



Hilfe gegen Absturz

Bei der Zeichensatzveränderung stürzt mein C 64 in einigen Fällen nach »POKE 648,192« und Löschen des Bildschirms rettungslos ab. Wo liegt der Fehler, und was kann ich dagegen tun?

Peter Scholz

Ausgabe 11/85

Der vermutete Absturz ist keiner. Der C 64 besitzt zwei Register für die Lage des Bildschirmspeichers. Das erste ist das in der Frage erwähnte bei der Speicheradresse 648. Hier ist das High-Byte der Bildschirmadresse für das Betriebssystem festgehalten.

Darüber hinaus gibt es auch das Register 24 des Video-Controllers (VIC), nämlich die Speicherstelle 53272. Hier ist in den höchsten vier Bits noch einmal die Adresse des Bildschirmspeichers, und zwar geteilt durch 1024, festgelegt. Dieses Register wird vom VIC zur Ausgabe des Bildes benutzt.

Wird nun durch »POKE 648, 192« die Bildschirmausgabe nach \$C000 oder dezimal 49152 umgelenkt, so liegt zwar aus der Sicht des Betriebssystems der Bildschirmspeicher bei 49152, der VIC aber zeigt auf dem Fernseher oder Monitor immer noch den Bereich ab \$0400 oder dezimal 1024. Um also den Bildschirmspeicher wirklich im neuen Bereich zu haben, muß noch das Register im VIC korrigiert werden. Jetzt beginnt die Arbeit: Der VIC kann nämlich von Haus aus nur 16 KByte des Speichers adressieren. Um nun trotzdem den gesamten Speicher von 64 KByte erfassen zu können, teilt er ihn sich in vier Blöcke ein. Zur Auswahl eines dieser Blöcke stehen die beiden Bits PA0 und PA1 in der CIA 2 zur Verfügung. Auf deutsch gesagt: Die beiden untersten Bits der Adresse 56576.

Bei der Wahl der Blöcke gilt: Block 1 von 0 bis 16383:

Blockzahl = 3

Block 2 von 16384 bis 32767:

Blockzahl = 2

Block 3 von 32768 bis 49151:

Blockzahl = 1

Block 4 von 49152 bis 65535:

Blockzahl = 0

Zur Wahl eines bestimmten Blockes gibt man ein:

POKE 56576, PEEK(56576) AND 252 OR Blockzahl

Jetzt kommen wir zur Lage des Bildschirmspeichers innerhalb des 16-KByte-Blocks. Dazu zieht man von der Startadresse des Bildschirmspeichers die Anfangsadresse des Blocks ab und dividiert das ganze durch 64. Es ist zu beachten, daß der Bildschirmspeicher nur in Schritten zu 1 KByte verschoben werden kann, das heißt die Startadresse des Bildschirmspeichers muß durch 1024 teilbar sein. Jetzt gibt man ein:

POKE 53272, PEEK(53272) AND 15 OR (errechneter Wert)

Der Bildschirmspeicher stimmt jetzt. Aber kaum ist das Problem gelöst, taucht das nächste auf: Der Zeichengenerator (oder der Zeichensatz). Die Schwierigkeit ist, daß der fest eingebaute Zeichensatz nur verwendet werden kann, wenn man ihn bei \$1000 oder \$9000 (4096 oder 36864) adressiert. Wollen wir nun unseren Bildschirm nach \$C000 (49152) legen, können wir den Zeichensatz in seiner ursprünglichen Form nicht mehr verwenden; wir müssen ihn verschieben. Das ganze Verfahren hierzu kann hier nicht geschildert werden, da es nahezu das gesamte Leserforum füllen würde. Nur soviel sei zu diesem Problem noch gesagt: bei der Benutzung eines Zeichensatzes im Speicher ist zu beachten, daß — die Startadresse durch 2048 teilbar ist, und

— der Zeichensatz im gleichen 16 KByte-Bereich liegt, wie der Bildschirmspeicher.

Ist beides der Fall, dann gilt:

Wert = (Zeichensatzadresse — Anfangsadresse des 16-KByte-Bereichs) / 1024

Durch

POKE 53272, PEEK(53272) AND 240 OR Wert

wird der Inhalt des Bildschirms mit diesem Zeichensatz ausgegeben. Rainer Wiesenfarth

Was kann die 1571-Floppy?

(1) Können eine 1541 und eine 1571 gleichzeitig am C 128 betrieben werden, um C 64-Dateien auf das 1571-Format zu überspielen?

(2) Ist das C 128 CP/M kompatibel zum C 64 CP/M, das heißt, können CP/M-Programme vom C 64 auch auf dem C 128 laufen?

(3) Wie können Dateien des C 64 CP/M auf das Format der 1571 gebracht werden?

(4) Kann man beim C 128 unter CP/M auch die alte 1541 benutzen?

