierbei war Portabilität, da für unterschiedliche Architekturen jetzt nicht das gesamte System, sondern nur ein C-Compiler in Assembler neu geschrieben werden musste.

Im selben Jahr stellte AT&T den Quellcode von Unix, und damit C, Universitäten und Behörden zur Verfügung, was die Verbreitung von beiden deutlich beeinflusste. Bis 1979 wurden noch diverse Änderungen vorgenommen, bis die erste Dokumentation von C erschien, „The C Programming Language“, von Brian Kernighan und Dennis Ritchie (daher auch „K&R“ genannt). Dies diente als informeller Standard, da bisher keine festen Standards für C festgelegt wurden.

1983 wurde C dann offiziell vom ANSI X3J11 Komitee, welches eigens dafür gegründet wurde, erstmals standardisiert.

**Philosophie: „Was soll C sein?“**

C wurde hauptsächlich zur Entwicklung von Systemsoftware entwickelt. Was die Benutzung von C angeht, so wird davon ausgegangen, dass der Programmierer weiß, was er tut; jemand, der bereits intensiv mit Assembler gearbeitet hat, benötigt keine Sprache, die beispielsweise nicht mehr genutzten Speicher leert, etc.

**Nutzungsgebiete**

Wie bereits erwähnt, wird C größtenteils zur Systemprogrammierung genutzt. Auch sind diverse andere Programmiersprachen in C geschrieben, beispielsweise Python, Perl 5 und PHP. Des Weiteren wurde C auch für End-Nutzer Applikationen eingesetzt, was heutzutage aber eher nicht mehr gebräuchlich ist.

**C-Derivate**

Zahlreiche Programmiersprachen basieren auf C. C++ und Objective-C beispielsweise sind Erweiterungen von C, die Objektorientierte Programmierung ermöglichen; beide sind echte Erweiterungen von C, d.h. diverse C-Programme sind auch Objective-C oder C++ Programme. Viele andere Sprachen sind lediglich syntaktisch von C beeinflusst, z.B. D, Go, Java, JavaScript, Limbo, LPC, Perl, PHP und Python.

**Compiler**

C ist eine kompilierte Programmiersprache, d.h. sein Quellcode wird von einem Programm, dem Compiler, in von der Maschine ausführbaren Maschinencode übersetzt.

Allgemein ist das Ziel von Compilern, eine Quellsprache in eine Zielsprache zu kompilieren, und damit bei gleichbleibender Funktion, eine Art Optimierung vorzunehmen.

Dies ermöglicht unter anderem erleichterte Cross-Platform Programmierung, da C-Code auf einer Maschine geschrieben, und auf einer anderen Maschine kompiliert und dann ohne Probleme genutzt werden kann.

Der Compiler kompiliert in zwei Phasen: zuerst wird der Code auf Korrektheit überprüft, was evtl. auch bei Fehlerentdeckung hilft. Danach wird der Maschinencode generiert.

**Quellen**

<http://cm.bell-labs.com/cm/cs/who/dmr/chist.html>

Ein interessantes Paper, von Dennis Ritchie selbst geschrieben.

[http://wr.informatik.uni-hamburg.de/\\_media/teaching/sommersemester\\_2011/cgk11-zuehlke-compiler-ausarbeitung.pdf](http://wr.informatik.uni-hamburg.de/_media/teaching/sommersemester_2011/cgk11-zuehlke-compiler-ausarbeitung.pdf)

[http://en.wikipedia.org/wiki/C\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/C_%28programming_language%29)