

ZCPR – das neue Kleid für CP/M

Komfortable Verbesserung für CP/M

Trotz allem Unken aus der MS-DOS-Ecke ist und bleibt CP/M ein Standard, der sehr weit verbreitet ist. Und wie bei allen Dingen ist auch hier die Entwicklung nicht stehengeblieben.

Das Betriebssystem CP/M hat, wie andere Programme auch, so seine Unzulänglichkeiten. Um sie auszubügeln, wurde ZCPR, im Heft 6-7/91 bereits vorgestellt, entwickelt. Im folgenden noch einige nützliche Hinweise zu dessen Anwendung.

CP/M ist wie MS-DOS dreigeteilt:

- 1). Der maschinenabhängige Teil, das BIOS (Basic Input/Output System); hier wird dem Gerät mitgeteilt, wie es sich verhalten soll.
- 2). Das Software-Interface zwischen Maschinenteil und Programm, das BDOS (Basic Disc Operation System).
- 3). Das Software-Interface zwischen dem Anwender und dem Programm, der CCP (Command Console Processor).

Historisch bedingt wurde CP/M im 8080-Code generiert, der langsamer und umständlicher ist als der Z80-Code. Da man davon ausgehen kann, daß die meisten CP/M-Computer heute einen Z80 oder höhere Verwandte wie den HD 64180 oder gar den Z280 verwenden, war es logisch, überflüssigen Code durch effizienteren zu ersetzen. Erstes Ziel war natürlich der CCP, da

dieser am leichtesten ersetzbar ist und zur Laufzeit sowieso per Definition vom Anwenderprogramm überschrieben werden darf. Bei jedem Control-C (Warm-Boot) wird er deshalb wieder nachgeladen. Als Nebeneffekt wird die Diskverwaltung "zurückgesetzt".

Richard Conn und Jay Sage sind die bedeutendsten Autoren des Z80 Console Processor Replacements, des ZCPR. Auch für 8085-Anwender kein Grund, traurig zu sein – der Prozessor NSC-800 ist Pin-kompatibel und kennt sich mit Z80-Instruktionen aus! Entsprechende Informationen kann jeder Triumph-Adler-Club geben; im Zweifelsfall einfach an die genannte Adresse wenden.

Bis zur Version 3.3 wurde ZCPR als nicht-kommerzielles Produkt über die Public Domain angeboten und ist nach wie vor (als Quellcode zur Selbstinstallation) erhältlich. Diese Versionen funktionieren ausschließlich unter CP/M 2.2! Für CPCs mit Dobbertin-Erweiterung gibt es eine fertig angepaßte Version gegen Zusendung einer Diskette mit 704 kByte Kapazität (Adresse siehe Box). Für andere Nutzer gibt es zwei Auto-Install-Versionen, NZCOM (für CP/M 2.2) und Z3PLUS

(für CP/M Plus). Beide sollen innerhalb von 15 Minuten lauffähige Systeme sein. An den Feinheiten kann man später feilen. Das Programm zum Preis von 126,61 DM, inklusive Versandkosten, beinhaltet einen Grundstock an Software mit zugehörigen Tools und ein deutsches Handbuch.

Jeweils neu erscheinende Tools bringt die Schneider/Amstrad CPC User Group gratis in Umlauf, die Verteilung übernehmen die User untereinander. Jeder Käufer erhält eine Anwender-Adreßliste, aus der er die Formate ershen kann, die vom Empfänger gelesen werden können. Das ist notwendig, weil diese Systeme auf so gut wie jedem CP/M-Computer laufen.

Gehen wir also davon aus, daß Ihr Z-System bereits läuft und daher ein definiertes Environment existiert. Das bedeutet, das Z-System verwaltet die Daten der Bildschirmsteuerung (TCAP ähnlich wie bei UNIX) und die Daten der gültigen Laufwerks- und User-Nummern. Eingeschlossen sind dabei auch die Zugriffs-Flags, die Suchpfad-Definition und die Namen der benannten Verzeichnisse. Alle für das Z-System geschriebenen Programme können diese Daten (mehr oder weniger gut, je nach Tool) auswerten. Auf diese Art und Weise können sogar Pull-down-Windows und Balkengrafiken portierbar programmiert werden! Damit hat man endlich etwas erreicht, was uns die MS-DOSler immer voraus hatten – eine einheitliche Umgebung (englisch = environment)!

