11 neue Einzeiler!

Der Einzeiler-Wettbewerb regt immer noch eine ganze Reihe Leser an, möglichst interessante Lösungen in eine Basic-Zeile zu packen. Wir haben wieder die besten für Sie ausgewählt.

Der Einzeiler-Wettbewerb ist nicht klein zu kriegen — und das wollen wir auch nicht. Viele Einsender nahmen die bisher veröffentlichten Mini-Programme als Anregung für Ihre nächsten »Schöpfungen«. Vor allem »Das kürzeste Adventure der Welt« erhielt große Resonanzen und Nachahmer, sowohl positive als auch negative. Natürlich kann man so eine Idee nur einmal verwerten — es war halt ein gelungener Einfall und mehr als Scherz gemeint. Doch diesmal haben wir wieder »ernste« Einzeiler herausgesucht, vielleicht bis auf den »aufgeregten Marsmenschen«. Auch die VC 20-Besitzer kommen diesmal nicht zu kurz. Doch lassen Sie sich überraschen!

Aufgeregter Marsmensch (VC 20 und C 64)

Das Programm spricht für sich selbst. Lediglich der POKE-Befehl verdient eine zusätzliche Anmerkung: Der Standardwert für die Speicherstelle 37877 ist 72. Indem man ihm auf 0 setzt, wird eine wirkungsvolle Verlangsamung des Programmablaufes erreicht. (Gerhard Silberbach)

```
1 PRINT"{HOME}{SHIFT-*}{SHIFT-*}{SHIFT-*}{SHIFT-*}{HOME,DOWN,SPACE,RVSON,
  DOWN}{SHIFT-*} {SHIFT-*}":PRINT"{SHIFT-*}{RVSON}{SHIFT-*} {RVOFF}{SHIFT-*}":PRI
  NT"{SPACE,RVSON,SPACE}{SHIFT-*} ":PRINT" {SHIFT-*}{RVSON
  }{SHIFT-*}{RVOFF}{SHIFT-*}":PRINT" {SHIFT-*} {SHIFT-*}":POKE 37877,0:PRI
  NT"{HOME}{SHIFT-*}{SHIFT-*}{SHIFT-*}{SHIFT-*}{HOME,2DOWN}({RIGHT,2DOWN,
  RVSON,SPACE}":RUN
10 :
20 REM  MARSMENSCH
```

Grafik-Bilder invertieren

Dieser Einzeiler invertiert eine Hires-Grafik an der Adresse A. Um das besser begreifen zu können, sollten Sie eine Hires-Grafik laden. Nehmen wir als Beispiel ein Bild aus der Dia-Show und laden es mit LOAD"Name",8,1 ab der Adresse $2000 (dezimal 8192) ein. Tippen Sie jetzt NEW ein und geben den Einzeiler ein. Jetzt geben Sie der Variable A den Wert des Grafik-Anfangs, als A=8192. Springen Sie dann mit GOTO1 in den Einzeiler. Die Grafik ist jetzt also invertiert, Sie können den Bereich 8192 bis 16384 als invertierte Grafik wieder abspeichern.

Erklärung der Zeile: Der String A$ enthält ein Maschinenprogramm, das durch den Print-Befehl in den Bildschirmspeicher ab 1024 gebracht wird. Die aktuelle mit EXOR FF zu verknüpfende Speicherstelle wird mit POKE 780,x im Akku abgelegt. Nach SYS 1024 steht dann der Wert in 780 und wird mit dem anschließenden POKE in die entsprechende Speicherstelle als invertiertes Bitmuster gebracht.

Noch eine Bemerkung: Die Grafik darf natürlich nicht unter einem ROM liegen, da dann das Problem nur in reiner Maschinensprache zu lösen ist (das ROM müßte abgeschaltet werden). (Guido Leister/gk)

```
1 A$="{HOME}{SHIFT-+}{RVSON}{SHIFT-B}{RVOFF,SHIFT-SPACE}":
  PRINT A$:FOR I=0 TO 8191:POKE 780,PEEK(A
  +I):SYS 1024:POKE A+I,PEEK(780):NEXT
10 :
20 REM  HIRES-INVERS
```

Zeilen löschen am Bildschirm

Dieser kleine Einzeiler löscht bestimmte Zeilen auf dem Bildschirm. Dabei wird eine Maschinenroutine des C 64 benutzt, die eine Zeile vom Bildschirm löscht, deren Zeilennummer im X-Register steht. Zunächst wird die Zeilennummer (hier wird von 0 bis 24 gezählt) in die Speicherzelle gePOKEt, deren Inhalt der SYS-Befehl in das X-Register übernimmt. Hierbei bleibt die Position des Cursors unbeeinflußt.

