Directory-Manipulationen II

Wollen Sie nicht, daß jemand unbefugt Ihre Programmsammlung benutzt oder kopiert? Schützen Sie doch einfach Ihr Directory! Wir zeigen Ihnen, wie man's macht.

In der 64'er, Ausgabe 6/85, berichteten wir zum ersten Mal über Manipulationen an Directories. Die Resonanz auf diesen Artikel war sehr groß. Es kamen sowohl neue Tips bei uns an, wie auch Verfahren zur Überwindung der schon vorgestellten Manipulationen. Beides wollen wir Ihnen nicht vorenthalten. Deswegen also einiges mehr über Directories und deren Manipulation. Alles, was Sie an Handwerkszeug benötigen, ist ein Diskettenmonitor wie beispielsweise den aus der 64'er, Ausgabe 8/85.

Mit dem Fragezeichen überlisten

Sehr ausführlich haben wir über Tricks berichtet, die mit dem Gänsefüßchen zusammenhängen. Ein Gänsefüßchen im Filenamen sorgt für Verwirrung, weil es sich ja nicht direkt mit dem LOAD-Befehl eingeben läßt. Floppy-Experte Karsten Schramm machte uns aber auf das Fragezeichen, das zweite, selten benutzte Jokerzeichen der 1541 aufmerksam. Will man ein mit Gänsefüßchen im Filenamen gesichertes Programm laden, dann benutzt man beim LOAD-Befehl einfach das Fragezeichen. Anstelle von »LOAD "64"CHR$(34)"ER",8« funktioniert also auch »LOAD "64?ER",8«. Ähnlich kann man natürlich vorgehen, wenn Grafikzeichen oder Steuercodes in den Filenamen untergebracht worden sind. Ein bekanntes Beispiel sind die auf Diskette gespeicherten Bilder des Koala-Painter mit dem reversen Pik im Filenamen. Auch hier leistet das Fragezeichen gute Dienste.

Endlose Directories

Die Ladezeit für ein Directory beträgt normalerweise nur einige Sekunden. Das ändert sich schlagartig, wenn man ein Directory unendlich lang macht. Wer ein solches endloses Directory mit dem LOAD-Befehl laden will und nicht nach mehreren Minuten entnervt die STOP-Taste drückt, der wird mit netten optischen Effekten und einem Systemabsturz belohnt. Spätestens dann, wenn der $D-Bereich (I/O-Bausteine, VIC, etc.) überschrieben wird, spielt Ihr Computer nicht mehr mit. Wer den Ladevorgang unterbricht, kann mit dem normalen LIST-Befehl das Directory nicht lesen, da die Basic-Zeiger nicht richtig gesetzt werden. Ein Maschinensprachemonitor oder ein RENEW (OLD) können die Filenamen aber sichtbar machen. Trotzdem ist der Verblüffungseffekt groß, wenn ein einfacher LOAD-Befehl, den man jeden Tag benutzt, zum Systemabsturz führt.

Doch wie erzeugt man ein endloses Directory? Benötigt wird nichts weiter als ein einfacher Disk-Monitor. Wenn Sie das Directory auf der Spur 18 mitverfolgen, stellen Sie fest, daß die beiden ersten Bytes des letzten Directoryblocks $00 und $FF lauten.

Ändern Sie einfach diese beiden Bytes auf »12 01« (hexadezimal). Der letzte Directoryblock zeigt nun auf den ersten Directoryblock. Das Directory befindet sich jetzt sozusagen in einer Endlosschleife: Ist es zu Ende, geht es gleich wieder von vorne los. Eine recht amüsante Sache, die übrigens auch bei professioneller Software häufig zu finden ist.

Ein Directory zieht um

Beim folgenden Trick könnte man auch von »multiplen« (mehrfachen) Directories sprechen, der Begriff »Umzug« ist allerdings viel anschaulicher. Den Effekt zu beschreiben, der beim Listen von umgezogenen Directories auftritt, ist praktisch unmöglich, da sich ein umgezogenes Directory durch fast nichts von einem normalen unterscheidet. Versucht man aber, eines der weiter hinten gelegenen Programme zu laden, beispielsweise das zehnte, dann passiert alles mögliche. Vom »FILE NOT FOUND ERROR« über den »OUT OF MEMORY ERROR« bis hin zum Systemabsturz. Allerdings lassen sich die ersten acht Files ganz normal laden und starten. Diese acht Files laden nun Programmteile nach, die entweder nicht von Hand geladen werden können, weil dann oben beschriebene Effekte eintreten, oder gar nicht im Directory stehen! Was ist passiert? Ein Blick mit dem Diskettenmonitor offenbart folgendes: Der erste Directoryblock (18,1) weist nicht auf den normalerweise zweiten Directoryblock (18,4), sondern auf einen anderen, beispielsweise 18,5. Trotzdem steht in 18,4 ein vernünftiger Teil des Directorys. Dieser wird normalerweise aber niemals gelesen. Das Ladeprogramm, das auf diese Teile des Directorys zugreifen will, muß vor dem Zugriff den Zeiger in 18,1 von 18,5 auf 18,4 ändern. Dann ist das Directory so, wie es sein sollte. Nach dem Zugriff wird der Zeiger dann wieder sofort auf das Dummy-Directory, das keinerlei Funktion außer der Verwirrung hat, zurückgestellt. Es existieren also zwei unterschiedliche Directories auf der Diskette, wobei das »echte« immer nur nach einer Vorbehandlung der Diskette erreichbar ist.

