

# Z3PLUS

Das Z-System für CP/M-Plus

## User's Manual

von

Jay Sage und Bridger Mitchell

© Copyright 1988  
Jay Sage und Bridger Mitchell

Plu\*Perfect Systems  
410 23rd Street  
Santa Monica, CA 90402

## Copyright Vermerk

Z3PLUS — Das Automatische, Dynamische Z-System für CP/M-Plus Computer — ist © Copyright 1988 Joe Wright, Alle Rechte vorbehalten. Das Z3PLUS User's Manual ist © Copyright 1988 - Jay Sage und Bridger Mitchell. Der ZCP834 Kommando-Prozessor ist © Copyright 1988 - Jay Sage, Alle Rechte vorbehalten. JETLDR.COM ist © Copyright 1988 Bridger Mitchell, Alle Rechte vorbehalten.  
Deutsche Übersetzung Copyright 1990 Helmut Jungkunz.

Ihre Einzel-Lizenz gestattet Ihnen, Z3PLUS auf Ihrem/(n) Computer(n) einzusetzen, Sicherheitskopien für eigene Verwendung anzufertigen und Z3PLUS anderen vorzuführen. Von geschützten Z3PLUS-Dateien oder von diesem Manual dürfen keine Kopien für andere angefertigt werden. Das verletzt unser Copyright und verstößt gegen die Lizenzvereinbarung mit uns.

Durch Copyright geschützte Z3PLUS Dateien sind: Z3PLUS.COM, Z3PLUS.LBR, JETLDR.COM, DEFPC3.ZRL, TIMECP3.ZRL.

Z-System Tools und andere Dateien sind auf der Z3PLUS Diskette als Zugeständnis an die Z3PLUS-Lizenz. Viele dieser Dateien stehen unter Copyright der jeweiligen Autoren und dürfen ausschließlich für nichtkommerziellen Gebrauch kopiert werden.

## Haftungs-Ausschluß

Die Autoren übernehmen keinerlei Garantien bezüglich des Inhalts und schließen besonders jede Art von stillschweigender Garantie für die Marktfähigkeit oder Eignung für einen bestimmten Zweck aus. Dieses Handbuch und die begleitende Software werden verkauft "so wie sie ist" und der Autor wird in keinem Fall verantwortlich sein für direkte, indirekte, zufällige, oder in der Folge auftretende Schäden, die aus irgend einem Defekt, Irrtum oder Funktionsfehler herrühren.

Es wird jedoch von den Autoren eine Garantie auf einwandfreie Beschaffenheit, auf Material und Kopierqualität der Disketten gewährt, auf denen das Z3PLUS Paket geliefert wurde. Dies bezieht sich auf eine Dauer von neunzig (90) Tagen vom Verkaufsdatum an, bei normalem Gebrauch, sofern eine glaubhafte Kopie des Kaufbelegs vorliegt.

## Würdigungen

Das Z3PLUS System wurde getestet und verbessert durch Anregungen von Mark Allen, Cam Cotrill, Dick LeBleu, Frank Prindle, Jay Sage und Joe Wright.

Al Hawley, der Sysop des Ladera Z-NODE #2 stellte Konferenzmöglichkeiten. George Borys von der Bay Area Micro Decision User's Association stellte einen MD-5 Entwicklungsrechner zur Verfügung.

Das Z-System ist eine sich entwickelnde Betriebs-System-Umgebung für Z80-kompatible Computer, zu der viele beigetragen haben. Die Entwicklung von Z3PLUS hat besonders profitiert durch die Arbeit von:

Michal Carson	-	VLU
Richard Conn	-	ZCPR v 3.0, HELP, TCSELECT
C. B. Falconer	-	LT
Rob Friefeld	-	SALIAS
Steven Greenberg	-	CRUNCH, UNCRUNCH, LDIR
Al Hawley	-	EDITNDR, SAVENDR
Jim Lopushinsky	-	CCP105
Bridger Mitchell	-	Z3PLUS, JETLDR
Bruce Morgen	-	LPUT, LGET
Rob Friefeld	-	LSH, ZERRLSH
Jay Sage	-	ZCPR v 3.3, v 3.4, ARUNZ, ZFILER, ZSHOW
Joe Wright	-	NZCOM

## Warenzeichen

Z3PLUS, DosDisk (Bridger Mitchell),  
CP/M, CP/M-Plus (Digital Research, Inc.),  
WordStar (WordStar Corp.)



## VORWORT

Automatisch, universell, dynamisch: zwei neue Z-Systeme — NZ-COM für CP/M-2.2 Computer und Z3PLUS für CP/M-Plus Computer — sind das Resultat der extensiven, vereinten Bemühungen von Joe Wright, Bridger Mitchell, und Jay Sage. Alle drei Autoren arbeiteten zusammen an vielen Aspekten beider Produkte; jeder war besonders verantwortlich für eins der essentiellen Charakteristika.

Mit Joe Wright kam die Automatische Funktion. Er war es, der zuerst das Z-COM konzipierte und damit bewies, was viele als unmöglich abgetan hatten, — eine Version des Z-Systems, das sich auf fast jedem CP/M-2.2 Computer selbst automatisch installieren würde. Aber immer noch, sogar nach dem Erfolg des Z-COM, schien es, als ob das Z-System niemals auf einem CP/M-Plus Computer mit seinem völlig anderen Kommando-Prozessor und dem gebankten Speicher laufen könnte. Z3PLUS beweist das Gegenteil.

Bridger Mitchell machte die Systeme universell. Seit seiner ersten Konfrontation mit dem Z-System, wertete er den komplexen, arbeitsintensiven Aufwand ab, der erforderlich sein würde, um jedes Code Modul mit jeder beliebigen Implementation des Systems laufen lassen zu können. Er entwickelte ein universelles ZRL-Dateiformat und einen Lader, der es einem einzelnen File erlaubte, sich an jedes beliebige Z-System anzupassen, ungeachtet dessen Konfiguration in Bezug auf Modul-Adressen und Größe.

Jay Sage fügte die Dynamik hinzu. Er entwarf ein Betriebs-System, das seine Größe und seinen Charakter — inmitten einer Befehlszeile, wenn es sein mußte — wandeln konnte, um sich an die Gegebenheiten einer speziellen Anwendung anzupassen. Der Anwender muß nicht mehr mit einem herben Kompromiß leben zwischen Betriebs-System-Features und Speicher-Belegung. Der jeweilige Nachteil kann zu jeder Zeit geändert werden.

Diese drei Autoren spielten die maßgebliche Rolle bei der Entwicklung der neuen Systeme und sind gleichermaßen verantwortlich für fundamentale Entwicklungen, die das neue Z-System beinhaltet, — die ZCPR Version 3.4 des Kommando-Prozessors (Jay Sage), den Z3PLUS-Lader und Kommando-Prozessor (Bridger Mitchell), sowie

den Z3PLUS-Lader (Joe Wright). Aber das ganze Stück schloß viele andere Darsteller ein, ja eine ganze Gemeinschaft von Ihnen. Es ist diese Rollenverteilung in der laufenden Entwicklung des Z-Systems im allgemeinen, was den Aufwand so befriedigend und lohnend für uns alle macht.

