R.Wobst 21.04.88

Ergänzungen und Änderungen zur Dokumentation von Pretty C - Version 1.1

Zum Speicherbedarf:

- Quelltextfiles dürfen erst ab 5300H beginnen (und nicht ab 5200H wie angegeben). Dann ist jedoch keine Übersetzung mehr möglich, d.h. es kann nur mit dem Editor gearbeitet werden.

- Die Längen von CLIB und CLIBM (und des erzeugten Codes) ändern sich z.Z. noch und sind kürzer als angegeben. Sie sind mit den Kommandos EC /DI und CC /DS zu ermitteln.

Zu 1.0

Als weiteres Kommando ist noch

CHS

implementiert worden ("checksum"). Vom Compiler wird eine Prüfsumme errechnet und als vierstellige Hexadezimalzahl ausgegeben.
Diese Zahl darf sich n a c h einer Installierung nicht mehr ändern.
Damit kann der Nutzer kontrollieren, ob durch Programmfehler versehentlich der Compiler überschrieben wurde.

Zu 3.3.5

Die Variable 'val' hat den Typ 'unsigned *' und nicht 'char *'.

Zu 6.2

Die arcsin-Funktion heisst 'asin' und nicht 'arcsn'.

Zu 8.1, Punkt 1

Auch die Funktion '_debvd' ist reserviert.

Zu 8.2

- Zusätzlich wurde das Kommando 'd' (delete) aufgenommen, daß das letzte f-Kommando rückgängig macht. Durch wiederholte Anwendung des d-Kommandos können so nacheinander alle Anzeigen von Variablen gelöscht werden.

- Die Anzahl der angezeigten Quelltextzeilen ist vorgegeben.
  Sind jedoch einzelne Zeilen länger als 40 Zeichen, so wird das Bildschirmfenster für den Quelltext entsprechend größer. Bei nachfolgender Kontraktion dieses Fensters bleiben Teile der alten Begrenzung erhalten. Das sollte nicht stören.

- Wird der gerade angezeigte Wert mit dem 'w'-Kommando verändert, so kann das unbeabsichtigte Nebenwirkungen haben. Der Wert muß nämlich nicht in einer Zelle für Zwischenergebnisse stehen, d.h. man kann auf diese Weise Variablen oder sogar Konstanten verändern!

Beispiele:
Die angezeigte Anweisung sei
while (a+b) ...
Das Ergebnis 'a+b' steht in einer Stackzelle für Zwischenergebnisse, seine Veränderung hat höchstens auf seine Auswertung in der while-Bedingung Einfluß.
Wird dagegen
if(a) ...
angezeigt, so verändert 'w'-Nr.0 gleichzeitig die Variable a.
Und in
puchar('\n');
darf man das Argument '\n' keinesfalls verändern, da sonst die Konstante 'newline' überschrieben wird.
Also: VORSICHT!

Zu 8.3

Durch Vergrößerung bzw. Verkleinerung des Wertes für die symbolische Konstante ANIM kann der 'animation mode' verlangsamt bzw. beschleunigt werden.

Zu Anhang A, Speicherklassen

Das Schlüsselwort 'extern' wird stets als Speicherklasse aufgefaßt, nie zur bloßen Kennzeichnung einer Vorabdeklaration. Es ist also nicht gestattet, 'static'-Variablen mit 'extern' vorab zu deklarieren.

Noch bekannte Fehler in Version 1.1:
---

1. Enthält ein Pointer eine Adresse größer oder gleich 8000H, so wird er fälschlicherweise als negative Zahl betrachtet.
   Dadurch fallen Vergleiche mit diesem Pointer falsch aus.
   Ausweg: Konvertierung auf 'unsigned' und Umkehrung der Relation, z.B.
   ```
   char *p,*q; p=0x8800; q=0x3000;
     ...
   statt if(q<p) ...  schreibe if((unsigned)q > (unsigned)p) ...
   ```

2. Wird eine Konstante auf einen Pointertyp konvertiert, so ergibt die *-operation keinen l-value.
   Ausweg: Zuweisung der Konstante an einen Pointer. z.B.:
   ```
   statt *(long*)0x6666 = ...   schreibe
     long *pl; ... *pl=0x6666) = ...
   ```

3. Beim Syntaxfehler
   ```
   int a,n; b[3];
   ```
   erscheint die unsinnige Meldung "missing '}' declaration".
   Der Fehler wird aber richtig lokalisiert.

4. Die Konvertierung float->double ist falsch und kann sogar zum Absturz führen. Sie ist aber leicht zu vermeiden.

5. Bei gesetztem Schalter /I+ werden Ausdrücke der Form
   Pointer + Integer
   falsch übersetzt.

6. Wird beim letzten Morphem einer Zeile ein Fehler erkannt und ist /LI:1 gesetzt, so erscheint diese Zeile doppelt.
   Ich bitte alle Nutzer um eine Vervollständigung dieser Liste!