



# **Die andere Seite des Programmierens**

Uwe Berger



Programmierer/Informatiker sind doch  
eigentlich abstrakte Künstler, oder...?



# Inhalt

- Motivation
- Klassisches
- Magisches
- Sinnfreies
- Meisterhaftes



## Motivation

Klassisches

Magisches

Sinnfreies

Meisterhaftes



# Computerprogramme

- sollen Aufgaben schneller, genauer, effizienter und kostengünstiger erledigen, als der Mensch
- sollen möglichst wartbar und wiederverwendbar sein
- man braucht dazu heutzutage:
  - Programmiersprachen
  - Algorithmen
  - ... und den *Programmierer*



Was machen *Programmierer*, wenn sie nicht gerade die Probleme der Welt lösen?



Motivation

**Klassisches**

Magisches

Sinnfreies

Meisterhaftes



# Die Magie des „Hello, world!“

- die wohl am häufigsten gelöste Programmieraufgabe
- „Hello, world!“, Brian Kernighan; internes C-Programmierhandbuch der Bell Laboratories; 1974
- Wozu:
  - Aufbau eines Programmes in einer Programmiersprache verdeutlichen
  - erfolgreiche Installation einer Umgebung testen
  - Hardware testen
- z.B.: <http://www.roesler-ac.de/wolfram/hello.htm>



# „Hello, world!“

- Der Normalo:

```
#include <stdio.h>
int main(void)
{
    printf("Hello, world!\n");
    return 0;
}
```

- Der Kürzeste: HQ9+

H

- Mein allfälliger Favorit:

```
label .l -text "Hello, world!"
pack .l
```

*Live-Demo*



# „Hello, world!“

- Mein derzeitiger Liebling:

```
#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 12000000UL

int main( void )
{
    DDRB = (1 << PB1);

    while( 1 ) {
        PORTB ^= (1 << PB1);
        _delay_ms(500);
    }
    return 0;
}
```



## „99 bottles of beer“

- Titel eines Trinkliedes...:  
`99 bottles of beer on the wall, 99 bottles of beer.  
Take one down and pass it around, 98 bottles of beer on the wall.  
  
98 bottles of beer on the wall, 98 bottles of beer.  
Take one down and pass it around, 97 bottles of beer on the wall.  
...  
No more bottles of beer on the wall, no more bottles of beer.  
Go to the store and buy some more, 99 bottles of beer on the wall.`
- Das zweite Programm nach „Hello World“: Wie programmiert man eine Schleife?
- 1253 Variationen (Stand 02.03.2009):
  - <http://www.99-bottles-of-beer.net/>
- auch als Kernelmodul implementiert (/dev/bottles)



# 99 bottles of beer

- Der Klassiker (Basic):

```
10 REM BASIC Version of 99 Bottles of beer
20 FOR X=100 TO 1 STEP -1
30 PRINT X;"Bottle(s) of beer on the wall,";X;"bottle(s) of beer"
40 PRINT "Take one down and pass it around,"
50 PRINT X-1;"bottle(s) of beer on the wall"
60 NEXT
```

- HQ9+:  
9

- Beispiele: 2x Tcl *Live-Demo*



Motivation

Klassisches

**Magisches**

Sinnfreies

Meisterhaftes



# Quines

- Programme, die ihren eigenen Quelltext wieder ausgeben
- Willard Van Orman Quine (1908-2000); Philosoph und Logiker
- Problem: Maschinencode ohne Quelltextbezug
- Echte Quines:
  - dürfen nicht auf externe Quellen zugreifen
  - sollten mit dem Standardsprachumfang auskommen



# Quines

- Quines mit Interpretersprachen:
  - theoretisch einfacher (Laufzeitumgebung)
  - praktisch doch nicht (Sicherheitsmechanismen)
- einfachstes Quine ist ein leeres Programm (zulässig z.B. in bash, Tcl, Perl, Python)
- <http://www.madore.org/~david/computers/quine.html>
- <http://www.nyx.net/~gthompso/quine.htm>



# Quines

- Klassiker in C:

```
main(){char *c="main(){char *c=%c%s%c;printf(c,34,c,34);}";printf(c,34,c,34);}
```

- Der Kurze in HQ9+

Q

- Und natürlich Tcl:

```
set l {set l {L};regsub L $l $l p;puts $p};regsub L $l $l p;puts $p
```