Dietmar Hoffmann

(1) Natürlich, das ist kein Problem. Sie brauchen dazu nur die Gerätenummer eines der beiden Laufwerke von 8 auf 9 zu ändern.

(2) Im Prinzip ja. Das CP/M Plus des C 128 ist kompatibel zum CP/M 2.2, das auf dem C 64 mit CP/M-Modul läuft. Bei einigen Programmen (wie Textverarbeitung etc.) kann eine neue Anpassung mittels eines speziellen (immer mitgelieferten) Install-Programms notwendig werden.

(3) Einfach mit der 1571 eine Diskette im 1571-Format neu formatieren, dann die alte Diskette darauf kopieren — fertig.

(4) Das CP/M des C 128 läuft auch mit der guten alten 1541. Eine CP/M-Diskette im 1541-Format liegt jedem C 128 bei.

Maschinensprache wider Willen?

Die Joystickabfrage bei meinem C 128 macht mir Probleme. Mit der Anweisung »PRINT JOY(1)« kann ich zwar den Joystick sehr komfortabel per Basic-Befehl abfragen, aber wenn ich im Verlaufe des Programms auch nur ein einziges Mal den Feuerknopf gedrückt hatte, erscheint nach dem Ende des Programms ganz ohne mein Zutun der Befehl MONITOR, und der eingebaute Maschinensprachemonitor ist aufgerufen. Ich muß also jedesmal nach Beendigung meines Programms erst mit »X« aus dem Monitor heraus, um in den Direktmodus zu gelangen. Wie kann ich das verhindern?

Stefan Schulte

Der Joystick-Port 1 besitzt beim C 128 (wie auch beim C 64) einige gemeinsame Abfrage-Leitungen mit der Tastatur. Die Feuerknopf-Leitung ist dabei so unglücklich verschaltet, daß sie dem Betätigen der Funktionstaste F8 (MONITOR) entspricht. Abhilfe: Im Programm selbst sollte nach jeder Joystick-Abfrage ein GET-Befehl folgen. Versuchen Sie es also statt »X = JOY(1)« mit der Befehlsfolge »X = JOY(1): GET A\$«. Durch den GET-Befehl wird jede durch den Joystick ausgelöste »Tastenfunktion« unterdrückt. Eine noch einfachere Lösung ist allerdings die Benutzung von Joystick-Port 2. Dort treten solche Probleme nicht auf, weswegen auch die meisten Spiele den Port 2 benutzen.

DOS 5.1 mit Hypra-Load

Besteht die Möglichkeit, DOS 5.1 zusammen mit »Hypra-Load Perfekt« in ein EPROM zu brennen?

Dieter Bast

Ausgabe 10/85

Ich selbst habe beide Programme in ein EPROM vom Typ 2732 A gebrannt. Anzumerken wäre dazu, daß ich die beiden Programme nicht im Bereich ab \$8000 (wo EPROMs normalerweise eingeblendet werden) laufen lasse, sondern sie in den Original-Speicherbereich verschiebe (\$0800 beziehungsweise \$C000). Dies macht sich zeitlich kaum bemerkbar und es laufen so mehr Programme mit diesen Erweiterungen.

Wer eine Kopie dieses EPROMs haben möchte, möge mir bitte ein EPROM 2732 (A) mit selbstadressierten Freiumschlag zuschicken.

Info: Frank Giese, Overbergstr. 49, 5758 Fröndenberg

Wer kennt das PTS 100?

Vor einiger Zeit erwarb ich das Dialogterminalsystem PTS 100 der Firma Raytheon Data Systems. Der Anschluß des Gerätes oder zumindest des Monitors an den C 64 erwies sich jedoch als unmöglich, da ich keine Schaltungsunterlagen für das PTS 100 besitze. Telefonate mit der Firma Raytheon blieben erfolglos. Deshalb meine Frage an das Leserforum: Wer kann mir Hinweise für den Anschluß des PTS 100 an den C 64 geben oder Schaltpläne dazu besorgen?

Andreas Möllemann

Geheimnisvoller SYNTAX ERROR?

Wenn ich beim C 64 schreibe:
»SYNTAX ERROR ...« (beliebig
viele Punkte), dann antwortet
der Computer mit »0 0 0 0« (je
nach Anzahl der Punkte).