Variablen:

LN = Zeilennummer (0 bis 24)
V = Von Zeile
B = Bis Zeile (Stefan Keimeier/gk)

```
10 FOR LN=V TO B:POKE 781,LN:SYS 59903:NEXT
20 :
30 REM FUER NUR EINE ZU LOESCHENDE ZEILEGILT:
40 :
50 POKE 780,LN:SYS 59903
60 :
70 REM                    STEFAN KEIMEIER
```

Ein einfaches Renumber

Dieser Einzeiler kann sich zwar nicht mit einem komfortablen Renumber-Programm messen, aber er funktioniert. Es werden die Zeilennummern eines Basic-Programms, das nicht länger als 255 Zeilen sein darf, in nur wenigen Sekunden neu umnumeriert (erste Zeilennummer = 0, Schrittweite = 1). Die Sprungadressen der Befehle GOTO und GOSUB bleiben jedoch unverändert. Beim Abtippen des Einzeilers ist zu beachten, daß die Basic-Befehle FOR, PEEK, POKE und NEXT abgekürzt werden müssen (siehe C 64-Handbuch, Anhang D).

Nun zur Erklärung des Einzeilers: Um die Zeilen eines Programms umnumerieren zu können, muß man zunächst wissen, an welchen Speicherstellen es sich befindet. Jedes Basic-Programm belegt die Speicherplätze 2048 bis PEEK(45)+PEEK(46)*256-3. PEEK(2049)+PEEK(2050)*256 gibt an, bei welcher Adresse die erste Zeile aufhört. Die Adressen 2051 und 2052 geben Aufschluß über die erste Zeilennummer (=PEEK(2051)+PEEK(2052)*256). Die zweite Zeilennummer findet man im Speicher an den Adressen PEEK (2049+PEEK(2050)*256+2 und PEEK(2049)+PEEK (2050)*256+3.

Zurück zum Renumber-Programm: Die Zeilennummern befinden sich jeweils an den Adressen A+2 (Low Byte) und A+3 (High Byte). Aus Platzgründen POKE ich an die Stelle A+2 den Wert z (z= 0,1,2..., n-1; n-1 steht für die Anzahl der Zeilen des Basic-Programms, das umnumeriert werden soll, inklusive dem Einzeiler) und an die Stelle A+3 den Wert 0. Daher darf das Basic-Programm 255 Zeilen nicht überschreiten, denn dann müßte ich den Einzeiler zum Zweizeiler abändern:

1 FOR A=2049 TO PEEK(45)+ PEEK(46)*256-3:POKE A+3,z/256:POKE A+2, z-INT(z/256)*256
2 A=PEEK(A)+ PEEK(A+1)*256-1:z=z+1:NEXT

(Georg Wichert/gk)

```
1 FOR A=2049 TO PEEK(45)+PEEK(46)*256-3:PO
  KE A+2,Z:POKE A+3,0:A=PEEK(A)+PEEK(A+1)*
  256-1:Z=Z+1:NEXT
10 :
20 REM RENUMBER (GEORG WICHERT)
```

Datum wandeln in Wochentag

Dieser Einzeiler berechnet nach Eingabe eines Datums den entsprechenden Wochentag.

Es bedeutet: 0 Sonntag, 1 Montag etc.

Benutzte Variablen:

T,M,J für Tag, Monat, Jahr

Die benutzte Gleichung ist eine Vereinfachung der folgenden:
T=T+365*J+INT((J+(M>3))/4)+31*M-31+2*(M>2)-INT((M-1+(M>8))/2))

Die erste INT-Funktion ersetze ich durch die Integerviariable T%. Weitere Vereinfachungen ergeben sich durch den Gebrauch der Modulo-Funktion, die in der Print-Anweisung benutzt wird. Es kommt durch Anwendung bestimmter Rechenregeln der Mod-Funktion zu folgenden Vereinfachungen:
365*J ergibt J, aus 31*M-31 wird 3*M-3.

