

Programmierung in C



Programmentwicklungszyklus

Editieren-Compilieren-Linken-Ausführen-Debugging

Syntaxdefinition mittels BNF/EBNF/ABNF

Quelltext Beispiele

Literatur/Informationsquellen

- Harbison/Steele: C: A Reference Manual. Prentice Hall 1995
- Isernhagen, Rolf/Hartmut Helmke: Softwaretechnik in C und C++. Hanser Verlag 2004
- Krüger, Guido: Programmieren in C. Grundlagen, Konzepte, Übungen. Addison Wesley 1995
- Kernighan, Brian W./Dennis M.Ritchie: Programmieren in C. dt.Übersetzung, Hanser Verlag 1983
- P.J.Plauger/James Brodie: Referenzhandbuch Standard C (Für den neuen ANSI- und ISO-Standard) Vieweg 1990 (Übersetzung von Microsoft Press 1989)
- ANSI-C: ANSI/ISO 9899:1990 - Programming Language C (C90)
ANSI/ISO 9899:1999 - Programming Language C (C99) (\$18)
<http://webstore.ansi.org/ansidocstore/>, <http://www.iso.org>
- Newsgroup: de.comp.lang.c
- Jutta Degener: <http://snake.cs.tu-berlin.de:8081/~jutta/c/>
- <http://www.c-programme.de/> , <http://gcc.gnu.org/> , <http://www.snippets.org/>
(Quelltexte für jeden Zweck!)
- Brian W. Kernighan: Programming in C: A Tutorial. Bell Laboratories, Murray Hill, N. J. 1974
(siehe INTERNET)
- Dinkum C99 Library: <http://www.dinkumware.com/manuals/reader.aspx?lib=c99>
- GNU C-Compiler unter Windows (GCC)

Programm

```
Programm ::= { [PräprozessorAnweisung] [Deklaration] [Anweisung] main  
( ) { [Deklaration] [Anweisung] } }
```

Mein erstes C-Programm:

```
/* Hello World Programm in C */  
main()  
{  
    printf("Hello World\n");  
}
```

Programmablauf- Steueranweisungen

```
/* Block */  
{  
  deklarationen  
  anweisungen  
}
```

```
/* Mehrfachverzweigung */  
switch (ausdruck)  
{ case ausdruck1: anweisungen1;  
  case ausdruck2: anweisungen2;  
  ...  
  case ausdruckN: anweisungenN;  
  default: anweisungen;  
}
```

```
/* while-Zyklus */  
while (ausdruck)  
{ anweisungen; }
```

```
/* do-Zyklus */  
do  
  anweisungen;  
while (bedingung);
```

```
/* Alternative */  
if (bedingung)  
{ anweisungen; }  
else  
{ anweisungen; }
```

```
/* bedingte Zuweisung */  
wert = (bedingung)?wert1:wert2;
```

```
/* FOR-Zyklus */  
for (anfangswert;wiederholbedingung;Veränderung)  
  { anweisungen; }
```

```
/* Sprünge */  
break;  
continue;  
goto marke;
```

```
/* Beispiel 1: Sekunden in Minuten, Stunden, Sekunden umrechnen */  
  
#define SEK_PRO_MIN 60  
#define SEK_PRO_STD 3600  
unsigned sek, min, std, sek_rest, min_rest;  
  
int main()  
{  
    printf("Eingabe der Sekunden: ");  
    scanf("%d", &sek);  
    std = sek / SEK_PRO_STD;  
    min_rest = sek % SEK_PRO_STD;  
    min = min_rest / SEK_PRO_MIN;  
    sek_rest = sek % SEK_PRO_MIN;  
    printf("%u Sekunden entsprechen ", sek);  
    printf("%u h, %u min und %u s\n", std, min, sek_rest);  
    return 0;  
}
```

```
C:\gcc-2.95.2\bin>gcc _sekunden.c -o _sekunden.exe  
C:\gcc-2.95.2\bin>_sekunden  
Eingabe der Sekunden: 3750  
3750 Sekunden entsprechen 1 h, 2 min und 30 s
```

```
/* Beispiel 2: Erzeugung der Quadratzahlen und Zweierpotenzen */
int  n,i,j;

void main()
{
    printf("Eingabe n= ");
    scanf("%i",&n);
    printf("\nQuadratzahlen:\n");
    for (i=1;i<n+1;i++)
    {
        printf(" %i", i*i);
    }
    printf("\n");
    j=1;
    printf("Zweierpotenzen: \n 1");
    for (i=1;i<=n;i++)
    {
        j*=2;
        printf(" %i",j);
    }
    printf("\nEnde\n");
}
```

```
C:\gcc-2.95.2\bin>gcc _quadrat1.c -o _quadrat1.exe
C:\gcc-2.95.2\bin>_quadrat1
Eingabe n= 10

Quadratzahlen:
 1 4 9 16 25 36 49 64 81 100
Zweierpotenzen:
 1 2 4 8 16 32 64 128 256 512 1024
Ende
```

```
/* Beispiel 3: Erzeugung der Quadratzahlen und Zweierpotenzen mit Formatierung */
int  n,i,j;

void main()
{
    n=10;
    printf("\nQuadratzahlen:\n");
    for (i=1;i<n+1;i++)
        {
            printf("%5i", i);
        }
    printf("\n");
    for (i=1;i<n+1;i++)
        { printf("%5i", i*i);
        }
    printf("\n");
    j=1;
    printf("Zweierpotenzen: \n      1");
    for (i=2;i<=n;i++)
        {
            printf("%6i",i);
        }
    printf("\n");
    for (i=1;i<=n;i++)
        {
            j*=2;
            printf("%6i",j);
        }
    printf("\nEnde\n");
}
```

```
C:\gcc-2.95.2\bin>gcc _quadrature2.c -o _quadrature2.exe  
C:\gcc-2.95.2\bin>_quadrature2
```