Stefan Thiem

Das Ganze ist weit weniger geheimnisvoll, als es auf den ersten Blick erscheinen mag. Bekanntlich entspricht das Fragezeichen beim C 64 ja dem PRINT-Befehl, so daß die genannte Zeile dem Computer erscheint als »PRINT SYNTAX ERROR ...«. Außerdem können beim C 64 auch lange Variablenamen verwendet werden, allerdings merkt sich der Computer immer nur die beiden ersten Zeichen eines Variablenamens. Wenn Variablenamen beliebig lang sein können (im Rahmen einer Basic-Zeile natürlich), woran erkennt dann der Computer das Ende eines Namens? Die Antwort ist einfach: Wenn der C 64 einen Variablenamen gefunden hat, dann merkt er sich die ersten beiden Zeichen dieses Namens und überliest alle weiteren Zeichen, bis entweder das Zeilenende, ein Doppelpunkt, eine öffnende Klammer oder ein Basic-Schlüsselwort gefunden wurde. Zu den Schlüsselworten gehören dabei auch alle mathematischen Operatoren wie »+«, »-« oder »/« — und auch das im Wort »Error« verborgene Schlüsselwort OR.

Zur näheren Erläuterung des oben genannten Phänomens habe ich meinen Redaktionskollegen Doktor Bobo gebeten, mir die Einrichtungen seines Software-Laboratoriums für ein Experiment zur Verfügung zu stellen. Wir können uns hier erstmals in der Geschichte der Home-Computer aus der Sicht des C 64 ansehen, was bei Eingabe der oben genannten Zeile passiert. Dazu bedienen wir uns eines Spezialgerätes aus Doktor Bobos Software-Laboratorium, des sogenannten »Computer-Telepathors«, der es ermöglicht, alle Gedankengänge des C 64 beim Bearbeiten von Basic-Programmen mitzuhören:

»Mein Gott, ist das langweilig. Jetzt warte ich schon zig Millionen Taktzyklen, das dieser Mensch endlich was auf meiner Tastatur eingibt. Das hat mir auch niemand an der Wiege gesungen, daß ich mal so einen langweiligen Job als Home-Computer bekommen würde (Gähnen ...). Doch halt, da regt sich was. Ich bekomme etwas zu tun! Eine Eingabe! Hoffentlich keiner dieser üblen POKEs, die mir manchmal so schlecht bekommen. Egal, gleich an die Arbeit ... Die ganze Zeile lautet »PRINT SYNTAX ERROR ...«. Schauen wir uns das mal genauer an. Zu-

nächst haben wir da die PRINT-Anweisung, abgekürzt durch das Fragezeichen. Aha, es soll also etwas ausgedruckt werden. Das muß ich mir merken. Aber was? Die nächsten beiden Zeichen hinter dem Fragezeichen sind »SY«. Es handelt sich dabei ganz offensichtlich um eine Variable. Also »SY« als Variablenamen merken und weitersehen. Es folgen noch weitere Zeichen hinter SY, also einer dieser unnützen langen Variablenamen ... nun ja, man kann sich seine Programmierer ja nicht aussuchen. Die weiteren Zeichen des Variablenamens interessieren mich nicht, daher halte ich bei den nächsten Zeichen nur nach einem Schlüsselwort Ausschau.

Also mal weitersehen: »NTAX«, »NTAX E«, »ERR« ... alles uninteressant. Doch halt, dann »ERR« folgt ein Schlüsselwort: »OR«. Endlich, da haben wir's. Der Variablenname war also »SYNTAX ERR« (nur »SY« merken) und es folgt das Schlüsselwort OR. Es soll also das Ergebnis der OR-Verknüpfung von »SY« und noch etwas ausgedruckt werden. Also sehen wir einmal weiter... ja, richtig, es folgt noch ein Zeichen, nämlich ein Punkt. Das kann nur eines sein, nämlich eine Zahl, die mit einem Dezimalpunkt beginnt. Schaffen wir also mal etwas Platz, um eine Zahl zu speichern und füllen wir diesen Platz zunächst nur mit einer Null. So, dann schauen wir mal, was hinter dem Punkt kommt. Noch ein Punkt, die erste Zahl muß also schon zu Ende sein. Welcher Wert war dafür noch eingetragen ... ach ja, eine Null. Gut, dann sind jetzt zwei Argumente für die OR-Verknüpfung vorhanden, nämlich die Variable »SY« und die Zahl Null. Also, das hätte dieser Mensch auch einfacher haben können. Warum schreibt er nicht »PRINT SY OR 0«, wie jeder normale Computer es machen würde. Naja, man muß Nachsicht mit ihm haben ... es ist ja nur ein Mensch. Also gut, was tut man nicht alles für seine täglichen Daten ... an die Arbeit. SY hat den Wert... Moment, eben nachschauen,...ach ja, den Wert Null. Also lautet die Aufgabe »0 OR 0«. Das ist einfach, das Ergebnis ist wieder Null. Was sollte mit diesem Rechenergebnis noch gleich geschehen? Ach ja, richtig, da war eine PRINT-Anweisung. Kein Problem, eben schnell eine Null als Ergebnis auf den Bildschirm schreiben und weiterschauen, ob noch etwas gedruckt werden soll.