*Live-Demo*



# Polyglott

- *polyglott*: mehrsprachig
- Quelltext, der in mehr als einer Programmiersprache gültig ist
- Methoden:
  - gleich lautende Befehle
  - geschickte Verwendung von Sprungbefehlen sowie exit/return etc.
  - Präprozessoranweisungen
  - unterschiedliche Bedeutung der verschiedenen Kommentarzeichen
- <http://www.nyx.net/~gthompso/poly/polyglot.htm>



# Polyglott

- (mein) Beispiel für C und Tcl:

```
#define VERSION my_polyglott.c; Uwe Berger; 2009
#define DESCRIP Polyglott fuer C und Tcl
#include <stdio.h>
#define set char*f =
#define F
set F "CLT2009: Die andere Seite des Programmierens...\n";
#define proc main(){
proc /* {a b} {
puts -nonewline $a
#*/printf("%s",f);}
/* $F */
```

```
> cc my_polyglott.c
> ./a.out
```

```
> tclsh my_polyglott.c
```

*Live-Demo*



# Self Modifying Code

- Selbstmodifizierender Code
- Begriff: „Freier Rechenplan“, Plankalkül; Konrad Zuse
- auf Maschinencode-Ebene, also meist nicht portierbar
- Wozu:
  - Ressourcenoptimierung
  - Kopierschutz
  - Erschweren von Reverse Engineering
- Beispiel: diverse alte Videospiele
- Esoterische Sprachen, z.B. Befunge



???

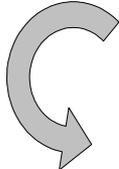
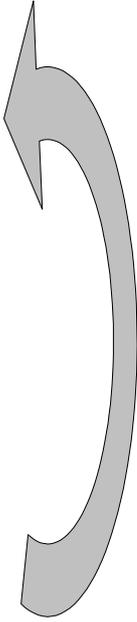
$$M = \left\{ c \in \mathbb{C} ; \lim_{n \rightarrow \infty} |f^n(0)| < \infty \right\}$$



???