Mit der jetzt erhaltenen Formel würde man schon auskommen. Um aber eine gebräuchliche Zahl-Wochentagszuordnung zu erhalten, addiere ich noch 2 zu T%.
Zur Erklärung der Gleichung:

Es wird die Gesamtzahl aller Tage von 0.0.0000 bis einschließlich des eingegebenen Datums berechnet. Nach Division durch 7 ergibt der ganzzahlige Rest den gesuchten Wert. Der Rest wird in der Print-Anweisung durchgeführt (Mod-Funktion). Die Gleichung selber ist wie folgt aufgebaut:

T	ist die Zahl der Tage im laufenden Monat,
365*J	sind die Tage aller vorherigen Jahre,
INT((3+(M<3))/4)	sind alle vorherigen Schalttage,
31*M-31	sind die Tage aller vorherigen Monate, allerdings mit dem Fehler, das einige Monate weniger als 31 Tage haben,
2*(M>2)	es werden 2 abgezogen wenn M>2 ist, das heißt aus Februar wird ein Monat mit 30 Tagen gemacht,
-INT((M-1+(M>8))/2	hier wird für jeden Monat mit 30 Tagen einer abgezogen.

Alle Teile werden addiert und so wie oben beschrieben umgeformt.

(Wilfried Mintrop/gk)

```
1 INPUT T,M,J:T%=T+J+(J+(M<3))/4+3*M+2*(M>
  2)-INT((M-1+(M>8))/2)+2:PRINT T%-7*INT(T
  %/7)
2 :
3 :
4 REM DATUM WANDELN
```

Soft-Scrolling beim C 64

Mit diesem Einzeiler kann ein beliebiger Text von rechts nach links punktweise über den Bildschirm verschoben werden.

Als Variablen werden verwendet:

A=53270	Register für horizontales Smooth-Scrolling
L=40	Anzahl der Zeichen, die gleichzeitig auf dem Bildschirm erscheinen sollen. (40 für gesamte Bildschirmbreite; weniger für kleineren Textausschnitt.)
A$	Soll den zu zeigenden Text enthalten. Letztes Zeichen des Strings sollte ein SPACE sein, weil das rechtsbündige Zeichen des Strings sich auf dem Bildschirm dupliziert, da es nicht gelöscht wird.

Das Prinzip: Der Text wird auf dem Bildschirm ausgegeben. Nun wird der Bildschirminhalt mit Hilfe des Smooth-Scrolling Registers punktweise nach links gezogen, bis er 7 Punkte verschoben worden ist. Jetzt wird der gesamte Text nach links geschoben und (fast) gleichzeitig das Scroll-Register zurückgesetzt, so daß es aussieht, als sei der Text um den achten Punkt verschoben worden.

Hinweis:
— Vor der Benutzung empfiehlt es sich, den Bildschirm zu löschen, da auch der restliche Bildschirminhalt verschoben würde.
— Außer dem verwendeten HOME können noch andere Cursor-Steuerzeichen eingesetzt werden, um den Text zu positionieren.

(Georg Brandt/gk)

```
1 FOR R=1 TO LEN(A$):FOR I=207 TO 200 STEP
  -1:PRINT"{HOME}"MID$(A$,R,L):POKE A,I:NE
  XT I,R
10 :
20 REM SCROLL
```

Zugriffszeit der Floppy verkürzen

Der vorliegende Einzeiler dient dazu, die Zugriffszeit der Floppy-Disk 1541 drastisch zu verkürzen. Der Schrittmotor, der den Schreib-Lesekopf bewegt, kann erfahrungsgemäß wesentlich schneller arbeiten, ohne daß eine sichere Funktion der Floppy gefährdet wird. Da der Schrittmotor im Interrupt bedient wird, genügt es, die Größe des Interruptintervalls zu verändern, um die Drehzahl des Motors zu beeinflussen. Standardmäßig wird etwa alle 15 Millisekunden ein Interrupt ausgelöst, der den Stepper um eine Viertelspur bewegt. Durch das vorliegende Programm wird diese Zeit auf etwa 4 Millisekunden verkürzt. Alle Bewegungen des Kopfes werden dadurch fast viermal schneller. Das hat neben der Zeitersparnis noch zwei weitere wesentliche Vorteile: Das Laufgeräusch des Kopfes wird angenehm leise und kurz, und im Falle einer Kopfjustage (MG-salvenartiges Geräusch) fährt der Kopf mit erheblich verminderter Kraft gegen den Anschlag, so daß die Gefahr einer Dejustage deutlich gemindert ist.

(Robert Loos/gk)

```
10 OPEN 1,8,15,"M-W"+CHR$(7)+CHR$(28)+CHR$
   (1)+CHR$(15)
20 :
30 REM ZUGRIFFSZEIT DER FLOPPY KUERZER
```

Input mit Komma

Die Idee:

Eine INPUT-Routine, die den normalen INPUT-Befehl ersetzt, aber zusätzlich Satzzeichen wie Komma, Doppelpunkt und Strichpunkt als Eingabe erlaubt. Alle sonstigen Vor- und Nachteile bleiben erhalten.