# Inhaltsverzeichnis

<b>1</b>	<b>Was ist Z3PLUS?</b>	<b>1</b>
1.1	Die Vorteile von Z3PLUS . . . . .	1
1.2	Die Eigenschaften des neuen Kommando-Prozessors . . . . .	2
1.3	Andere Haupt-Eigenschaften . . . . .	3
1.4	Zum Gebrauch des Handbuchs . . . . .	3
<b>2</b>	<b>Erstes Arbeiten mit Z3PLUS</b>	<b>5</b>
2.1	Einleitung . . . . .	5
2.2	Das Zusammenstellen der Dateien . . . . .	6
2.3	Der Betrieb von Z3PLUS . . . . .	8
2.4	Die Organisation Ihrer Dateien . . . . .	8
<b>3</b>	<b>Z3PLUS kennenlernen und verstehen</b>	<b>11</b>
3.1	Wie man einen Alias erzeugt . . . . .	11
3.2	Welche Befehle gibt es? . . . . .	13
3.2.1	RCP Kommandos . . . . .	14
3.2.2	FCP Kommandos . . . . .	15
3.2.3	CPR Commands . . . . .	19
3.2.4	Transiente Kommandos . . . . .	21
3.3	Mehr über die System-Konfiguration . . . . .	22

3.4	Inkompatible Programme . . . . .	23
<b>4</b>	<b>Mehr vom Z-System</b>	<b>25</b>
4.1	Residente Z3PLUS Komponenten . . . . .	25
4.2	Automatische Kommandos . . . . .	26
4.2.1	Erweiterter Kommando-Prozessor . . . . .	26
4.2.2	Error Handler . . . . .	27
4.2.3	Kommando Shells . . . . .	28
4.3	Andere Z-System Tools . . . . .	33
4.3.1	ARUNZ . . . . .	34
4.3.2	HELP . . . . .	36
4.3.3	Library Tools . . . . .	37
4.3.4	Datei-Kompression . . . . .	38
4.3.5	Benannte Directory Tools . . . . .	38
4.3.6	Andere Tools . . . . .	40
4.4	Kommando-Hierarchie . . . . .	40
4.4.1	Kommando Akquisition . . . . .	40
4.4.2	Kommando Auflösung . . . . .	41
<b>5</b>	<b>Wie man mehr aus Z3PLUS herausholt</b>	<b>43</b>
5.1	Alternative Befehle zum Aufruf von Z3PLUS . . . . .	43
5.1.1	Alternative Lademethoden . . . . .	43
5.1.2	Alternative Default Systeme . . . . .	44
5.2	Fliegender Wechsel der Systeme . . . . .	45
5.3	Anpassen Ihres Z3PLUS Systems . . . . .	48
5.3.1	Vorübergehende Änderungen . . . . .	48
5.3.2	Sonderanpassungen . . . . .	49
<b>6</b>	<b>Technisches Nachschlagwerk</b>	<b>55</b>



6.1	Definition von Datei-Typen . . . . .	55
6.2	Bei Z3PLUS mitgelieferte Dateien . . . . .	55
6.2.1	Z3PLUS System-Dateien . . . . .	56
6.2.2	Tools und Utilities . . . . .	56
6.3	Z3PLUS Befehls-Zeilen . . . . .	59
6.3.1	Help Seiten . . . . .	59
6.3.2	Laden des Default-Systems . . . . .	59
6.3.3	Das Entfernen von Z3PLUS . . . . .	60
6.3.4	Das Laden spezifischer System-Moduln . . . . .	61
6.4	Das Programm JetLDR . . . . .	62
6.5	Patchen von Z3PLUS.COM . . . . .	63
6.6	Unterschiede zwischen Z3PLUS und CP/M-Plus . . . . .	63
6.6.1	Das GET Kommando . . . . .	63
6.6.2	IOP — Input/Output Package . . . . .	64
6.6.3	Public Files . . . . .	64
6.6.4	Paßwörter . . . . .	64
6.6.5	Der Kommando-Suchpfad . . . . .	65
6.6.6	Mehrfach-Kommandos . . . . .	65
6.6.7	Konditionelle Ausführung . . . . .	65
6.7	Funktionale Theorie . . . . .	66
<b>7</b>	<b>Bibliographie</b>	<b>69</b>
7.1	Z-Nodes . . . . .	69
7.2	<i>The Computer Journal</i> . . . . .	70
7.3	Andere Publikationen . . . . .	71



# Kapitel 1

## Was ist Z3PLUS?

Ihr CP/M-Plus-Computer wird normalerweise durch ein Hauptprogramm gesteuert: — das CP/M-Plus Betriebssystem. Das Betriebssystem Ihres Computers ist sein Hauptprogramm, das vom Augenblick des Einschaltens vorhanden ist. Es interpretiert Befehle, lädt Programme, führt Sie aus und koordiniert die Dateien und Verbindungen zu Ihrem Terminal, Drucker, und Modem.

Z3PLUS stellt eine wesentliche Bereicherung zum Standard CP/M-Plus Betriebssystem dar. Es ermöglicht Ihnen, mit vollständig neuer Software zu arbeiten, die im wesentlichen die gleiche ist, wie die der weitverbreiteten, beliebten ZCPR3 Systeme oder der Z-Systeme für CP/M-2.2 Computer. Gleichzeitig bewahrt Z3PLUS die maschinennahe Software (das BIOS und das BDOS) des CP/M-Plus, wodurch Ihnen die besten Vorteile Ihres augenblicklichen Systems erhalten bleiben.

Wir nennen Z3PLUS das Z-System für CP/M-Plus. Dieser Ausdruck beinhaltet sowohl das Z3PLUS Befehlsverarbeitungs-System, als auch seine vielen nützlichen Utility Programme.

### 1.1 Die Vorteile von Z3PLUS

Hier sind einige der allgemeinen Vorteile, die man durch Z3PLUS erhält.

- Das neue Befehlsverarbeitungs-System ist viel angenehmer, mächtiger und flexibler. Man kann es leicht an den eigenen Stil oder eigene Gewohnheiten anpassen.
- Hunderte exzellenter Programme, die für das Z-System geschrieben wurden, werden jetzt auf Ihrem Computer laufen.
- Sie können Public Domain und User Group Z-System Programme aus Z-System-Mailboxen (sogenannten Z-Nodes) downloaden und verwenden.
- Neue, hochqualitative User-Group- und Kommerzielle Software wird ständig für Z-System-Computer geschrieben, Programme, die endlich auch bei Ihnen laufen.
- Z3PLUS bewahrt die Vorteile von CP/M Plus: schnelle Diskettenzugriffe; bequeme Umleitung von Bildschirm, Tastatur und Drucker und die automatische Ausführung von SUBMIT Dateien.
- Das Z3PLUS-System ist gebrauchsfertig ("ready-to-run"). Assemblierung oder technische Installation ist nicht notwendig.
- Z3PLUS kann zu jeder Zeit aus dem System entfernt und später wieder gestartet werden.

## 1.2 Die Eigenschaften des neuen Kommando- Prozessors

Die zentrale Funktion des Z3PLUS-Systems ist die Kontrolle und die Abarbeitung von Befehlen, die das CP/M-System steuern. Es sorgt für

- bequemes Editieren falsch eingegebener Befehlszeilen
- Rückholmöglichkeit vorher eingegebener Kommandos zur Wiederverwendung, Korrektur, oder Modifikation
- konditionelle Ausführung einer Befehls-Sequenz, wobei die Ausführung abhängen kann von:
  - Erfolg/Mißerfolg des vorangegangenen Programms
  - Vorhandensein und Charakteristika notwendiger Dateien

- Benutzer-Privilegien und anderen System-Charakteristika
- Benutzer-Eingaben zur Laufzeit
- benannte Directories, um Dateien bequem zu trennen und optional Directories mit Paßwort zu schützen
- Kommando-Aliase, einzelne Befehle, die eine ganze Macro-Script-Datei auf einmal zur Ausführung bringen können

## 1.3 Andere Haupt-Eigenschaften

Zusätzlich zur Befehlsbearbeitung sorgt Z3PLUS für

- eine standardisierte Methode, die Full-Screen Fähigkeiten eines Terminals zu nutzen, verfügbar für alle Anwendungsprogramme, die davon Gebrauch machen möchten
- eine Programmier- und Benutzer- Umgebung, die Innovation und Kooperation begünstigt
- Portierbarkeit der Programme zwischen einem weiten Feld von Z80-kompatiblen Computern

Alles zusammen, bedeuten diese Eigenschaften, daß Sie von von einer reichhaltigen Sammlung von Programmen profitieren können, ohne daß diese speziell für Ihren Computer geschrieben wurden.

## 1.4 Zum Gebrauch des Handbuchs

In diesem Manual ist eine Menge Material und Sie werden gerne hören, daß Sie nicht alles lesen und verdauen müssen, bevor Sie das Z3PLUS anwenden können!

Kapitel 2 erklärt die Funktion der Installation des Systems und wie man es zum Laufen bringt.

Kapitel 3 zeigt die Grundlagen, was man mit dem System anfangen kann. Es demonstriert den Gebrauch einer Reihe von neuen Befehlen und schließt Beispiele mit ein. Diese zwei Kapitel sind grundlegend wichtig zu Lesen.

Kapitel 4 enthält viel mehr Information, was das Z-System ist und über einige seiner mächtigsten Eigenschaften. Um wirklich die Vorteile des Z-Systems zu nutzen, werden Sie über kurz oder lang auch dieses Kapitel lesen wollen.