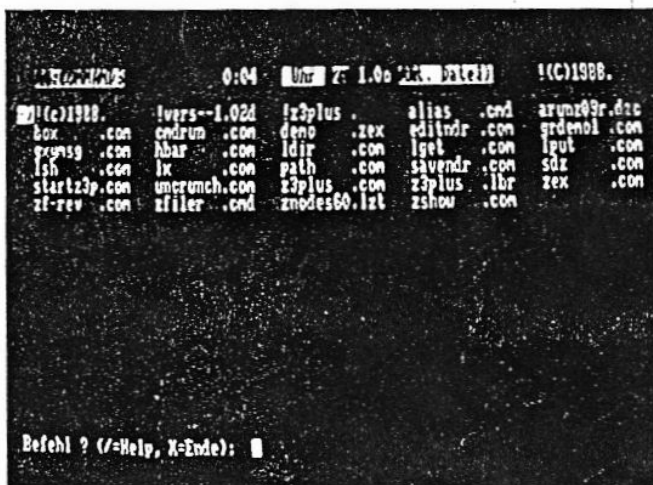
Hier ein Beispiel:

Ein Anwender hat die Hardware-Konfiguration Floppy A: und Floppy B: sowie eine RAM-Disk (eine Hard-Disk würde gegebenenfalls die Rolle der RAM-Disk übernehmen). Da wir oben Z3PLUS als Beispiel genommen haben, wollen wir dabei bleiben, denn für die CP/M-2.2-Variante NZCOM des Autoinstallers gilt fast das gleiche wie für die CP/M-Plus-Variante.

Nach dem gewohnten Booten des CP/M Plus starten wir Z3PLUS einfach wie ein Utility:

Z3PLUS <ENTER>

Die User-Bereiche, die der Anwender zum Arbeiten benötigt, wurden bereits vorher festgelegt und ihre Namen aus einer Datei geladen. Diese stehen jetzt als benannte Verzeichnisse zur Verfügung. Hier erkennt man, wozu User-Bereiche sinnvoll verwendet werden können. Die Übersicht auf großen Disketten und speziell auf Festplatten nimmt erheblich zu. Im Gegensatz zu MS-DOS ist



Ein Dateiverzeichnis unter ZFILER

jedoch eine hierarchische Schachtelung nicht vorgesehen, da es fast keine CP/M-Programme gibt, die ein derartiges Verzeichnis unterstützen.

User-Bereiche erhalten einen Namen

Sind mehrere Laufwerke oder User-Bereiche vorhanden, ist es manchmal schwierig, eine Datei zu finden oder aus einem anderen Bereich aufzurufen. Einfacher geschieht dies mit einem sogenannten SUCHPFAD. Für CP/M Plus ist neu, daß anstelle des normalen Suchpfades die Verwendung von User-Angaben zulässig sind. Der CP/M-2.2-Anwender mußte auf solchen Luxus bisher verzichten. Der PATH-Befehl dient zum Errichten und Anzeigen des Suchpfades (siehe Textkasten).

Beispiel für eine praktische Anwendung:

1. DBASE.COM, DBASEOVR.COM und mehrere .CMD-Dateien sollen nach Laufwerk M0: kopiert werden.
2. Eine Diskette in B: einlegen und, wenn vorhanden, alle .C?D-Dateien und alle ADR*.LBR nach M0: kopieren.
3. Alle komprimierten Dateien sollen entkomprimiert und die in Libraries (.LBR) abgelegten Dateien ausgepackt werden, damit unter DBASE damit gearbeitet werden kann.
4. DBASE soll viel Speicher zur Verfügung gestellt bekommen; während seines Ablaufs und anschließend sollen sofort wieder alle ZCPR-Extras zu Verfügung stehen.
5. Nach Beendigung der Arbeit eine Daten-Diskette in B: einlegen und sowohl die Datenbank als auch die veränderten CMD-Programme, die für DBASE nötig waren, als Backup speichern.