Quadratzahlen:

1	2	3	4	5	6	7	8	9	10
1	4	9	16	25	36	49	64	81	100

Zweierpotenzen:

1	2	3	4	5	6	7	8	9	10
2	4	8	16	32	64	128	256	512	1024

Ende

```
/* Beispiel 4: Sternchen-Dreieck */
#include <stdio.h>

int main()
{
    int i,j;
    for (i=1;i<10;i++)
    { for (j=1;j<=i;j++) printf("*");
      printf("\n");
    }
}
```

```
C:\gcc-2.95.2\bin>gcc _dreieck.c -o _dreieck.exe
C:\gcc-2.95.2\bin>_dreieck
*
**
***
****
*****
*****
*****
*****
*****
*****
```

```
/* Beispiel 5: Menueauswahl */
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    char wahl;
    wahl=' ';
    while (wahl != 'B')
    {
        printf("    Hauptmenue\n");
        printf("    =====\n");
        printf("    (A) usgabe \n");
        printf("    (E) ingabe \n");
        printf("    (D) rucken \n");
        printf("    (L) aden \n");
        printf("    (S) peichern \n");
        printf("    (B) eenden \n");
        printf("    Bitte Menuepunkt auswählen! \n");
        wahl = toupper(getch());
        printf("xxx%cy\n", wahl);
        switch (wahl) {
            case 'A' : printf(" \nAusgabe wurde gewaehlt!\n\n"); break;
            case 'E' : printf(" \nEingabe wurde gewaehlt!\n\n"); break;
            case 'D' : printf(" \nDrucken wurde gewaehlt!\n\n"); break;
            case 'L' : printf(" \nLaden wurde gewaehlt!\n\n"); break;
            case 'S' : printf(" \nSpeichern wurde gewaehlt!\n\n"); break;
            case 'B' : printf(" \nDas Programm wird beendet!\n\n"); break;
            default : printf(" \nFalsche Taste!\n\n"); break;
        }
    }
}
```

```
C:\gcc-2.95.2\bin>gcc _menue.c -o _menue.exe
```

```
C:\gcc-2.95.2\bin>_menue
```

```
  Hauptmenue
```

```
=====
```

```
(A) usgabe
```

```
(E) ingabe
```

```
(D) rucken
```

```
(L) aden
```

```
(S) peichern
```

```
(B) eenden
```

```
  Bitte Menuepunkt auswöhlen!
```

```
xxxZyyy
```

```
Falsche Taste!
```

```
  Hauptmenue
```

```
=====
```

```
(A) usgabe
```

```
(E) ingabe
```

```
(D) rucken
```

```
(L) aden
```

```
(S) peichern
```

```
(B) eenden
```

```
  Bitte Menuepunkt auswöhlen!
```

```
/* Beispiel 6: Kreisumfang und Kreisflächeninhalt berechnen */
#define PI 3.14159

float radius, umfang, flaeche;

int main()
{
    printf("Berechnung von Kreisumfang und Kreisfläche zum Radius (in cm) r=");
    scanf("%f", &radius);
    umfang = 2*PI*radius;
    flaeche = PI*radius*radius;
    printf("Der Kreisumfang beträgt: %f", umfang);
    printf("\nDer Kreisflaecheninhalt beträgt: %f", flaeche);
}
```

```
C:\gcc-2.95.2\bin>gcc _kreis.c -o _kreis.exe
C:\gcc-2.95.2\bin>_kreis
Berechnung von Kreisumfang und Kreisflöche zum Radius (in cm) r=3
Der Kreisumfang betrögt: 18.849541
Der Kreisflaecheninhalt betrögt: 28.274309
```

```

/* Beispiel 7: Erzeugung der ASCII-Zeichen */
int  n,i,j;

void main()
{
  printf("ASCII-Zeichensatz:\n ");
  printf("\t128  \200");printf("\t129  \201");printf("\t130  \202");printf("\t131  \203\n");
  printf("\t132  \204");printf("\t133  \205");printf("\t134  \206");printf("\t135  \207\n");
  printf("\t136  \210");printf("\t137  \211");printf("\t138  \212");printf("\t139  \213\n");

  ... u.s.w. ...

  printf("\nEnde\n");
}