Wie man so ein Dummy-Directory erstellt? Nichts einfacher als das: Kopieren Sie mit einem Disketten-Monitor Directory-Blöcke von anderen Disketten auf freie Blöcke der zu schützenden Diskette und gleichen Sie die Zeiger an. Der Begriff »Umzug« ist deswegen treffend, weil man sich ja nicht nur auf die Spur 18 beschränken muß: Der Zeiger des ersten Blockes kann beispielsweise auf 1,1 zeigen, das Directory (oder der Dummy) geht also auf der Spur 1 weiter, wo man es nicht vermutet und somit auch mit einem Diskmonitor nicht so leicht findet (Wer sucht schon alle 683 Blöcke einer Diskette nach einem Directory ab?).

Wer jetzt besonders gemein zu seinen lieben Mitmenschen sein will, der kann das Directory zum Laufwerks-Killer umgestalten. Man kombiniert dazu den Umzugtrick mit dem des endlosen Directorys: Der erste Directoryblock verweist auf 1,1, dieser verweist auf 35,1 und der wieder zurück auf 1,1. Die Folge: Beim Ladeversuch des Directories

rast der Schreib-Lese-Kopf wie wild zwischen den Spuren 1 und 35 hin und her. Bis zu über zweihundert Mal kann das gehen, betätigt man vorher nicht die STOP-Taste. Danach erhält man dann wieder den schon bekannten Systemabsturz. Warnung! Mit dieser Methode kann man die Laufwerksmechanik überlasten und somit beschädigen!

Geheimnisvolles Leerzeichen

Mit einem Leerzeichen kann man ganz verrückte Sachen anstellen. Es geht hier allerdings nicht um das Leerzeichen, das man bei Druck auf die SPACE-Taste erhält, sondern um seinen nahen Verwandten: SHIFT-SPACE. Optisch lassen sich die beiden nicht auseinanderhalten. Nur durch ihre ASCII-Codes, 32 für das normale und 160 für SHIFT-SPACE, unterscheiden sie sich. Wenn es aber um Directories geht, dann erhält SHIFT-SPACE eine tragende Bedeutung, die es weit über das normale Leerzeichen hinaushebt. Um sich die fatalen Wirkungen von SHIFT-SPACE vor Augen zu führen, sehen Sie sich doch einmal das Directory in Bild 1 an. Das sieht alles so bekannt aus, werden Sie sagen. Richtig: Ähnliche Effekte lassen sich auch mit einem Gänsefüßchen als erstem Buchstaben im Filenamen erzielen. Dummerweise lassen sich diese Programme aber nicht mit »LOAD CHR$(34)"name",8« laden. Und sogar das Fragezeichen als Ersatz für den ersten Buchstaben versagt! Was geht hier vor?

Ein Blick mit einem Disk-Monitor offenbart das Geheimnis: Alle Filenamen beginnen mit SHIFT-SPACE. Normalerweise werden Filenamen, die kürzer als 16 Buchstaben sind, mit SHIFT-SPACE aufgefüllt. Findet das DOS beim Übertragen des Directory nun ein SHIFT-SPACE, denkt es, daß der Filename somit zu Ende ist und sendet ein Gänsefüßchen an den Computer. Nachfolgende Zeichen werden aber nicht ignoriert. Will man so gesicherte Files laden, dann muß der Befehl »LOAD" SHIFT-SPACE Filenamenrest",8« lauten. Das Fragezeichen als Joker arbeitet in diesem Fall nicht einwandfrei.

Der Trick funktioniert nicht nur, wenn SHIFT-SPACE der erste Buchstabe ist. Taucht es allerdings mitten im Filenamen auf, dann ist dieser tatsächlich an der entsprechenden Stelle zu Ende. Auf diesem Trick beruhen auch Directories, an deren Filenamen ein »,8,1« angehängt wurde. Dort folgen die entsprechenden ASCII-Codes einem SHIFT-SPACE.