Ganz schön aufwendig, wenn man das ohne ZCPR realisieren will! Natürlich geht einiges über die raffinierte SUBMIT-Steuerung, was aber gerade unter

CP/M 2.2 einen Schreibzugriff auf A0: auslöst, wenn das Standard-Submit verwendet wird.

Automatisierung mit ARUNZ.COM beziehungsweise CMDRUN.COM

Unter ZCPR genügt zur Ausführung ein Befehl, der nur aus einem Buchstaben oder einem Wort besteht, solange der Befehl eindeutig ist. Wie geschieht das? Die Kommandozeile wird analysiert und über den Suchpfad ein passender Befehl A.COM beziehungsweise ADRES.COM gesucht. Da dieser nicht aufzufinden ist, tritt der ALIAS-Prozessor ARUNZ in Aktion, den wir in CMDRUN.COM umbenannt haben. Dieser sucht die Datei ALIAS.CMD im Heimat-User und dort den passenden Befehl. Die zugehörige, zusammengesetzte Kommandozeile wird interpretiert und ausgeführt. Für die obigen Operationen sieht das etwa so aus (Ausschnitt aus ALIAS.CMD - einer Textdatei(!) des Anwenders):

```
A=ADRES
ACOPY $*"1M1J1GBitte DBASE-Diskette einlegen und ENTER! "
; B0:DBASE*.COM RAM: /d
; if exist B0:*C?D;ACOPY
B0:*C?D RAM::else;ECHO Keine
*C?D Dateien gefunden;fi;
IF exist B0:*LBR;ACOPY B0:*LBR
RAM: /d;else;ECHO Keine Dateien
.LBR gefunden;fi;UNLIBM;UCRAM;
Z3PLUS small; DBASE
adres;Z3PLUS large;echo "Backup-
Disk einlegen und Taste! ";SAK;
UNLIBM
M;;W lget *.lbr **;$D$U
URAM
M;;uncrunch *.?Z?;PROTECT
*.CMD;$D$U
```

Das liest sich jetzt sehr kompliziert, im Grunde ist aber die Syntax nicht komplizierter als etwa in BASIC. Ein solches Mini-Menü kann jeder Anwender aus den reichhaltigen Beispielen innerhalb kürzester Zeit zusammenstellen und

dann immer wieder verwenden. Die Fachleute unter Ihnen haben bestimmt nicht nur die Leistungsfähigkeit des Systems erkannt, sondern schon gemerkt, daß viele "ALIASse" in einer einzigen Datei Platz finden (bei den kleinen AMSTRAD-3-Zoll-Disks oft ein Problem!) und daß ein gegenseitiger Aufruf von ALIAS-Befehlen möglich ist.

Alles, was der Anwender in unserem Beispiel eingeben muß, ist A oder ADRES und <ENTER>. Er sieht danach nur noch die Meldungen

"Bitte DBASE-Diskette einlegen und ENTER!"

sowie die Lademeldungen und danach die Aufforderung "Backup-Disk einlegen und Taste!" und "- Strike any key -".

Darüber hinaus kann jemand, der lieber menügesteuert arbeitet, mit dem Programm MENU.COM arbeiten, dessen Scripts leichter erlernbar sind. Hochinteressant sind die Utilities VLU.COM und ZFILER.COM. VLU dient zum Bearbeiten sogenannter LIBRARY-Dateien, also Sammeldateien, die eine Anzahl von anderen Dateien beinhalten. Man erkennt sie durch die Endung .LBR. Erzeugt werden sie unter CP/M mit NULU oder unter ZCPR eben auch mit VLU. NULU ermöglicht das Lesen gesqueezter Dateien, VLU das Lesen von gecrunchten Dateien innerhalb von Libraries, ohne diese auspacken. Unkomprimierte COM-Dateien kann man aus Libraries heraus ohne Auspacken aufrufen.