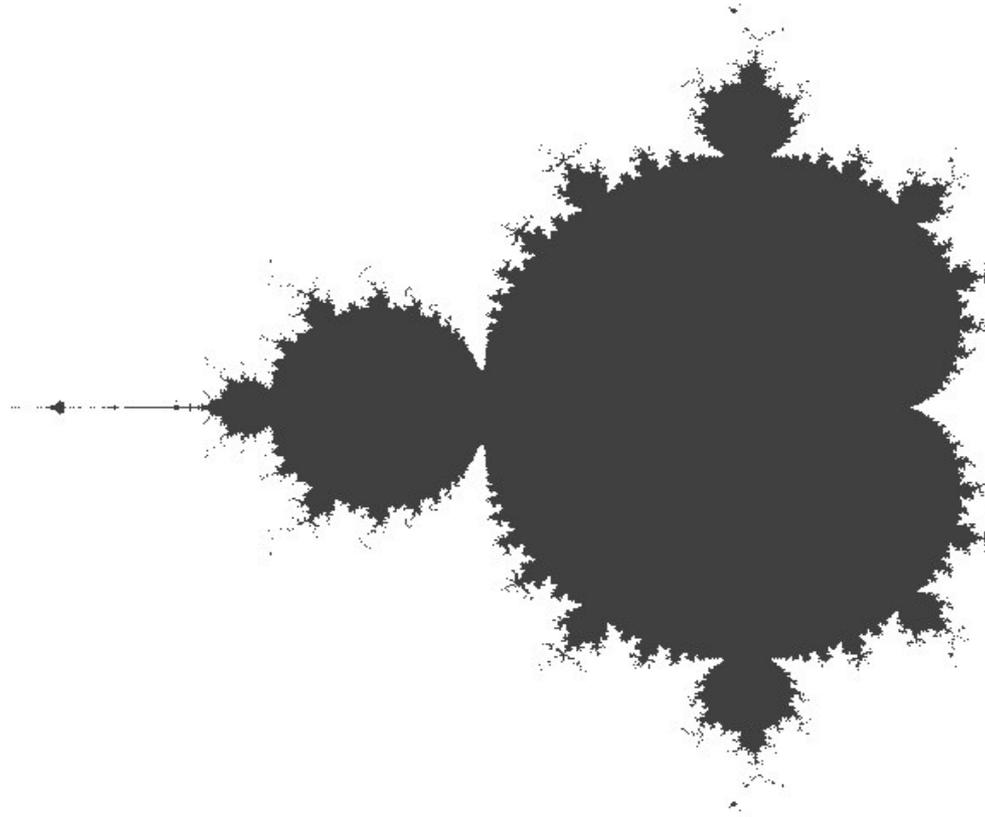
$$x_{n+1} = x_n^2 - y_n^2 + c_x$$

$$y_{n+1} = 2x_n y_n + c_y$$


$$|z_{n+1}|^2 \leq x_{n+1}^2 + y_{n+1}^2$$



!!!





# Fraktale

- *fractus*: gebrochen; *frangere*: in Stücke zerbrechen
- Gebilde mit Skaleninvarianz bzw. Selbstähnlichkeit
- Benoit Mandelbrot, 1975 (Waclaw Sierpinski, Gaston Maurice Julia etc.)
- Digitale Kunst, Fraktalkunst, Chaostheorie
- sehr viele Beispiele in der Natur
- Anwendung z.B.: Generierung von Strukturen und künstlichen Landschaften und ...Performance-Tests
- zum Nachlesen: Lindenmayer- bzw. L-Systeme (Produktionsregeln, Ersetzungssystem); z.B. LOGO, Kturtle

Live-Demo



Motivation

Klassisches

Magisches

**Sinnfreies**

Meisterhaftes



# Programmiersprachen

- Bindeglied zwischen Mensch und Computer, der nur Maschinencode versteht
- dienen zur Darstellung (Notation) von Computerprogrammen
- Gratwanderung zwischen Universalität und Anwendungsorientierung einer Programmiersprache
- haben manchmal auch nur theoretischen Charakter
- weit über 8000 Programmiersprachen
- <http://hopl.murdoch.edu.au/>



# Esoterische Programmiersprachen

- haben eigentlich nichts mit Esoterik zu tun, ...bis auf: Synonym für „Mystik“ oder „spirituelles Wissen“
- verfolgen meist unkonventionelle Konzepte
- „Kategorien“:
  - Minimalistisch (z.B. Brainfuck)
  - „konzeptionell ungewöhnlich“ (z.B. Befunge, Unlambda)
  - Thematisierend (z.B. Shakespeare, Chef)
  - „unerwartet anders/verkomplizierend“ (z.B. INTERCAL)
  - Scherzhaft (z.B. HQ9+, Ook!)



# Esoterische Programmiersprachen

- viele Sprachen sind nur Dialekte bzw. haben ähnliche Konzepte
- [http://esoteric.voxelperfect.net/wiki/Language\\_list](http://esoteric.voxelperfect.net/wiki/Language_list)



## HQ9+

- Cliff L. Biffle; 2000-2001
- Löst die Probleme eines Programmieranfängers
- Name ist vollständige Syntaxzusammenfassung:
  - H** „Hello World!“
  - Q** Quine
  - 9** ... das Trinklied...
  - +** Zähler erhöhen
- <http://www.cliff.biffle.org/esoterica/hq9plus.html>
- <http://www.almnet.de/esolang/hq9plus.php>



# Brainfuck

- Urban Müller, 1993
- die esoterische Programmiersprachen schlechthin...
- Implementierung einer möglichst einfachen „turing-vollständigen“ Sprache
- Turingmaschine:
  - Speicherband, welches vor- und zurückspulen kann
  - ein Schreib- und Lesekopf
  - Programm beschreibt die Bewegung des Speicherbandes und die Aktion des Kopfes



# Brainfuck

- Syntax

>	nächste Position	<i>++p;</i>
<	vorherige Position	<i>--p;</i>
+	Inhalt der aktuellen Position erhöhen	<i>++*p;</i>
-	Inhalt der aktuellen Position erniedrigen	<i>--*p;</i>
.	Inhalt der aktuelle Position ausgeben	<i>putchar(*p);</i>
,	Einlesen und an aktuelle Position schreiben	<i>*p=getchar();</i>
[	Schleifenbeginn	<i>while (*p) {</i>
]	Schleifenende	<i>}</i>