```

```

C:\gcc-2.95.2\bin>gcc _zeichentabelle1.c -o _zeichentabelle1.exe
C:\gcc-2.95.2\bin>_zeichentabelle1
ASCII-Zeichensatz:
   128  Ç  129  ü  130  é  131  â  132  ä  133  à  134  á  135  ç  136  ê
137  ë  138  è  139  ì  140  î  141  ï  142  Ä  143  Å  144  É  145  æ  146  Æ
147  ò  148  ö  149  ò  150  û  151  ù  152  ÿ  153  Ö  154  Ù  155  ø  156  £
157  ø  158  ×  159  f  160  á  161  í  162  ó  163  ú  164  ñ  165  Ñ  166  ª
167  °  168  ¸  169  ©  170  ¬  171  ½  172  ¼  173  ¡  174  «  175  »  176  ¶
177  ¶  178  ¶  179  |  180  †  181  Á  182  Â  183  Ã  184  Ä  185  ©  186  ¶
187  ¶  188  ¶  189  ¢  190  ¥  191  ¶  192  L  193  L  194  T  195  ¶
197  †  198  ä  199  Å  200  ¶  201  ¶  202  ¶  203  ¶  204  ¶  205  =  206  ¶
207  ¶  208  ø  209  Ð  210  È  211  É  212  Ê  213  Ì  214  Í  215  Î  216  Ï
217  ¶  218  ¶  219  ¶  220  ¶  221  ¶  222  Ì  223  ¶  224  Ó  225  ß  226  Ô
227  Ò  228  Ò  229  Õ  230  µ  231  þ  232  þ  233  Ú  234  Û  235  Ü  236  Ý
237  Ý  238  ¯  239  ´  240  -  241  ±  242  =  243  ¾  244  ¶  245  §  246  ÷
247  ,  248  °  249  ¨  250  ·  251  ¡  252  ¸  253  º  254  ¶  255

Ende

```

```
/* Beispiel 8: Erzeugung des Quelltextes-Fragmentes des obigen Beispiels */
int i;

void main()
{ for (i=128; i<=255; i++)
  { if (i%4 == 0) {printf(" ");}
    if (i%4 != 3) {printf("printf(\"\\t%i \\%o\");",i,i);}
    else { printf("printf(\"\\t%i \\%o\\");\n",i,i);}
  }
}
```

```
C:\gcc-2.95.2\bin>gcc _erzeuge.c -o _erzeuge.exe
C:\gcc-2.95.2\bin>_erzeuge > ccc.c
```

