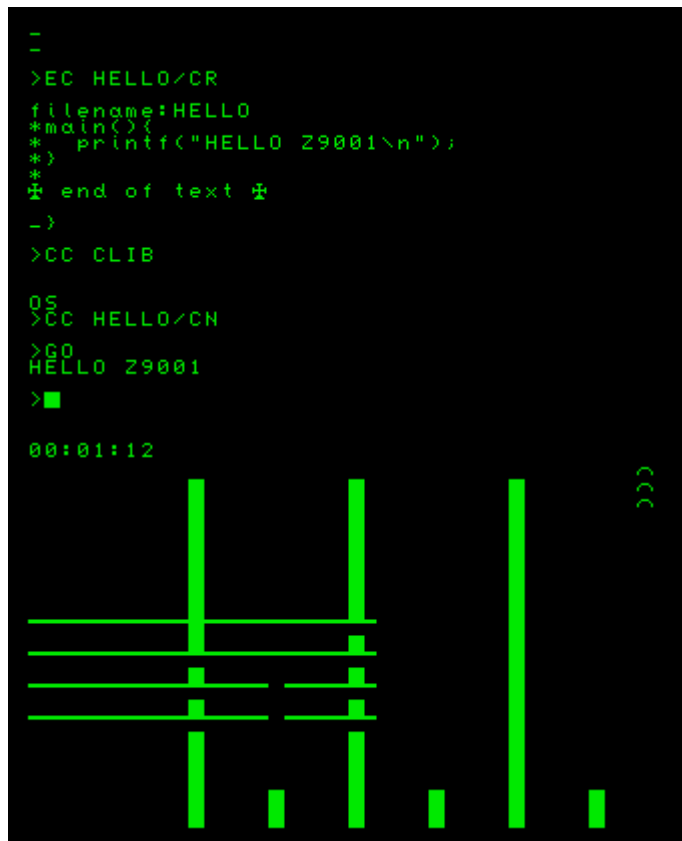


# Pretty C

## Ein C-Compiler für den KC 85/1

Dr. Reinhard Wobst, 1987

Pretty C ist ein kleiner Einpass-C-Compiler zum Lernen der Sprache C (K&R-Syntax!). Er wurde speziell für Kleincomputer und Kassetten-Rekorder als Massenspeicher entwickelt.



Vor Nutzung des Compilers ist es erforderlich, das Handbuch zu lesen!

Geladen wird der Compiler mit

```
>PRETTYC
```

Dann stehen die Kommandos

- EC Editor
- CC Compiler
- C@ Kommandoprozessor
- GO Programmausführung

zur Verfügung.

## Downloads

prettyc.zip

- Handbuch Version 1.0 incl. Ergänzung Version 1.1 (digitalisiert U. Zander)
- Beschreibung K&R C (Aus: Kernighan/Ritchie, The C Programming language, 1978, Anhang C). In Englisch!
- Softwarepaket (PRETTYC.COM, INSTALL.COM, Bibliotheken CEXT,CLIB,CLIBM,DEBUG, Beispiele HELLO, SCHAUKELE, SIEB)
- mein Patch

## Hinweise

Die originale Version lässt nur mit Kassettenrekorder nutzen! Meine gepatchte Version arbeitet auch mit USB und SD-Karten.



Bei „filename“ darf **keine Endung** angegeben werden, nur max. 8 Zeichen für den Dateinamen. Die Originalversion ignoriert einen Dateityp; die gepatchte Version erwartet keinen Dateityp. Der Filename muss bei USB/SD in **Großbuchstaben** eingegeben werden (FAT-Filesystem)!



Es darf kein [Schalt-ROM-Modul](#) zum Einsatz kommen! Der Compiler macht Schreibzugriffe auf OFFFEh und weitere Adressen, die zum Abschalten eines Schalt-ROMs führen!

Auf der Tastatur nicht vorhandene Zeichen wie [, {, ~ können aus vorhandenen Zeichen durch nachfolgendes Drücken von '@' dargestellt werden (s. Handbuch Abschnitt 2.3.1, Sondertasten).

```
>CD PRETTYC
>PRETTYC
```

### Empfehlung

```
erst Buffer für CLIB
dann Buffer für CMD
dann Buffer MAIN für Programme
```

```
CMD: Erst MAIN kompilieren, dann CLIB; denn das Kompilieren von CLIB dauert
recht lang,
und so sieht man eigene Fehler im MAIN sofort, und muss nicht immer warten,
bis alles
compiliert ist.
```

```
>EC CLIBM/CR          Textpuffer CLIBM anlegen
*                    Leerzeile
*STRG-C              Insert-Mode beenden
ESC+R,CLIBM          Datei CLIBM einlesen
```

```
ESC+E          Editor Ende

>EC CMD/CR      Textpuffer CMD anlegen
*CC MAIN/BP:0/LI:0
*CC CLIBM/CN
*STRG-C        Insert-Mode beenden
ESC+E          Editor Ende
```

```
EC MAIN/CR      Textpuffer MAIN anlegen
*              Leerzeile
*STRG-C
ESC+E          Editor Ende
```

Speicherabzug CLIBM+CMD+MAIN auf Band sichern, dann spart man sich diese Schritte,  
sondern lädt einfach im OS.  
( prettyc muss aber vorab init werden, dazu einfach einmal CC ausführen )

```
>EC /DI          alle Textpuffer auflisten
8000 8E8F CLIBM
9000 902A CMD
9200 9217 MAIN
```

```
>SAVE 8000 921F FFFF    OS-Kommando SAVE
filename: EC_CLIB.COM
```

Dann alles compilieren mit  
>C@  
und Start mit  
>G0

## Bedienung

```
EC /DI          Directory
```

```
EC TEST/CR      legt Puffer mit Namen TEXT an
```

Cursor runter/cursor hoch - durch die Zeilen blättern

```
ESC+T    Pufferfanang
ESC+B    Pufferende
```

\* kennzeichnet Insertmodus (neue Zeilen)  
es muss mindestens eine Zeile da sein, also mindestens 1x Enter drücken

```
^C -> _ (Kommandomode)
^D  Zeile löschen
^E  Insertmodus
```

**ESC+W WRITE:**

Ausgabe des gesamten Puffers auf Magnetband

**ESC+R READ:**

Lesen eines Files von Magnetband, Anhängen an das Pufferende. WIRD STATT DES Filenamens nur 'ENTER' gegeben, so wird dieser Vergleich unterdrückt

ESC+C Großbuchstaben nutzen

ESC+E end

Textpuffer mit dem Namen CMD schreiben.

OS-Kommando

C@

arbeitet seine Zeilen nacheinander ab.

CC CQ/BP:2/LI:1/AS:100

CC debug/BP:0/LI:0/CN

CC printf/CN

## Internes

Es wird reservierter Systemspeicher genutzt. Auch die RST-Adressen werden verwendet. Das muss bei anderen Erweiterungen (z.B. Mega-Flash) beachtet werden!

90 - 18F Stack für Syntaxanalyse (nur Compiler)

210- 2FF interne Puffer (Compiler + Laufzeit)

300 - 4F00 Pretty C mit Arbeitszellen

...

From:

<https://hc-ddr.hucki.net/wiki/> - Homecomputer DDR

Permanent link:

<https://hc-ddr.hucki.net/wiki/doku.php/z9001/software/prettyc?rev=1750852267>

Last update: **2025/06/25 11:51**