# Brainfuck

- das Programm beginnt immer an Speicherstelle 0
- (triviale) Beispiele:
  - $p[1]=p[1]+p[0]$       `[>+<-]>.`
  - $p[1]=p[0]*2$       `[>+++<-]>.`
- Interpreter für Linux z.B.:
  - bf
  - Beef
- <http://cydathria.com/bf/>
- <http://koti.mbnet.fi/villes/php/bf.php>



# Befunge

- Chris Pressey
- stackorientiert, FORTH-ähnlich
- zweidimensionaler Quelltext! (80x24 Zeichen)
- Befehle zur Modifikation des Quelltextes enthalten!
- <http://quadium.net/funge/>
- Wasabi - eine Befunge-IDE (Java):  
[http://sourceforge.net/project/showfiles.php?group\\_id=203888](http://sourceforge.net/project/showfiles.php?group_id=203888)

*Live-Demo*



# INTERCAL

- Donald R. Woods und James M. Lyon; 1972
- Angeblich die erste esoterische Programmiersprache
- Ziele: Programmieren so schwer wie möglich machen; Quellcode undurchschaubar gestalten. Mittel z.B.:
  - Umkehrung des GOTO in COME FROM
  - Schlüsselwort PLEASE entscheidet über die Laune des Compilers
  - usw.
- <http://www.catb.org/~esr/intercal/>



# Chef

- David Morgan-Ma
- Programmieren mit Kochrezepten
- Sprachelemente:
  - Zutatenliste (Variablen mit ihrem Inhalt)
  - Küchengefäße, -orte (Speicherorte)
  - Kochtätigkeit (Rechenoperation)
  - <http://www.dangermouse.net/esoteric/chef.html>
- Shakespeare, .Gertrude usw. verfolgen ähnliche Konzepte (oder umgekehrt?!)



# Chef „Hello World!“

Hello World Souffle.

Ingredients.

72 g haricot beans

101 eggs

108 g lard

111 cups oil

32 zucchinis

119 ml water

114 g red salmon

100 g dijon mustard

33 potatoes

Method.

Put potatoes into the mixing bowl. Put dijon mustard into the mixing bowl. Put lard into the mixing bowl. Put red salmon into the mixing bowl. Put oil into the mixing bowl. Put water into the mixing bowl. Put zucchinis into the mixing bowl. Put oil into the mixing bowl. Put lard into the mixing bowl. Put lard into the mixing bowl. Put eggs into the mixing bowl. Put haricot beans into the mixing bowl. Liquefy contents of the mixing bowl. Pour contents of the mixing bowl into the baking dish.

Serves 1.



Motivation

Klassisches

Magisches

Sinnfreies

**Meisterhaftes**



## „Merkblatt für unlesbaren Quelltext“

- Der Quelltext sollte erst auf den 2. Blick unwartbar sein!
- Nie das Wichtige, sondern nur das Triviale kommentieren und das am besten in Isländisch!
- Variablen- und Funktionsnamen sollten mit Bedacht gewählt werden!
- Quelltextformatierung ist nur etwas für Warmduscher!
- Compiler-Entwickler haben nicht umsonst Präprozessoren und den GOTO-Befehl erfunden!
- Nicht jede Quellcodepassage muss auch eine Funktion haben!



# IOCCC

- The International Obfuscated C Code Contest
- *to obfuscate*: von lat. *obfuscare*: verdunkeln
- Initiatoren: Landon Curt Noll, Larry Bassel; seit 1984
- <http://www.ioccc.org>
- Jährlich stattfindender Wettbewerb mit jeweils variierenden Regeln
- Grundziel ist aber immer: möglichst unverständlicher Quelltext
- diverse Ableger: Perl, Ruby, Erlang



# IOCCC

- typisch:
  - originelles Layout des Quelltextes
  - Extensive Benutzung des Präprozessors
  - selbstmodifizierender Programmcode
- sind eigentlich nur als Gesamtkunstwerk zu betrachten, deshalb im Editor...

*Live-Demo*



# The Underhanded C Contest

- „Die dunkle Seite des Programmierens“
- underhand: hinterhältig
- Ziel: ein möglichst harmlos aussehendes Programm, welches aber höchst schädlich sein sollte...
- <http://underhanded.xcott.com/>
- Beispiele: ... nein danke, nicht auf meinem Rechner!



# Achtung, Werbung...

- „Coding for Fun“; G. Wolmeringer; Verlag: Galileo Computing; 2008; (ISBN 978-3-8362-1116-1)



*Danke für die Aufmerksamkeit!*