Die Wirkung:

Alle Zeichen der Tastatur werden übernommen, auch Leerzeichen vor beginnendem Text (führende Leerzeichen).

Variablenliste:

AA	= aktueller ASCII-Code der Eingabe
II	= Laufvariable für Schleife
XX$	= enthält eingegebenen Text

Programmbeschreibung:

Das »Herz« des Programms bildet die Eingaberoutine ab Adresse 42336. Diese schreibt alle, von einer Bildschirmzeile (80 Zeichen) erfaßten Zeichen in den Basic-Eingabepuffer, welcher bei Adresse 512 beginnt. Das Eingabeende wird im Puffer mit einer 0 gekennzeichnet. Der Rest des Programms liest nun Zeichen für Zeichen bis zur genannten 0 den Eingabepuffer aus, und stellt dabei den String XX$ zusammen. Die Schleife ist zwar auf 88 Durchläufe programmiert, wird aber niemals soweit kommen, da die Zeichenzahl durch den Bildschirm begrenzt wird (2*40 Zeichen/Zeile=80 Zeichen). Erfolgt keine Eingabe (es wird nur die RETURN-Taste betätigt), so wird das Programm mit XX$ = CHR$(32) verlassen. Ansonsten enthält XX$ alle sichtbaren, eingegebenen Zeichen. Also alle außer Steuerzeichen.

(Jürgen Reinert/gk)

```
1 SYS 42336:XX$="":FOR II=512 TO 600:AA=PE
  EK(II):IF AA THEN XX$=XX$+CHR$(AA):NEXT
10 :
20 REM INPUT
```

Spiralen mit dem Plotter 1520

Das Programm malt mit dem Plotter 1520 eine Spirale, die aus lauter Dreiecken entsteht. Zugrunde liegt eigentlich die Berechnung eines Kreises, da die X-Koordinaten mit SIN und die Y-Koordinaten mit COS ermittelt werden.

Heraus kommt jedoch eine Spirale, da X und Y mit der Laufvariablen I multipliziert werden.

Wird der Faktor »2« innerhalb der Sinus und Cosinus-Klammern gegen einen anderen Wert vertauscht, so ändert sich auch das Aussehen der Grafik.

Auch dieser Einzeiler muß wieder mit den abgekürzten Basic-Befehlen eingegeben werden.

(Christoph von Rhein/gk)

```
1 OPEN 1,6,1:PRINT#1,"M",240,0:FOR I=1 TO
  250:PRINT#1,"S",240+SIN(I*2)*I,COS(I*2)*
  I:NEXT
```

Doppelt großer Zeichensatz für den VC 20

Mit dieser kleinen Routine ist es möglich, sich einen neuen Zeichensatz zu erstellen. Die neuen Zeichen haben die gleiche Breite, jedoch die doppelte Höhe. Es lassen sich alle Zeichen (mit Ausnahme der Sonderzeichen) darstellen.

Alle vier POKEs müssen abgekürzt werden: P geshiftetes O, ebenso PEEK:P geshiftetes E und PRINT:?

Mit POKE36867,25 wird die Video-Matrix der Zeichen auf eine 16 x 8-Matrix umgestellt. POKE36869,205 bestimmt die Lage des Zeichensatzes (bei Grundversion ...,255). »a« beinhaltet den Wert der Speicherzellen 32768 bis 33279. »x« ist die Laufvariable. Die neuen Zeichen werden letztlich mit den zwei POKEs definiert.

Bei Verwendung der Grundversion muß man die Commodore- und die Shift-Taste nach dem Start zusammen drücken.

(Heiko Schmidt/gk)

```
1 POKE 36867,25:POKE 36869,205:FOR X=0 TO
  511:A=PEEK(32768+X):B=X*2:POKE 5120+B,A:
  POKE 5121+B,A:NEXT:PRINT"{CLR}"
5 :
10 GROSSER ZEICHENSATZ VC 20
```

Schlagzeug für VC 20

Dieser Einzeiler macht aus dem VC 20 ein heißes Schlagzeug, das bei entsprechender Lautstärke kaum noch von einem echten zu unterscheiden ist.

Das Programm ist auf jeder Speicherausbau-Version lauffähig und braucht wohl nicht extra erklärt zu werden, denn so kompliziert ist es nicht. Dennoch ist es erstaunlich, was es aus dem VC 20 macht.