Kapitel 5 beschreibt einige der fortgeschrittenen Möglichkeiten des Z3PLUS, besonders, wie man das System auf den eigenen, persönlichen Anspruch und Geschmack "zurechtschneidert". Bevor Sie nicht etwas Erfahrung mit dem Z-System haben, werden Sie wahrscheinlich keinen effektiven Nutzen aus dem Material dieses Kapitels ziehen können. Wir empfehlen trotzdem, dieses Kapitel "zum Gucken" zu überfliegen, damit Sie eine leichte Vorahnung der verfügbaren Eigenschaften erhalten.

Kapitel 6 ist in erster Linie ein technisches Nachschlagwerk. Es faßt zusammen und sammelt das Material, das an anderen Stellen des Handbuchs auftaucht, wie zum Beispiel die Befehls-Syntax und die Namen und Funktionen der mit Z3PLUS gelieferten Dateien. Es ist das Kapitel mit den meisten technischen Details. Es gibt jedoch einige Abschnitte, wie der, der die Unterschiede zwischen einem Z3PLUS System und einem CP/M-Plus System behandelt, die sich jeder Anwender irgendwann einmal ansehen sollte.

Zum Schluß sind im Kapitel 7, der Bibliographie, Referenzen auf andere Quellen für nützliche und hilfreiche Information über Z3PLUS das Z-System enthalten. Jeder Benutzer sollte dieses Kapitel lesen.

Während Sie sich weiter voranwagen mit Z3PLUS, wünschen wir Ihnen aufregende und vergnügliche Erlebnisse. Wir hoffen, daß dieses Handbuch und Ihr Ausprobieren des Z3PLUS Ihnen helfen kann, besser zu verstehen, wie Computer tatsächlich arbeiten und Ihren Rechner einfacher, produktiver und mit mehr Spaß anzuwenden!

## Kapitel 2

# Erstes Arbeiten mit Z3PLUS

### 2.1 Einleitung

Eine Schlüsseleigenschaft des Z-Systems ist seine Flexibilität. Innerhalb einer großen Bandbreite, kann man es auf seine eigenen Bedürfnisse zurechtschneiden. Natürlich bedeutet diese Flexibilität, daß es nicht eine einzige "Standard"-Einstellung gibt. Der Anfang ist sehr leicht, aber um das System voll auszuschöpfen braucht es schon etwas Zeit, um etwas über die Optionen zu lernen und eine Auswahl zu treffen.

Das Z-System ist in der Tat so flexibel, daß alte Hasen regelmäßig neue Eigenschaften entdecken, die im ursprünglichen Design nicht vorgesehen waren. Diese dynamische, innovative Eigenschaft ist ungewöhnlich für ein Microcomputer-Betriebssystem und zieht sowohl Anwender und Entwickler an, um fortwährend Verbesserungen zu machen.

Für Ihren Einstieg haben wir eine grundlegende Start-Konfiguration eingestellt. Sie beinhaltet alle verfügbaren Z-System-Features und ermöglicht Ihnen, sich damit vertraut zu machen. Ein voll ausgebautes System wie dieses reduziert Ihre TPA (Speicher, der Anwenderprogrammen zur Verfügung steht) um 6.75K. Später (S. 49) werden Sie lernen, wie man Spezialkonfigurationen erstellt, so daß Sie Ihre eigenen Kompromisse wählen können zwischen verfügbaren Features und

TPA-Größe, oder wie Sie Z-Systeme erzeugen können, die gerade 1K TPA abknapsen.

Dieses Manual ist nur eine Einführung in das Z-System. Während Sie beginnen, von Ihrem neuen Z3PLUS-System Gebrauch zu machen, werden Sie sicher die früheren, ausführlicheren Manuals und anderes schriftliches Material interessieren, das das Z-System im Detail beschreibt. Einiges davon ist in der Bibliographie aufgeführt (Kap. 7).

## 2.2 Das Zusammenstellen der Dateien

### NOTIZ

Sowohl Z3PLUS.COM, Z3PLUS.LBR und die Z3PLUS Dokumentations-Dateien, als auch das Manual sind durch Copyright geschützt und auf einen einzigen Anwender lizenziert. Es ist ungesetzlich, diese zu Kopieren oder an irgendeine andere Person weiterzugeben. Viele der begleitenden Z-System-Utilities, die im Lieferumfang von Z3PLUS enthalten sind, dürfen für nichtkommerzielle Zwecke zum Gebrauch durch andere Z-System-Anwender kopiert werden. Näheres hierzu entnehmen Sie bitte auch dem Copyright-Vermerk am Anfang dieses Handbuchs.

**Zuerst, bevor Sie irgendetwas anderes machen, erstellen Sie einen Vollbackup und eine Arbeitsdiskette der Z3PLUS-Distributionsdiskette und verstauen Sie die Originaldiskette an einem sicheren Ort.** Dann legen Sie die Arbeitskopie in eins Ihrer Zusatzlaufwerke (nicht das A-Laufwerk) und loggen diese ein. Die Datei RELEASE.NOT, so vorhanden, enthält Zusatzinformation, die zum Druckzeitpunkt noch nicht vorlag. Sie sollten diese einer kurzen Prüfung unterziehen, bevor Sie weitermachen.

Als nächstes verwenden Sie TCSELECT um das Terminal-Capability File für Ihren speziellen Computer zu erstellen. Es gibt fast für alle gängigen Terminals eine fertige Anpassung. Geben Sie daher ein:

```
TCSELECT myterm<cr>
```

wobei für "myterm" ein für Ihr Terminal geeigneter Dateiname Ihrer



Wahl stehen kann.<sup>1</sup>

TCSELECT bringt Ihnen ein Auswahlmenü der verfügbaren Terminals. Suchen Sie Ihr Terminal oder Ihren Computer und treffen Sie diese Selektion. Wenn Sie einmal Ihre Wahl bestätigt haben, schreibt TCSELECT eine Datei namens "myterm.Z3T"<sup>2</sup>. Sie benötigen daher Platz auf der Diskette und sollten den Schreibschutz entfernen. Sollte aus irgendeinem Grund Ihr Terminal oder Computer nicht im Menü enthalten sein, sehen Sie bitte im Handbuch Ihres Gerätes nach, ob eines der aufgeführten Terminals emuliert wird. Wenn Sie hier immer noch keinen Erfolg haben, sollten Sie entweder den Händler kontaktieren, wo Sie das Z3PLUS gekauft haben, oder einen der Z-Helfer, die in der Datei ZHELPERS.LST erwähnt werden. Diese können Ihnen wahrscheinlich bei der Erstellung Ihres Z3T-Files behilflich sein. In der Zwischenzeit können Sie Z3PLUS ohne Terminal-Definition verwenden, aber Sie können keine solchen Programme verwenden, die bestimmte Terminaleigenschaften voraussetzen, dazu gehören zum Beispiel SALIAS, SHOW, und ZFILER.

Als nächstes kopieren sie mit PIP oder einem anderen Kopierprogramm die folgenden Dateien auf Laufwerk A User 0 (Normalbereich beim Booten):

Z3PLUS.COM	ARUNZ.COM	LSH.COM
Z3PLUS.LBR	ALIAS.CMD	ZFILER.CMD
LGET.COM	LX.COM	LPUT.COM
SALIAS.COM	IF.COM	myterm.Z3T
SDZ.COM	ZEX.COM	SHOW.COM
LDIR.COM	ZERRLSH.COM	

Zum Schluß sollten Sie eine der beiden ZFILER-Versionen kopieren. Dieses Programm verwendet Bildschirm-Hervorhebung. Weil sich Reverse-Video und Dim-Video so stark unterscheiden, stehen zwei verschiedene Versionen des ZFILER zur Verfügung: ZF-REV.COM für Reverse-Video und ZF-DIM.COM für Dim-Video. Sie sollten die für Ihr Terminal geeignete Version wählen. Für ein Dim-Video Terminal würde man zum Beispiel folgendes eingeben:

<sup>1</sup>Wenn Sie einen AMSTRAD CP/M PLUS Rechner besitzen, können Sie mit dem Heath-Zenith Terminal arbeiten, oder aber von Ihrem Händler, falls nicht vorhanden, ein vordefiniertes Terminal erhalten (AMSTERMX.LBR). Benennen Sie dieses in DEFAULT.Z3T um, und benutzen Sie für PCW-Rechner CPSET CON 90 32 30

<sup>2</sup>Dieser Dateiname ist nur symbolisch. Er könnte genausogut ein Phantasienamen sein. Wichtig ist nur die Endung .Z3T. Damit wird ein Terminalfile gekennzeichnet.

```
PIP A:ZFILER.COM=ZF-DIM.COM<cr>
```

## 2.3 Der Betrieb von Z3PLUS

Jetzt sind Sie startbereit für Z3PLUS. Einfach auf Laufwerk A: einloggen und eingeben:

```
Z3PLUS<cr>
```

Ihr neues Z3PLUS System wird generiert, geladen, und automatisch ausgeführt. Beim ersten Mal meldet Z3PLUS

```
STARTZ3P?
```

Diese Meldung können Sie für's erste ignorieren; sie bedeutet, daß Z3PLUS die Datei STARTZ3P.COM nicht finden kann (wir haben sie noch nicht erzeugt). Damit befassen wir uns auf Seite 11.