Was man mit dem geheimnisvollen Leerzeichen noch so alles anstellen kann, sollten Sie mit einem Disk-Monitor mal selbst ausprobieren. Nur eines sei noch gesagt: Hinter SHIFT-SPACE im Directory werden Steuercodes nicht mehr in Grafikzeichen umgesetzt, sondern ausgeführt.

Steuercodes im Filenamen

Manche Leser wollten nach dem ersten Teil wissen, was man mit Steuercodes im Filenamen anfangen kann. Nach einem Gänsefüßchen oder SHIFT-SPACE im Filenamen werden alle gesendeten Steuercodes, deren ASCII-Werte unter 128 liegen, ausgeführt. Hier ein paar Beispiele:

— CHR$(18) — REVERS ON
— CHR$(19) — HOME
— CHR$(5) — Weiß
— CHR$(28) — Rot
— CHR$(30) — Grün
— CHR$(31) — Dunkelblau
— CHR$(20) — DELETE

Ein Anwendungsbeispiel: Ist ein Filename nur sieben Buchstaben lang, dann kann man ihn wirkungsvoll im Listing verschwinden lassen, wenn hinter den sieben Zeichen des Filenamens ein SHIFT-SPACE und dann acht DELETE-Codes folgen.

Weiterhin lassen sich im Filenamen per Disk-Monitor die beiden verbotenen Jokerzeichen »*« und »?« einbauen. Ihre ASCII-Codes sind dezimal 42 und 63.

So, damit dürften wir jetzt fast alle Möglichkeiten der Directory-Manipulation ausgelotet haben. Und gleich damit verbunden eine Bitte: Senden Sie uns keinesfalls Programme, die auf diese oder andere Arten geschützt sind. Wir können nur Programme bearbeiten, die LIST-fähig sind, sowohl Directory als auch Programm. (bs)

```
0 "DIRECTORY-MANIE" TEIL 2
12    ""64'ER          .   PRG
0     ""DAS MAGAZIN FUR   PRG
63483  ""COMPUTER-FANS     USR
256   ""DA GUCKSTE          REL
568 BLOCKS FREE.

READY.
```

Bild 1. So kann ein Directory aussehen, das mit den hier beschriebenen Verfahren manipuliert wurde.

Fortsetzung von Seite 153

Befehle, um damit optimal arbeiten zu können. Zusätzliche Treibersoftware wird damit überflüssig.

10. System

Mit dem System-Package können eine Vielzahl von speziellen Eigenschaften des C 64-Systems angesprochen werden.

11. Font

Mit einfachen Befehlen kann der Zeichensatz des C 64 geändert, geladen oder gespeichert werden. Es ist möglich, einen speziellen Zeichensatz fest mit einem Comal-Programm zu verbinden. Wird dieses Programm gespeichert, so wird auch der zugehörige Zeichensatz ohne weiteres Zutun automatisch mit abgespeichert. Bei jedem Laden eines derart präparierten Programms steht dieser Zeichensatz dann wieder zur Verfügung.

Fazit

Die hervorragenden Eigenschaften dieser neuen Programmiersprache lassen auf eine Durchsetzung auf breiter Front hoffen. Comal ist wie kaum eine andere Sprache dazu geeignet, strukturiertes und übersichtliches Programmieren interaktiv, im Dialog mit dem Computer zu lernen. Comal nimmt dem Benutzer im Gegensatz zu anderen Sprachen so leicht nichts übel, im Gegenteil: offensichtliche Fehleingaben werden vom System automatisch korrigiert. Der eingebaute Syntax-Check bereits bei der Eingabe von Programmzeilen, ist nicht nur für den Anfänger eine große Hilfe. Ausführliche Fehlermeldungen lassen den Benutzer nicht im unklaren, was er falsch gemacht hat. Die hohe Ausführungsgeschwindigkeit von Comal-Programmen ermöglicht Anwendungen, die beim C 64 sonst nur durch Maschinensprache zu realisieren sind. Der Komfort bei der Programmierung in Comal ist für C 64-Verhältnisse so enorm, daß man sich an einen anderen Computer versetzt glaubt. Alle, wirklich alle, Unzulänglichkeiten des C 64 sind durch Comal 80 behoben. Und wenn jemand doch noch etwas finden sollte — kein Problem, denn Comal ist durch das Package-Konzept flexibel genug, vom Anwender fast beliebig erweitert werden zu können. Mit einem Wort: Comal ist nicht eine, sondern die Programmiersprache für den C 64.

(S. Bauer/ev)

Comal 80 gibt es derzeit nur als Steckmodul für den C 64 bei Commodore Data a/s, Bjerrevej 67, DK-8700 Horsens, Dänemark. Preis zirka 1000 DKr. (etwa 300 Mark).