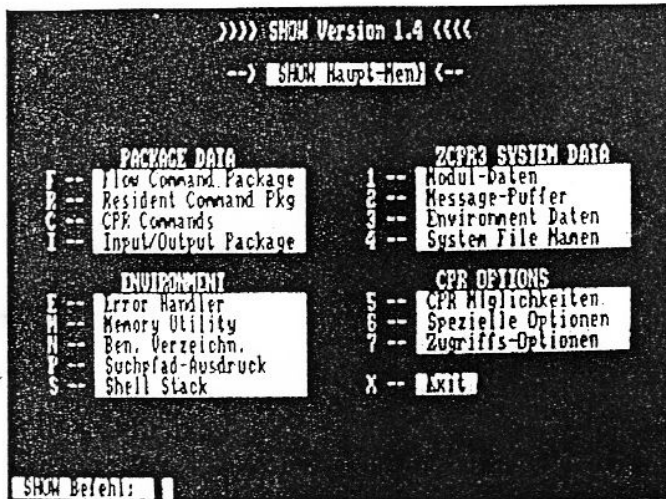
Ein Vorteil von VLU ist die Fähigkeit, bei geöffneter Library auf ein anderes Laufwerk zuzugreifen und dorthin die Dateien "mitzunehmen", um sie dann auszupacken. Mit ESC schaltet man den Pfeil-Cursor zwischen dem Dateiverzeichnis der Diskette und dem Inhalt der Library.

ZFILER ist wohl das ausgefeilteste Tool der Z-Systeme nach ARUNZ. Es erlaubt ebenfalls komplexe Scripts, die jedoch über eine Makro-Taste wie etwa ESC-A (bis Z) direkt ausgelöst werden können. Dabei wird ein Zeiger mit den Cursor-Tasten über das Inhaltsverzeichnis bewegt und der damit indizierte Dateiname in einem Puffer abgelegt. Diese Dateien werden dann beim Aufruf eines solchen Makros als Parameter angenommen.

Tritt ein Fehler auf, den das DOS bemerkt, so werden entsprechende ZCPR-Register gesetzt. Diese können von anderen Programmen oder Programmteilen gelesen und übernommen oder auch modifiziert werden. ZCPR-

A0:3ZOLL>PWD<ENTER>			
PWD, Version 1.0			
DU: DIR Name	DU: DIR Name	DU: DIR Name	DU: DIR Name
A 0: 3ZOLL	A 1: NOTIZEN		
B 0: UT	B 1: UT1	B 2: UT2	B 3: UT3
B 10: MAGAZINE	B 15: HELP		
M 0: RAM	M 1: BASIC	M 2: GRAFIK	
M 5: MODEM	M 10: TEMP	M 14: DOCS	M 15: BASE
A0:3ZOLL> (hier ist wieder unsere Kommando-Zeile!)			

Die Liste der Verzeichnisnamen



Das Hauptmenü von ZSHOW

Programme haben direkten Zugriff auf diese Register, auch ohne daß das DOS dies bemerkt. Hier hat man auch die Möglichkeit, seine eigenen, beispielsweise deutschen Fehlermeldungen auszugeben. Abhängig vom Erfolg oder Mißerfolg eines Befehls wird der IF-Status des Flow-Command-Prozessors, TRUE oder FALSE, wenn dies abgefragt wurde. Hierdurch läßt sich ein Mißbrauch oder eine ungültige Behandlung unterbinden.

ZCPR-Programme sind universell einsetzbar

Ein ausgefeilter Spezial-SUBMIT namens ZEX rundet die vom Anwender programmierbaren Hilfs-Utilities ab und ergänzt sie um wichtige automatische Input-/Output-Möglichkeiten, die durch ein Software-Interface zum ZCPR ausgewertet werden können. Hier kommt die IF-ELSE-FI-Steuerung perfekt zum Einsatz.