Das System ist komplett, außer daß es noch keine Ahnung von Ihrem Terminal hat. Wir stellen jetzt eine weitere Fähigkeit von Z3PLUS vor, indem wir es veranlassen, den Terminal-Deskriptor zu laden. Geben Sie ein

```
Z3PLUS myterm.Z3T<cr>
```

Das war alles, damit Z3PLUS läuft! Jetzt können Sie mit dem nächsten Kapitel fortfahren, um mehr über das System zu lernen.

## 2.4 Die Organisation Ihrer Dateien

Nachdem Sie Z3PLUS und das Z-System kennengelernt haben und bereit sind, diese auf einer normalen Basis anzuwenden, sollten Sie die Dateien auf Ihren Disketten organisieren.

Wenn das Laufwerk A: eine Hard-Disk oder ein Floppylaufwerk mit größerer Kapazität ist, können Sie fast alle Z3PLUS- und Z-System-Tools im Verzeichnis A0: halten und es zu Ihrem Root Directory machen (das letzte Verzeichnis im Suchpfad). Einige Z-System-Anwender ziehen A15: für diese Dateien vor, aber Z3PLUS, benutzt in der Lieferversion A0: als Root (Hauptverzeichnis) seines Suchpfades.

Wenn Sie ein kleineres Diskettenformat haben, belassen Sie nur die am häufigsten benötigten Dateien dort und halten Sie die anderen auf einer separaten Z-Tools-Diskette parat. Wenn Z3PLUS geladen ist, werden die Dateien Z3PLUS.COM und Z3PLUS.LBR erst wieder benötigt, wenn Sie die gegenwärtige System-Konfiguration ändern wollen, sie können daher auf eine separate "Boot Disk" verbannt werden. Sie können Z3PLUS so konfigurieren, daß Z3PLUS.LBR in einem anderen Verzeichnis gesucht wird (siehe S. 6.5), etwa auf B: oder M:. Sie könnten dann auch vom momentanen Laufwerk A: Z3PLUS "booten":

```
B:Z3PLUS<cr>
```



## Kapitel 3

# Z3PLUS kennenlernen und verstehen

Wir empfehlen Ihnen, die folgenden Schritte wirklich auszuführen, um mit ein paar der normalerweise benutzten Features des Z3PLUS Z-Systems vertraut zu werden.

### 3.1 Wie man einen Alias erzeugt

Ein Alias ist ein einzelnes Wort (ein Befehl), stellvertretend für einen längeren oder zusammengesetzten Befehl. Um einen Alias zu erzeugen, rufen Sie das Programm SALIAS (Screen ALIAS) auf. Wir werden jetzt einen Startup-Alias namens STARTZ3P erzeugen, der automatisch Ihre Terminal-Datei in das Z3PLUS System einbindet, wenn Sie das nächste Mal Z3PLUS eintippen. (Sie haben diesen Namen STARTZ3P schon einmal gelesen. Als Sie das erste Mal Z3PLUS starteten, war er auf dem Bildschirm zu sehen, gefolgt von einem Fragezeichen.)

Geben Sie ein

```
SALIAS STARTZ3P<cr>
```

um das SALIAS-Programm zu starten und den STARTZ3P-Alias vorzubereiten. Sie werden sofort bemerken, daß dieses Utility, wie viele Z-System-Tools, sehr effektiven Gebrauch macht von den Darstel-

lungsmöglichkeiten Ihres Terminals (und deshalb war es wichtig, am Anfang eine Terminal-Capabilities-Datei auszuwählen).

Das SALIAS-Programm ist ein Alias-Editor, eine Art Textverarbeitung für Alias-Scripts. Man gibt den Text des Alias-Scripts ein und benutzt Funktionstasten, um Ihre Eingaben zu editieren. Die meisten Tasten, einschließlich der für die Cursorbewegung, entsprechen denen von WordStar. Gibt man Control-J ein, erhält man eine Liste der zur Verfügung stehenden Befehle.

Für diesen Alias ist die einzige einzugebende Befehlszeile

```
Z3PLUS myterm.Z3T /Q<cr>
```

Wir haben hier die “/Q” Quiet Option benutzt, um die extensive Information zu unterdrücken, die Z3PLUS ausgeben kann, wenn es etwas lädt. Jetzt tippen Sie Control-K und danach Control-X zum Verlassen und Speichern der neuen Datei.

Versuchen Sie jetzt, diesen Alias auszuführen durch Eingabe von

```
STARTZ3P<cr>
```

Auf dem Bildschirm sieht man, daß er tatsächlich das Z3PLUS Programm ausführt und (wiederum) die Z3T-Datei lädt. Bis hierher erspart uns das lediglich etwas Tipparbeit. Wir haben jedoch genau diesen Namen gewählt für den Alias, weil Z3PLUS diesen Alias automatisch ausführt, wenn es das nächste Mal gestartet wird. Probieren wir es soweit aus. Geben Sie ein

```
Z3PLUS OFF<cr>
```

Jawohl, das Z3PLUS-System ist weg und der Standard-CP/M PLUS Kommando-Prozessor ist wieder aktiv. Geben Sie jetzt ein

```
Z3PLUS<cr>
```

um das Z3PLUS-System zu starten. Dieses Mal wird es automatisch Ihre Z3T-Terminal-Datei laden, indem es den neuen STARTZ3P-Alias verwendet.

Der Startup-Alias, den Sie eben erzeugt haben, ist ganz simpel, aber nichtsdestoweniger nützlich. Ein Alias kann jedoch viel mehr. Er kann

eine ganze Sequenz von Befehlen sein, mit symbolischen Parametern für Dateinamen, Directories, und Befehlszeilen-Parameter. Eines der Dinge, die der Startup-Alias beispielsweise tun kann, ist, sofort einen Befehls-Suchpfad einzurichten, der sich vom Standard-Suchpfad unterscheidet, für den Z3PLUS.COM konfiguriert ist. Auf Seite 18 finden Sie ein etwas ausgearbeiteteres Beispiel. Für mehr Information schlagen Sie in den Referenz-Werken der Bibliographie nach. Und vergessen Sie nicht SALIAS. Es wird Ihnen viele Aufgaben Ihrer Computerarbeit abnehmen können.

## 3.2 Welche Befehle gibt es?

Geben Sie jetzt ein

```
H<cr>
```

Das wird Ihnen alle eingebauten Befehle des aktiven Z-Systems anzeigen. Der Bildschirm zeigt eine Ausgabe ähnlich der in Tafel 3.1.

FCP				
IF	AND	OR	ELSE	FI
IFQ	XIF	ZIF		
CPR				
GET	GO	JUMP		
RCP				
CLS	ECHO	ERA	H	NOTE
P	POKE	PORT	R	REG
SP	TYPE	WHL	WHLQ	

Table 3.1: Residente Befehle, die durch den H-Befehl angezeigt werden.

Die angezeigten Befehle sind die, die zu dem Zeitpunkt im Speicher waren, als der H-Befehl gegeben wurde. Sie werden variieren, je nachdem welcher FCP (Flow Command Package), RCP (Resident Command Package), und CPR (Command Processor) geladen sind. Wie man sieht, ist der H-Befehl (Abkürzung für "HELP") Bestandteil des RCP Pakets.