Dateiinhalt von ccc.c

```
printf("\t128 \200");printf("\t129 \201");printf("\t130 \202");printf("\t131 \203");
printf("\t132 \204");printf("\t133 \205");printf("\t134 \206");printf("\t135 \207");
printf("\t136 \210");printf("\t137 \211");printf("\t138 \212");printf("\t139 \213");
printf("\t140 \214");printf("\t141 \215");printf("\t142 \216");printf("\t143 \217");
printf("\t144 \220");printf("\t145 \221");printf("\t146 \222");printf("\t147 \223");
printf("\t148 \224");printf("\t149 \225");printf("\t150 \226");printf("\t151 \227");
printf("\t152 \230");printf("\t153 \231");printf("\t154 \232");printf("\t155 \233");
printf("\t156 \234");printf("\t157 \235");printf("\t158 \236");printf("\t159 \237");
printf("\t160 \240");printf("\t161 \241");printf("\t162 \242");printf("\t163 \243");
printf("\t164 \244");printf("\t165 \245");printf("\t166 \246");printf("\t167 \247");
printf("\t168 \250");printf("\t169 \251");printf("\t170 \252");printf("\t171 \253");
printf("\t172 \254");printf("\t173 \255");printf("\t174 \256");printf("\t175 \257");
printf("\t176 \260");printf("\t177 \261");printf("\t178 \262");printf("\t179 \263");
printf("\t180 \264");printf("\t181 \265");printf("\t182 \266");printf("\t183 \267");
printf("\t184 \270");printf("\t185 \271");printf("\t186 \272");printf("\t187 \273");
printf("\t188 \274");printf("\t189 \275");printf("\t190 \276");printf("\t191 \277");
printf("\t192 \300");printf("\t193 \301");printf("\t194 \302");printf("\t195 \303");
printf("\t196 \304");printf("\t197 \305");printf("\t198 \306");printf("\t199 \307");
printf("\t200 \310");printf("\t201 \311");printf("\t202 \312");printf("\t203 \313");
printf("\t204 \314");printf("\t205 \315");printf("\t206 \316");printf("\t207 \317");
printf("\t208 \320");printf("\t209 \321");printf("\t210 \322");printf("\t211 \323");
printf("\t212 \324");printf("\t213 \325");printf("\t214 \326");printf("\t215 \327");
printf("\t216 \330");printf("\t217 \331");printf("\t218 \332");printf("\t219 \333");
printf("\t220 \334");printf("\t221 \335");printf("\t222 \336");printf("\t223 \337");
printf("\t224 \340");printf("\t225 \341");printf("\t226 \342");printf("\t227 \343");
printf("\t228 \344");printf("\t229 \345");printf("\t230 \346");printf("\t231 \347");
printf("\t232 \350");printf("\t233 \351");printf("\t234 \352");printf("\t235 \353");
printf("\t236 \354");printf("\t237 \355");printf("\t238 \356");printf("\t239 \357");
printf("\t240 \360");printf("\t241 \361");printf("\t242 \362");printf("\t243 \363");
printf("\t244 \364");printf("\t245 \365");printf("\t246 \366");printf("\t247 \367");
printf("\t248 \370");printf("\t249 \371");printf("\t250 \372");printf("\t251 \373");
printf("\t252 \374");printf("\t253 \375");printf("\t254 \376");printf("\t255 \377");
```

```
/* Beispiel 9: Matrix Ein- und Ausgabe */
int i,j;
int matrix[3][4] = { {11, 12, 13, 14},
                    {21, 22, 23, 24},
                    {31, 32, 33, 34}
                    };

void main()
{
    printf("***** Matrixausgabe *****");
    for (i=0; i<=2 ; i++) /* Zeilen */
    {
        printf("\n\nZeile %i\n", i+1);