Ja, richtig, da ist noch ein Punkt. Scheint sich wieder um eine Zahl zu handeln. Also einen Speicherplatz für diese Zahl reservieren und vorsichtshalber mal auf Null setzen. Was kommt

Leser fragen — Willi Brechtl antwortet Hallo liebe Leser, hier bin ich wieder, um Eure Fragen zu beantworten.

Ich werde mich hauptsächlich um Leserbriefe kümmern, die nicht in das sachliche Einerlei des Leserforums passen. Zum Beispiel Fragen, die sich aus dem einen oder anderen Grund nur ganz subjektiv beantworten lassen. Oft genug tauchen auch Probleme auf, die sich nicht mit einem kurzen Antwortsatz abhandeln lassen. Und wenn

selbst eine längere Antwort im Rahmen des Leserforums nicht mehr ausreichen würde, dann ist das ganz klar ein Fall für Willi Brechtl.

Also: Wenn Sie als Anfänger Probleme mit Computer, Software oder Handbuch haben, dann wenden Sie sich in Zukunft doch einfach vertrauensvoll direkt an mich.

dann? Wieder so ein Punkt. Die Zahl ist damit also schon zu Ende gewesen. Lächerlich einfach. Es ist gar nichts mehr zu rechnen. Einfach nur die Zahl drucken, also eine Null auf den Bildschirm. Dann weiter im Text. Es folgt wieder ein Punkt. Das kenne ich inzwischen schon. Für wie dumm hält mich dieser Mensch eigentlich? Na gut, der Klügere gibt nach. Hier drucke ich nochmal eine Null und meinetwegen noch so viele hinterher, wie Dezimalpunkte vorhanden sind ... diese Menschen sind alle ein bißchen verrückt ...»

An dieser Stelle hatte Doktor Bobo genug von diesem überheblichen Exemplar eines C 64 und drückte kurz entschlossen den Netzschalter. Immerhin, das Experiment war erfolgreich. Das Geheimnis der Zeile mit dem Syntax Error war gelöst. Für alle Beteiligten überraschend war die Erkenntnis, daß zumindest der am Test beteiligte C 64 eigenständige Überlegungen nicht nur zu seiner Aufgabenstellung, sondern auch zur Person seines Programmierers anzustellen in der Lage war. Sollten Sie, liebe Leser, ähnliche Erfahrungen mit Ihrem C 64 gemacht haben oder auf andere, merkwürdige Phänomene gestoßen sein, dann schreiben Sie doch bitte an Doktor Bobo in der Redaktion 64'er, Stichwort Software-Laboratorium.

Basic-Programm im \$C000-Bereich?

Ich möchte mein Datenverwaltungs-Programm (Basic) in den freien Speicherbereich ab Adresse 49152 (\$C000) verlegen mit den Befehlen »POKE 44,192 : POKE 56,208 : POKE 49152,0 : NEW«. Damit möchte ich erreichen, daß der gesamte normale

Basic-Bereich (Speicheradressen 2048 bis 40959) ganz mit den Daten der Dateiverwaltung belegt werden kann (um so mehr Speicherplatz zu erhalten). Leider funktioniert es nicht so ganz. (Helmut Hegeler)

Die Idee, den \$C000-Bereich für Basic-Programme zu nutzen, ist zwar auf den ersten Blick bestechend, aber leider so nicht durchführbar. Sie können zwar durch entsprechende POKE-Befehle das C 64-Basic dazu zwingen, das Programm in diesem Speicherbereich zu vermuten, aber Sie haben dabei eine Eigenheit des Basic-Interpreters übersehen. Der Variablenbereich muß nämlich immer hinter dem Programmtext im Speicher stehen. Wenn Sie dem C 64 durch entsprechende POKE-Befehle einen Variablenanfang vor dem Programmtext aufzwingen wollen, dann geraten einige interne Zeiger durcheinander. Also: Hände weg von derlei Experimenten.

Aber selbst wenn es so klappen würde, wie Sie es versucht haben, würden Sie maximal vier KByte zusätzlichen Speicherplatz für Ihre Daten gewinnen (also volle 38 KByte statt 34 KByte). Wenn Sie also mit 34 KByte reine Daten bei 4 KByte Programm nicht auskommen, dann sollten Sie sich Ihre Datenorganisation einmal genauer ansehen. Es gibt halt immer eine gewisse Obergrenze, was an Daten im Speicher stehen kann. Wenn Sie jetzt mit allen Tricks nochmals vier KByte mehr für Daten herausholen, dann hält Sie das wohl auch nicht viel länger über Wasser. Mein Ratschlag daher: Setzen Sie sich doch einmal mit relativen Dateien auseinander und ändern Sie Ihr Programm so ab, daß nicht der gesamte Datenbestand in den Speicher geholt werden muß.