Später werden Sie sehen, daß Sie Versionen von Z3PLUS erzeugen können, die weniger Speicher brauchen, indem Sie das RCP weglassen. Auch gibt es einige RCP-Pakete, die den H-Befehl nicht unterstützen. In diesen Fällen kann SHOW, beschrieben auf Seite 22, benutzt werden, um diese Information (und noch viel mehr) auszugeben..

### 3.2.1 RCP Kommandos

Sehen wir uns zuerst einmal die Befehle des RCP (Resident Command Package) an. Eben haben wir schon den H-Befehl benutzt. Versuchen Sie einen anderen, indem Sie eingeben

```
ECHO Das ist ein Test.<cr>
```

Bevor wir weitermachen, sollten wir speziell darauf hinweisen, daß ein Strichpunkt (“;”) zum Trennen der Befehle benutzt wird, um mehrere Befehle in einer Zeile zu trennen. Versuchen Sie

```
ECHO Test 1;ECHO Test 2;ECHO Test 3<cr>
```

ECHO ist ein einfacher, aber wichtiger Befehl, der oft verwendet wird, um dem Anwender mitzuteilen, was gerade passiert, wenn eine Sequenz von einem Script ausgeführt wird. Wir werden davon ein Beispiel sehen auf Seite 18. Weil Befehls-Zeilen immer in Großbuchstaben umgewandelt werden, ist Ihre Nachricht so erschienen. Der ECHO Befehl wird die Buchstabenfolge “%>” als “in Kleinbuchstaben umwandeln” interpretieren und die Sequenz “%<” als “in Großbuchstaben wandeln.”

Versuchen Sie jetzt das SP Kommando, das Ihnen Auskunft über den freien Speicherplatz auf einer Diskette erteilt:

```
SP<cr>
SP B:<cr>
```

Der P-Befehl “peekt” in den Speicher, der Inhalt wird gelesen. Beispiel

```
P 100<cr>
```

Wenn Sie von Maschinensprache keine Ahnung haben, ist das nicht sehr aufschlußreich, aber wenn es der Fall sein sollte, ist das ein nützlicher Befehl. Versuchen Sie jetzt den POKE-Befehl:



POKE 100 "Das ist ein Poke-Test<cr>

Vergewissern Sie sich, daß Sie die Anführungszeichen am Anfang eingegeben haben; damit weiß POKE, daß Sie Text eingeben und nicht etwa Hex-Code. Sehen wir uns das Ergebnis an:

P 100<cr>

Es ist klar, daß man POKES für spezielle Zwecke reserviert, wenn man weiß, daß man nichts anderes zerstört (siehe S. 20). Peeken kann man jedoch nach Belieben. ERA sollte Ihnen vertraut sein; damit löscht man Dateien. TYPE zeigt den Inhalt einer Text-Datei auf dem Bildschirm an. CLS löscht den Bildschirm (praktisch in Scripts). NOTE, ein Tue-nichts-Befehl, wird in Scripts für Kommentare verwendet. PORT ist eine Kombination von Peek/Poke für die Input/Output Ports eines Computers; es liest die ankommenden Datenbytes oder gibt Sie an eine bezeichnete I/O-Schnittstelle aus. Wie beim Poke Befehl für den Speicher, sollten Sie auch hier größte Vorsicht walten lassen.

Der R-Befehl führt einen Reset des Disk-Systems durch; er wirkt wie Control-C, kann aber in eine Befehlssequenz eingebettet werden. REG zeigt oder modifiziert Software-Register des Z-Systems, die untereinander Werte austauschen. Der WHL Befehl findet beim Sicherheits-Byte des Z-Systems, dem sogenannten "Wheel Byte" Verwendung. Sie werden feststellen, daß viele Befehle, besonders riskante, wie zum Beispiel ERA, nicht funktionieren, wenn das Wheel Byte "off" ist. WHLQ (WHeel Query) (nicht mehr in Verwendung, siehe RELEASE.NOT). WHL, wenn danach das korrekte Paßwort angegeben wird, schaltet das Wheel Byte ein; andernfalls wird es abgeschaltet. Das vorgegebene (Default-) Paßwort ist SYSTEM.<sup>1</sup>

### 3.2.2 FCP Kommandos

Das Z-System kann durch den Gebrauch der Befehle im Flow Command Package (FCP) ganze Sets von Kommandos auf intelligente, automatisierte Weise ausführen. Flow Control benötigt etwas Zeit zum Verständnis, ist aber so mächtig und nützlich, daß wir glauben, daß es den Aufwand zu Lernen wert ist. In einer Nutshell ist es der Zweck der

<sup>1</sup>Wenn Sie es ändern wollen, kann die Datei DEFRCP.ZRL in Z3PLUS.LBR gepatcht werden. Suchen Sie mit Ihrem Lieblings-Patchutility (zum Beispiel ZPATCH) nach dem String "SYSTEM" und ändern Sie ihn nach Belieben. Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren Händler oder einen Z-Helfer.

Fluß Steuerung, Ihren Kommandos verschiedene Ebenen der IF-ELSE-ENDIF-Abprüfung (wie in Hochsprachen) zu ermöglichen, um zu kontrollieren, welcher von einer Reihe von alternativen Befehlen ausgeführt wird.

Das Hauptkonzept, das Sie begreifen müssen, ist das von “Flow States”. Wenn das Z-System erstmals läuft, ist kein Ablauf-Status aktiv (wir nennen ihn den “Null” Status). Unter der Kontrolle der Befehle im FCP-Modul können bis zu 8 Ebenen von Flußzuständen aktiv werden. Jede Ebene kann entweder den Wert TRUE oder FALSE annehmen (der Null-Status wirkt wie ein “TRUE” Status). Ist der gegenwärtige Ablaufstatus TRUE, laufen die Befehle normal ab; ist er FALSE, ignoriert der Kommando-Prozessor alle Befehle, außer denen im FCP. Wie das funktioniert, werden Sie gleich anhand von praktischen Beispielen sehen.

Der IF-Befehl wird mit einer Art Prüfausdruck verwendet, der entweder “TRUE” oder “FALSE” bewertet wird. Für jeden Wert wird eine neue Ebene im Ablauf aktiviert. War der vorangegangene Status FALSE, wird der neue Status immer FALSE, unbeschadet des Ergebnisses der durchlaufenen Bedingung. Auf der anderen Seite, wenn der vorangegangene Status TRUE war, nimmt der neue Status den Wert des letzten Ergebnisses an.

Hier sind einige Beispiele von Bedingungs-Prüfungen, die vom IF-Befehl ausgeführt werden können:

IF ERROR	prüft, ob ein vorangegangenes Programm das Programm-Errorflag gesetzt hat
IF REG 3 > 1	prüft den Wert von Anwender-Registern
IF INPUT RUN WS?	gibt den Prompt aus “RUN WS?” und wartet auf die Reaktion des Anwenders
IF ~EXIST DIR:FN.FT	prüft auf (Nicht-)Existenz einer Datei

Geben Sie die Befehlszeile

```
IF //<cr>
```

ein, um eine geraffte Übersicht der benutzbaren Prüfbedingungen zu erhalten. Wie Sie sehen können, gibt es ganz schön viele davon. Später können Sie noch damit experimentieren. Nur die ersten zwei Buchstaben werden wirklich benutzt, um die Prüfbedingung zu untersuchen (daher ist EX dasselbe wie EXIST), und eine vorangestellte Tilde (“~”)

kehrt die Wirkung um. Die meisten Prüfungen werden durch das Programm IF.COM vorgeneommen, das vom FCP automatisch geladen und ausgeführt wird. Der IFQ (IF Query)-Befehl ist im FCP enthalten, um Ihnen den Ablauf zu verdeutlichen und Ihnen beim Erlernen des Gebrauchs der Ablaufsteuerung zu helfen. Wir werden es jetzt bei einigen Beispielen anwenden.

Probieren Sie erst den IFQ-Befehl aus. Vorausgesetzt, Sie haben nicht schon auf eigene Faust experimentiert (in diesem Fall geben Sie bitte ein: "ZIF<cr>", um zum Null-Status zurückzukehren), sollten Sie die Meldung "IF None" sehen, die Ihnen mitteilt, daß kein aktiver Ablaufstatus vorliegt (d.h., "Null"-Status). Geben Sie jetzt ein :

```
IF T<cr>
```

Diese Prüfung ist, wie der IFQ-Befehl, hauptsächlich als Lernhilfe gedacht. Wie Sie vermutlich erraten haben, gibt das einen Wert TRUE zurück. Rufen Sie jetzt noch einmal IFQ auf. Die Anzeige lautet "IF T", damit wird angezeigt, daß ein Ablauf-Status aktiv ist und den Wert TRUE besitzt. Versuchen Sie einen Befehl wie "ECHO TESTING<cr>" und sehen Sie, daß er normal abläuft.