Ein weiteres Plus von ZCPR besteht darin, daß einmal an die Adresse HEX 100 geladene COM-Dateien mit GO noch einmal aufgerufen werden können, ohne daß sie dazu vom Laufwerk geladen werden müssen. Das kann ganz erheblich Zeit sparen. Außerdem kann sogar dazwischen die Diskette gewechselt werden, solange das auszuführende Programm ohne Overlays arbeitet. Die neuen Möglichkeiten des erweiterten SUBMIT-Prozessors ZEX sind ein zusätzlicher Pluspunkt. Sprungmarken und Rekursion sind die herausragendsten Merkmale dabei. ZCPR3 und höher hat vier sogenannte SHELL-Stacks. Das sind Puffer, in denen die Namen und Adressen von "auf Eis gelegten" Dateien enthalten sind. Bei Bedarf können diese in der entsprechenden Reihenfolge aktiviert werden. Mit

ZSHOW können Sie sich eine solche SHELL-Seite ansehen.

Der Shell-Stack erleichtert die Arbeit

Wie auf einem Kartenstapel werden Programme hier abgelegt und wieder abgeholt. Man kann also aus WordStar (zumindest aus der Version 4.0) den ZFILER aufrufen und aus diesem dann

VLU. Ruft man statt dessen SHOW auf, kann man sich das Ergebnis im SHELL-Stack ansehen. Nach dem Verlassen des momentan aktiven Programms wird anstelle des Betriebssystems das darunterliegende Programm wieder aktiv und so weiter, bis wir wieder in WordStar sind. Durch die übersichtliche Darstellung mit SHOW wird klar, wie so ein Shell-Stack arbeitet. Aber auch alle anderen wesentlichen Systemeigenschaften lassen sich gut erkennen.

Viele Anwendungen für CP/M existieren auch in speziellen ZCPR-Versionen, die auch die Eigenschaften der Register und Pfade mit berücksichtigen. Insgesamt sind seit dem Januar 1988 etwa 30 MByte an Public-Domain- und Shareware-Utilities produziert worden, die alle über die Adresse der SCUG bezogen werden können.

Helmut Jungkunz/jg

Eventuell auftretende Fragen richten Sie bitte an:

Schneider/Amstrad CPC User Group
c/o Helmut Jungkunz
Zacherlstr. 14
8045 Ismaning

```
A0:3ZOLL > path
SETPATH Version 1.0                                ;ab hier Zusatzkommentar

Symbolic      DU      DIR      ;Form der Darstellung
M15:          M15:    BASE:    ;Heimatbereich dieses ZCPRs
B0:           B0:     UT:      ;Suche auf B0: fortsetzen
M0:           M0:     RAM:     ;oder auf der RAM-Disk User 0
$0:           A0:     3ZOLL:   ;aktuelles Laufwerk, User 0
M15:          M15:    BASE:    ;notwendiger Abschluß Heimat

A0:3ZOLL >
Da in diesem System ein anderer Heimat-User als A0: eingetragen ist, können die Disketten in
A: und B: beliebig gewechselt werden.
```

Der PATH-Befehl zum Errichten und Anzeigen des Suchpfades

```
!QUIET!                                           ;; keine Meldung ausgeben
shv /q $1 $2 $3 $4 $5 $6 $7 $8                 ;; Parameter-Dummys in SHELL-VAR
!RING WAIT!                                       ;; piepst und wartet auf ENTER
:= START                                          ;; Sprungmarke START (Label)
!SAY!                                             ;; Beginn einer Textausgabe Guten Tag!
Sie können jetzt mit dem Kopieren be-
ginnen, bitte die Diskette 1 #
einlegen! == > Abbruch mit ENDE
möglich < ==
!ENDSAY!                                         ;; Ende der Textausgabe
!WAIT!                                           ;; wartet auf Tastendruck
:= KOPIEREN < M::SDZ                             ; automatische Eingabe
!SAY!                                             Welche Datei nach B: Kopieren?
!ENDSAY!                                         ; Hole $1 IF error 7 goto ERROR7
!UNTIL! $1 B:                                     ;; ERROR7 = Disk Full COPY
                                                ;; warte auf Enter und kopiere $1 goto
                                                Kopieren
                                                ;; Kopiere $1 nach B: bis Disk Full

:= ERROR7
!SAY! Wieso legen Sie eine randvolle Diskette ein? !ENDSAY!
goto START und so weiter
```

Ein ZEX-Script