        for (j=0; j<=3; j++) /* Spalten */
            { printf("matrix[%d,%d]=%d   ",i+1,j+1,matrix[i][j]);}
    };

    printf("\n\n***** Matrixeingabe *****");
    for (i=0; i<=2 ; i++) /* Zeilen */
    {
        printf("\n\nZeile %i\n", i+1);
        for (j=0; j<=3; j++) /* Spalten */
            { printf("matrix[%d,%d]= ",i+1,j+1);scanf("%d",&matrix[i][j]);}
    }; printf("\n***** Matrixausgabe *****");
    for (i=0; i<=2 ; i++) /* Zeilen */
    {
        printf("\n\nZeile %i\n", i+1);
        for (j=0; j<=3; j++) /* Spalten */
            { printf("matrix[%d,%d]=%d   ",i+1,j+1,matrix[i][j]);}
    };
}
```

```
C:\gcc-2.95.2\bin>gcc _matrix.c -o _matrix.exe
C:\gcc-2.95.2\bin>_matrix
***** Matrixausgabe *****

Zeile 1
matrix[1,1]=11   matrix[1,2]=12   matrix[1,3]=13   matrix[1,4]=14

Zeile 2
matrix[2,1]=21   matrix[2,2]=22   matrix[2,3]=23   matrix[2,4]=24

Zeile 3
matrix[3,1]=31   matrix[3,2]=32   matrix[3,3]=33   matrix[3,4]=34

***** Matrixeingabe *****

Zeile 1
matrix[1,1]= 1
matrix[1,2]= 2
matrix[1,3]= 3
matrix[1,4]= 4

Zeile 2
matrix[2,1]= 5
matrix[2,2]= 6
matrix[2,3]= 7
matrix[2,4]= 8

Zeile 3
matrix[3,1]= 9
matrix[3,2]= 10
matrix[3,3]= 11
matrix[3,4]= 12

***** Matrixausgabe *****

Zeile 1
matrix[1,1]=1   matrix[1,2]=2   matrix[1,3]=3   matrix[1,4]=4

Zeile 2
matrix[2,1]=5   matrix[2,2]=6   matrix[2,3]=7   matrix[2,4]=8

Zeile 3
matrix[3,1]=9   matrix[3,2]=10  matrix[3,3]=11  matrix[3,4]=12
```

```
/* Beispiel 10: procpid.c */

#include "eighdr.h"
#include <sys/types.h>
#include <unistd.h>

int main (void)
{ printf("Meine PID ist ---%d---\n", getpid());
  printf("Die Eltern-PID ist ---%d---\n", getppid());
  printf("Die User-ID ist ---%d---\n", getuid());
  printf("Die effektive User-ID ist ---%d---\n", geteuid());
  printf("Die Group-ID ist ---%d---\n", getgid());
  printf("Die effektive Group-ID ist ---%d---\n", getpegid());
  exit(0)
}
```

```
/* Beispiel 11: uidgid.c */

#include "eighdr.h"
in main(void)
{
  printf("UID = %d", getuid(), "\n GID = %d", getgid(), "\n");
  exit(0);
}
```

```
/* Beispiel 12: Prozesse kreieren mittels forkdemo.c */  
  
#include <sys/types.h>  
#include "eighdr.h"  
int global_var=100;  
int main (void)  
{ int lokal_var;  
  pid_t pid;  
  lokal_var=1;  
  printf("vor dem fork-Aufruf \n");  
  switch (pid=fork())  
  { case -1: printf("Fehler bei fork()\n"); break;  
    case 0: lokal_var++; global_var++;  
            printf("Ich bin der Kindsprozess!\n");  
            break;  
    default:  
  }  
  printf("%s; global_var=%d, lokal_var=%d\n",  
        (pi==0)?"Kindsprozess": "Elternprozess", global_var, lokal_var);  
  exit(0);  
}
```

```
/* Beispiel 13: geschachtelte Prozesse */  
  
#include <sys/types.h>  
#include "eighdr.h"  
  
int main (void)  
{ int var=0;  
  printf("vor dem fork-Aufruf \n");  
  if (fork()==-1) {printf"Fehler beim ersten fork()"}  
  else  
    { var++;  
      printf("var=%d\n,var);  
      if (fork()==-1) {printf"Fehler beim zweiten fork()"}  
      else  
        { var++;  
          printf("var=%d\n,var);  
        }  
    }  
  exit(0);  
}
```