Geben Sie jetzt die Befehlszeile ein:

```
IF F;IFQ<cr>
```

Die Anzeige lautet jetzt "IF FT". Das zeigt uns, daß zwei Ebenen von Ablauf-Zuständen aktiv sind, wobei der gegenwärtige FALSE und der darunterliegende TRUE. Versuchen Sie jetzt den Befehl "ECHO TESTING<cr>" . Er wird ignoriert.

Der FI-Befehl, der ein rückwärts gelesenes IF darstellt, beendet den augenblicklichen Ablauf-Status und versetzt das System in den früheren Zustand (oder in den Null-Status). Geben Sie die Befehlszeile

```
FI;IFQ<cr>
```

mehrmals hintereinander ein und achten Sie darauf, wie sich die Meldungen ändern. Zurück im Null-Status, geben Sie den Befehl ein

```
IF T;IF F;IFQ<cr>
```

um uns auf den alten Stand zu bringen (mit dem gegenwärtigen Status FALSE und einem darunterliegenden Ablaufstatus, der TRUE ist).

Jetzt werden wir etwas über den ELSE Befehl erfahren, der den gegenwärtigen Status umkehrt, vorausgesetzt, der darunterliegende Status ist TRUE. Geben Sie ein

```
ELSE;IFQ<cr>
```

und beachten Sie, daß die Anzeige sich von "IF FT" auf "IF TT" ändert. Jetzt funktionieren die Befehle wieder. Geben Sie es ein paar Mal mehr ein, und sehen Sie, wie es den Ablauf-Status "toggelt", d.h. umschaltet.

Ab hier wissen Sie genug über Flußkontrolle (Flow Control) und die FCP Befehle, um selbst zu experimentieren. Wenn Sie fertig sind, geben Sie "ZIF<cr>" ein, um den Ablauf-Status zurückzusetzen.

Bevor wir das Thema verlassen, möchten wir Ihnen ein letztes Beispiel vorstellen, das Ihnen eine Idee geben soll, wie die Ablaufkontrolle in Verbindung mit Aliases sehr mächtige Befehle ergeben kann.

Verwenden Sie SALIAS, um einen Alias namens TEST zu erzeugen, indem Sie den Befehl eingeben

```
SALIAS TEST<cr>
```

Wenn der Eingabe-Bildschirm verschwindet, geben Sie die folgende Befehls-Sequenz ein:

```
IF EX $1
ECHO FILE $1 EXISTS
IF EQ $1 *.COM
ECHO COM %>FILE: %<YES
ELSE
ECHO COM %>FILE: %<NO
FI
ELSE
ECHO FILE $1 NOT FOUND
FI
```

Beachten Sie, daß wir die spezielle Buchstabenreihenfolge benutzen, die ECHO als Signale zum Umschalten zwischen Groß - und Kleinschreibung erkannt werden. (siehe S. 14) Wenn Sie fertig sind, drücken Sie Control-K Control-I. Beachten Sie, wie SALIAS komfortabel die Befehle einrückt, um die Verschachtelung der Ablaufsteuerung zu zeigen. Sichern Sie jetzt den Alias durch Eingabe von Control-K Control-X.

Der Ausdruck “\$1” im Alias steht für den ersten Parameter, der dem Befehl TEST folgt, wenn der Alias angewendet wird. Versuchen Sie die folgenden Befehle und beachten Sie, was in jedem Fall passiert:

```
TEST Z3PLUS.COM<cr>
TEST Z3PLUS.LBR<cr>
TEST NOFILE.JNK<cr>
```

### 3.2.3 CPR Commands

Die Standard-Version des Kommando-Prozessors hat nur drei Befehle. Der Großteil des Codes im CPR wird eher benötigt, um raffinierte Operationen mit *Ihren* Befehlen auszuführen, als um eigene Befehle auszuführen. Die meisten der (im Speicher) residenten Befehle werden vom RCP zur Verfügung gestellt, der Ihnen den Vorteil bietet, daß Sie den Befehlsvorrat zu ändern, indem Sie unterschiedliche RCP-Module laden, abhängig von Ihren augenblicklichen Bedürfnissen.

Die Befehle im CPR machen Gebrauch von Code, der schon für andere Zwecke existiert. GET und GO/JUMP sind die zwei grundlegenden Hälften des normalen Laden und Ausführen Prozesses, der immer dann abläuft, wenn Sie einen Befehl auswählen, der sich auf eine COM-Datei bezieht. Alle drei Befehle können gefährlich sein; bei inkorrektter Anwendung kann ein System-Absturz erfolgen, der Neubooten erfordert. Bei richtiger Verwendung können sie sehr mächtig und nützlich sein.

Der GET-Befehl führt nur den Ladevorgang aus. Er bietet mehr Flexibilität als der normale Befehlsloader. Erstens geben *Sie* die Speicher-Adresse an, an die die Datei geladen werden soll, zweitens kann jede Art von Datei (nicht nur COM-Dateien) geladen werden. Nichts kann verhindern, daß Sie ein Programm an eine falsche Adresse laden (sogar an eine für das Betriebssystem reservierte Adresse!) oder daß Sie eine Datei laden, die gar kein Programm ist.

Versuchen Sie die Befehlszeile

```
GET 200 Z3PLUS.LBR;P 200<cr>
```

Die Library wird in den Speicher geladen an die ADRESSE 200H, und der PEEK Befehl wird Ihnen den Anfang der Datei zeigen. Wenn Sie sich je gefragt haben sollten, wie eine Library-Datei in Wirklichkeit aussieht, können Sie das jetzt sehen. GET und P können als Partner ein nützliches Lernwerkzeug sein.

So, wie GET nur das Laden einer Datei übernimmt, führen GO und JUMP lediglich aus, was schon im Speicher ist. GO führt automatisch aus, was immer sich an der Adresse 100H befindet. JUMP ist ähnlich, aber läßt es zu, daß man die Ausführungsadresse angibt. (GO entspricht JUMP 100). Für normale COM-Programme, erfolgt durch die Kombination von GET und GO eine normale Ausführung. Daher bewirkt die Befehls-Sequenz

```
GET 100 SDZ.COM;GO *.LBR<cr>
```

das Gleiche <sup>2</sup> wie

```
SDZ *.LBR<cr>
```

Warum also, würde dann jemand diese separaten Befehle benutzen? Wir wollen dazu zwei Beispiele betrachten. Zuerst, kann der GO-Befehl benutzt werden, um ein Program wiederholt auszuführen, ohne Zeit mit dem Laden der Datei von Diskette verschwenden zu müssen. Versuchen Sie

```
SDZ *.COM<cr>
```

und dann

```
GO *.LBR<cr>
```

Vorsicht: einige Programme sind nicht zur Wieder-Ausführung gedacht und könnten Ärger machen bei dieser Anwendungsart. Die meisten Z-System-Tools können wiederholt ausgeführt werden, und falls nicht, brechen diese of wenigstens gnädig ab. Andere Programme könnten weniger nachsichtig sein, daher sollten Sie sehr vorsichtig experimentieren.