Es ist ein Beispiel für ein Programm mit guter Idee, relativ einfacher Umsetzung und erstaunlicher Wirkung.

(Hannes Kaltenbach/ev)

```
1 POKE 36877,RND(1)*256:FOR I=15 TO 0 STEP
  -1:POKE 36878,I:FOR T=1 TO 2:NEXT T,I:GO
  TO 1
```

Einige Einzeiler enthalten Steuer- und Grafikzeichen, die aus Gründen der Übersichtlichkeit als Klartext ausgegeben sind. Bitte beachten Sie unseren Beitrag über den Checksummer.

VC 20-Tips

SYS 64802 Wirkt wie ein Reset
SYS 64821 Stellt Einschaltmodus wieder her
SYS 65499 Setzt »TI$« und »TI« wieder auf Null
SYS 65511 Schließt alle Files
? PEEK(152) Ergibt die Anzahl der offenen Dateien
? PEEK(182) Ergibt nach »LOAD« die Anzahl der Lesefehler
? PEEK(186) Ergibt die zuletzt benutzte Gerätenummer
? PEEK(202) Ergibt die Cursorspalte
? PEEK(214) Ergibt die Cursorzeile
POKE 792,34 :POKE 793,253 Reset nach RESTORE-Taste
POKE 792,173:POKE 793,254 RESTORE wieder normal
POKE 818,34 :POKE 819,253 SAVE-Schutz einschalten
POKE 818,133:POKE 819,246 SAVE-Schutz ausschalten

(Herbert Lickes)

Mehr Struktur mit Spaces

Die Strukturierung von Basic-Programmen wird bei Commodore-Computern dadurch etwas behindert, daß der Interpreter führende Leerzeichen zwischen Zeilennummer und Basic-Befehl einfach überliest. Beim LISTen erscheint stets genau ein Space nach der Zeilennummer, was immer dann unerwünscht ist, wenn der Übersicht halber einige Programmzeilen eingerückt erscheinen sollen. In der Regel hilft man sich in solchen Fällen, indem ein Doppelpunkt an den Zeilenanfang geschrieben wird, auf den dann die gewünschte Anzahl von Leerzeichen folgt.

Es geht jedoch auch wesentlich eleganter. Man kann sich einfach die Tatsache zunutze machen, daß der Basic-Interpreter bei der Zeileneingabe Grafikzeichen einfach überliest. Tippen Sie nach der Zeilennummer irgendein Grafikzeichen (oder der Einfachheit halber SHIFT und SPACE gleichzeitig), danach die gewünschte Anzahl führender Spaces und dann die vorgesehenen Basic-Befehle. Zwar steht zunächst noch das Grafikzeichen mit auf dem Bildschirm, beim AufLISTen werden Sie jedoch feststellen, daß das Grafikzeichen nicht in den Programmtext übernommen wurde. Dennoch erscheint die Zeile um die gewünschte Anzahl von Leerstellen eingerückt.

Mit dieser Methode lassen sich übrigens auch echte Leerzeilen erzeugen. Geben Sie zur Demonstration einmal folgendes ein:

Irgendeine Zeilennummer, gefolgt von einem beliebigen Grafikzeichen, ein Space und zum Schluß nochmals ein Grafikzeichen. Nach Drücken der RETURN-Taste wird nur das eine Space in die Zeile übernommen, wodurch nach LIST eine Zeile erscheint, die nur aus der Zeilennummer besteht.

Zum Schluß in diesem Zusammenhang noch ein POKE-Befehl:

Nach »POKE 129,58« werden Basic-Zeilen, die nicht in Anführungszeichen stehende Spaces enthalten, einfach nicht mehr ausgeführt. Der Interpreter meldet dann nur noch »?SYNTAX ERROR«. Mit »POKE 129,32« erreicht man wieder den Normalzustand.

(Herbert Heise)

Nützliche POKEs für den C 64

POKE 650,100 Cursor blinkt langsam
POKE 788,81 Cursor bleibt stehen
POKE 788,52 Cursor blinkt schnell
POKE 788,51 Cursor abgeschaltet
POKE 788,49 Cursor blinkt normal
POKE 775,199 LIST abgeschaltet
POKE 775,167 LIST normal
POKE 792,193 RESTORE-Taste aus
POKE 792,71 RESTORE-Taste ein
POKE 657,128 COMMODORE-Taste aus
POKE 657,0 COMMODORE-Taste ein

(Peter Gaß)