Ein zweites Beispiel des Gebrauchs von GET und GO ist die Technik, die man "GET, POKE, und GO" nennt, oder einfach POKE&GO. Der Grundgedanke hierbei ist, die Datei zu laden, indem man den GET-Befehl verwendet, sie durch den POKE-Befehl zu modifizieren und dann mit GO (oder JUMP) aufzurufen. Zum Beispiel, wenn Ihre Textverarbeitung (nennen wir sie mal WP.COM) die Werte für den rechten Rand an der Adresse 83BH hat und normalerweise den Wert 76 benutzt, könnten Sie einen Alias-Befehl namens WP60 erzeugen, wobei der folgende Alias-Script einen Wert von 60 einstellt:

---

<sup>2</sup>Naja, fast das Gleiche

```
GET 100 WP.COM;POKE 83B 3C;GO $* <cr>
```

Beachten Sie, daß der gepokte Wert Wert ist in hexadezimaler Form (3CH=60) angegeben wird und daß der Ausdruck "\$\*" in einem Alias-Script durch die Benutzereingabe (Parameter) in der Befehlszeile hinter dem Namen des Befehls ersetzt wird. Dadurch wird der Befehl

```
WP60 NEWFILE.DOC <cr>
```

WP mit der Datei NEWFILE.DOC ausführen, aber mit einem rechten Rand von 60. Wenn man das außerhalb des Z-Systems machen möchte, bräuchten Sie dafür eine Zweitversion des ganzen Textverarbeitungs-Programms. Ein SALIAS-Alias ist nur 1K lang; ein ARUNZ-Alias (S. 34) für diese Funktion nimmt nur etwa 40 Bytes des Diskettenplatzes ein.

### 3.2.4 Transiente Kommandos

“Transientes Kommando” ist der Name, den Digital Research (Herausgeber des CP/M), solchen Befehlen gegeben hat, die nicht im Speicher verbleiben, sondern von Diskette geladen werden. Unter CP/M Plus gehören dazu Dateien des Typs COM wie unter CP/M 2.2 und des Typs SUB oder PRL, (je nach Ausführung des CP/M Plus). SUB und PRL Dateien laufen unter Z3PLUS so ziemlich wie unter CP/M Plus ab.

Zusätzlich zu den normalen COM-Dateien, die CP/M unterstützt, gibt es einige spezielle Typen von COM-Dateien, die das Z-System unterstützt. Diese Dateien sind als type-1, type-3<sup>3</sup>, und type-4 angelegt.

Type-1-Dateien ähneln den Standard CP/M COM-Dateien, außer, daß diese einen speziellen Header an ihrem Anfang besitzen, in den der Kommando-Prozessor die Adresse dessen einträgt, das man Z-System Environment Descriptor nennt. Das ist ein Speicherbereich, das eine vollständige Beschreibung der Z-System-Konfiguration enthält. Mit dieser Information kann ein Programm von den speziellen Z-System-Gegebenheiten Gebrauch machen.

Type-3 Programme sind ähnlich wie type-1-Programme, außer, daß sie, im Gegensatz zu Standard CP/M-Programmen, nicht zwingend an die Standard-Adresse 100H geladen werden. Ihre Ausführungsadresse

---

<sup>3</sup>Ja, es gibt auch type-2-Dateien, aber die sind heutzutage selten in Gebrauch.

ist in dem speziellen Header enthalten, und der Kommando-Prozessor lädt sie automatisch an die dort angegebene Adresse. Zum Beispiel das Programm ARUNZ.COM, das im Z3PLUS Paket enthalten ist, ist ein type-3-Programm, das bei 8000H abläuft..

Type-4-Programme sind spezielle Z-System-Programme, die von PRL-(sogenannten Page ReLocatable)-Dateien. Sie enthalten Information die es ihnen gestatten, an jeder Adresse zu laufen. Der Prozessor berechnet automatisch die höchste Adresse im Speicher, an der ein solches Programm mit der bestehenden Konfiguration laufen kann, lädt es und ruft es an dieser Adresse auf.

Die type-3- und type-4-Programme wurden erfunden für transiente Programme, die als residente Teile des Systems fungieren sollen (z.B. als Erweiterter Kommando-Prozessor, als Shell, oder Fehlerbehandlung (Error Handler)), oder als transiente Versionen gemeinsamer residenter Befehle (z.B. ERA.COM oder REN.COM). Da das Z-System selbst sehr oft automatisch diese Befehle aufruft, können Sie nicht immer voraussehen, wann einer davon aktiv sein wird und es ist nützlich, wenn Sie nicht die unteren Speicherbereiche stören, wo die Programme laufen.

Spezielle Z-System-Programme laufen gewöhnlich nicht korrekt unter CP/M. Da sie entstanden sind, um auf Computern mit *dauerhaft* installiertem Z-System zu laufen, enthalten die meisten keinen Code, um den Versuch zu vereiteln, sie unter gewöhnlichem CP/M zu starten. Die meisten mit Z3PLUS gelieferten, werden jedoch bei einem solchen Versuch gnädig aussteigen. Einige werden eine Meldung ausgeben, andere werden nur einen Warmstart erzwingen.

### 3.3 Mehr über die System-Konfiguration

Wir erwähnten oben, daß Sie unter bestimmten Umständen den H Befehl benutzen, um zu sehen, welche residenten Befehle zur Verfügung stehen. Das Z-System-Utility SHOW kann immer benutzt werden, um dies anzuzeigen und noch reichlich mehr Information über Ihr System. Die beste Art und Weise, etwas über die Konfiguration Ihres Z3PLUS-Systems zu erfahren, ist, SHOW aufzurufen und mit seinem Menü zu experimentieren. Einige der Dinge, die man mit SHOW tun kann sind:

- Die Speicherbelegung als Memory-Map ausgeben



- anzeigen der Befehle, die von CPR, FCP, und RCP erkannt werden, mit der Anzeige, welche Befehle den Wheel-Byte Status “on” erfordern
- anzeigen der besonderen Eigenschaften, die im CPR gegeben sind
- anzeigen des Befehls-Suchpfads und der Verzeichnis-Namen
- anzeigen von Information im Shell-Stack und dem Message-Puffer

### 3.4 Inkompatible Programme

Die meisten Programme, die für das Z-System geschrieben wurden, werden unter Z3PLUS perfekt laufen. Es gibt jedoch ein paar, die inkompatibel zu Z3PLUS sind und nicht aufgerufen werden sollten. Die bekanntesten sind:

CLEANDIR	Directory Saubermann Utility
DU3	ZCPR Disk Utility
MOVE	Datei anderem Userbereich zuordnen
MAKE (MAKEUSER)	Datei anderem Userbereich zuordnen
ZEX vor V 5.0	Residenter Submit Prozessor ältere Versionen laufen nicht

Jedes dieser Programme, außer ZEX, greifen direkt auf CP/M-2.2 BIOS Routinen zu, wenn es um Datei- oder Disketten-Operationen geht, etwas, das unter CP/M-Plus abweichende Routinen verlangt. Solche Programme werden ziemlich wahrscheinlich einen “Absturz” herbeiführen und ein Neubooten erfordern. Daher sollten Sie diese vermeiden. ZEX versucht in den älteren Versionen vor Version 5.0 I/O Umleitungen auf eine Art zu machen, die nicht zu CP/M-Plus kompatibel war.

Gelegentlich treffen Sie auf ein Programm, das bizarre Disk-Kapazitäten meldet. Das ist wenig informativ, aber harmlos und wird durch einen weiteren Unterschied zwischen CP/M-2.2 und CP/M-Plus bedingt. Nach und nach sollten die meisten Z-System Tools verbessert werden, so daß der freie Speicherplatz unter beiden Betriebssystemen korrekt angezeigt wird.



# Kapitel 4

## Mehr vom Z-System

Das Z-System unterteilt sich in

- (Speicher-) Residente Komponenten
- Automatische Tools und Shells — die Spezial-Programme, die der Kommando- Prozessor für Sie ausführt
- andere Tools — Programme, die von residenten Komponenten kontrolliert werden oder die solche verwenden

Die Komponenten, die im Z3PLUS-Paket enthalten und unten beschrieben sind, stellen eine relative Grund-Ausrüstung dar, und doch, sind sie schon reichhaltig an Features. Die folgenden Beschreibungen sind hier für Sie zum Nachschlagen, aber Sie müssen sicher nicht alles beim ersten Lesen absorbieren.

### 4.1 Residente Z3PLUS Komponenten

- Der Z3PLUS Kommando-Prozessor: Er interpretiert Kommandos und lädt Programme.
- Ein RCP (Residentes Kommando-Paket): Es beinhaltet einen erweiterten Befehls-Vorrat, der schnell gewechselt werden kann (Siehe S. 14).

- Ein FCP (Fluß -Kommando-Paket): Es prüft logische Bedingungen und managt den “Flow state”, den Flußstatus (IF-Test) des Systems, so daß Kommandos konditional ausgeführt werden können (siehe S. 15).
- Ein NDR (Named Directory Register, Register für Benannte Verzeichnisse): Es enthält eine Tabelle, die Directories (Laufwerk/User-Kombinationen-“drive/user=DU”) Namen und optional Paßwörter zuordnet.
- Einen Pfad PATH: Er zeigt die Sequenz von Directories, die nach Programmen durchsucht werden.
- Z3T Terminal Deskriptor: Es enthält die Kontroll-Sequenzen, die von Ihrem Terminal benötigt werden, um den Cursor zu positionieren, den Bildschirm zu löschen, etc.
- Andere Komponenten des Z-System-Environments: Diese beinhalten den Kommandozeilen-Puffer, einen “Message Puffer”, einen Shell Stack, das Wheel byte, den externen File Control Block, und einen Kommando Prozessor Stack.

## 4.2 Automatische Kommandos

Einige Z-System-Tools werden unter bestimmten Bedingungen automatisch ausgeführt.

### 4.2.1 Erweiterter Kommando-Prozessor

Wenn der Kommando-Prozessor einen Befehl weder als speicherresidentes Kommando, noch als Disk-Datei finden kann, dann wird dieser an ein Programm übergeben, das als Erweiterter Kommando-Prozessor (ECP) bezeichnet wird. Z3PLUS versucht stets, als ECP ein Programm mit dem Namen CMDRUN.COM im ROOT-Directory zu laden (das ist das letzte Directory in Ihrem Suchpfad). Das im Vertrieb befindliche Z3PLUS-System beinhaltet zwei Programme, die oft als ECPs eingesetzt werden:<sup>1</sup>

---

<sup>1</sup>Es gibt sogar einen Weg, beide als ECP zusammenwirken zu lassen.

- ARUNZ (Alias-RUN-für-das-Z-system), dessen Funktion sehr ähnlich ist zu den Aliases, die mit SALIAS erstellt wurden, außer, daß
  - eine große Zahl von Aliases in einer einzigen Textdatei (ALIAS.COM) angelegt werden können und
  - eine viel ausführlichere Parameter-Bearbeitung möglich ist.

ARUNZ kann als sehr leistungsfähiger Kommandogenerator und Übersetzer verwendet werden.

- LX (Library eXecutive), das Kommandos aus einer Library namens Command.LBR <sup>2</sup> heraus aufruft. COM-Dateien in Command.LBR abzulegen spart sowohl Directory-Platz, als auch Diskettenkapazität. LX ist besonders nützlich auf Computern mit einem Diskettenformat von geringer Kapazität, oder wenn zu wenige Verzeichnis-Einträge möglich sind, wie etwa auch auf Festplatten machmal der Fall ist.

Um eines dieser Programme als Ihren Erweiterten Kommando-Prozessor einsetzen zu können, kopiert man es unter dem Namen CMDRUN.COM.<sup>3</sup> Um zum Beispiel ARUNZ als ECP zu verwenden, gibt man ein:

```
PIP CMDRUN.COM=ARUNZ.COM<cr>
```

Sie sollten das jetzt tun, da wir in Kürze (S. 34) ARUNZ für einige Beispiele verwenden wollen. Später, wenn Sie lieber LX verwenden wollen, geben Sie ein:<sup>4</sup>

```
PIP CMDRUN.COM=LX.COM<cr>
```

### 4.2.2 Error Handler

Wenn der Kommando-Prozessor einen Befehl nicht ausführen kann, weil der ECP nicht gefunden wurde, oder weil der Erweiterte

---

<sup>2</sup>Diese Datei wird durch Einpacken häufig benötigter Befehle mittels LPUT selbst hergestellt

<sup>3</sup>Es muß im ROOT-Directory sein (letztes Element im Suchpfad). Wenn Sie das nicht selbst geändert haben, wird dies A0 sein, aber man kann mithilfe des PATH-Kommandos nachsehen, welches das ROOT-Directory ist.

<sup>4</sup>Sie sollten das alte CMDRUN.COM entfernt haben

Kommando-Prozessor den Befehl auch nicht bearbeiten kann, dann wird CPR einen Error-Handler aktivieren. Hochentwickelte Z-System-Errorhandler können Information darüber geben, was mit dem Befehl nicht stimmt und Ihnen ein Editieren der Kommandozeile erlauben, um irgendwelche Fehler zu beseitigen.

Jede Art von Fehlermeldung kann mit dem ERRSET-Utility (nicht im Lieferumfang enthalten) geladen werden. Die meisten Error-Handler werden sich automatisch installieren, wenn sie manuell als Kommando geladen werden (sie können unterscheiden, ob sie vom Anwender oder vom CPR) aufgerufen wurden. Das Z3PLUS Paket beinhaltet den ZERLSH Error-Handler in Kombination mit der History-Shell LSH, mehr darüber im nächsten Abschnitt.

### 4.2.3 Kommando Shells

Shells sind Tools, die automatisch ausgeführt werden, wann immer der Kommando-Prozessor alle übergebenen Kommandos ausgeführt hat — einschließlich Stapel-Befehlen von SUBMIT — und ist bereit für neue Befehlseingaben. Ist keine Shell geladen, dann wird der normale Prompt am Bildschirm auftauchen und der Anwender wird nach dem nächsten Befehl gefragt. Ist eine Shell aktiv, dann wird statt dessen diese ablaufen. Die meisten Shells erzeugen Kommandos *für* den Anwender und geben diese an den Kommando Prozessor weiter. Wir haben verschiedene Shells beim Z3PLUS-System mitgeliefert.

Die Shell Befehlszeile (manchmal im Verbund mit anderer Information, die von der Shell benutzt wird) liegt in einem Speicher-Puffer, genannt der “Shell-Stack”. Der Shell-Stack des Z3PLUS-Systems kann bis zu vier Shell-Befehlszeilen speichern. Das zuletzt abgelegte Kommando ist das aktive Shell-Kommando. Wenn diese Shell beendet wird (normalerweise durch einen gesonderten Befehl), wird ihre Befehlszeile vom Shell-Stack entfernt (in Anlehnung an die Assemblersprache: “gepoppt”), wobei die darunterliegende Shell zuoberst auf dem Stack zu liegen kommt. Sie wird dadurch aktiv. Dieser Mechanismus erlaubt die “Verschachtelung” von Shells.

#### ZERR und LSH - Error Handler und History-Shell

LSH (Log Shell) ist ein gutes Beispiel für das Shell-Konzept. Es ist außerdem besonders praktisch. Wenn LSH aktiv wird, wird ein

Prompt sichtbar, ähnlich dem CPR, aber zusätzlich mit der eingestellten Uhrzeit. Daran kann man erkennen, daß LSH läuft. Es gibt jedoch drei sehr wichtige Unterschiede:

1. Wenn man ein Kommando eingibt, steht ein sehr mächtiger Editor zur Verfügung, eine Art Textverarbeitung für die Kommando-Zeile. Diese kann vorwärts und rückwärts durchgeblättert werden und man kann Zeichen einfügen oder löschen.
2. Bei der Eingabe eines Kommandos wird durch Drücken der Return Taste die Kommando-Zeile in eine "History" Datei gespeichert, bevor der Befehl dem CCP übergeben wird.
3. Früher eingegebene Kommandos können aus der History-Datei geholt, nach Wunsch editiert und wieder ausgeführt werden.

Nachdem Sie gesehen haben, wie praktisch diese Features sind, wollen Sie möglicherweise LSH der Liste der Kommandos im STARTZ3P Startup Alias hinzufügen.

LSH wird mit ZERR als Error-Handler erst perfekt.

LSH ist, so wie es mit Z3PLUS geliefert wird ein Type-3 Programm (siehe S. 21), das an Adresse 8000H abläuft (32K). Als Resultat wird LSH beim Wiederaufruf eines Programms mit Hilfe des GO-Befehls nicht stören, es sei denn, das Anwenderprogramm war größer als 32K.

Die Installation von LSH geschieht mit LSHINST und definiert zum Beispiel die Mindestlänge eines Befehls, der gespeichert werden soll. Sie können aber auch schon so probeweise<sup>5</sup> arbeiten. Sinnvoll ist der Aufruf von einer RAMDISK aus, da die History-Datei dann beim Kaltstart stets leer ist. Geben Sie also ein:

```
LSH<cr>
```

Wir empfehlen Ihnen das jetzt mit den folgenden Beispielen zu üben. Geben Sie die die folgenden Kommandos nacheinander ein, durch Return getrennt:

```
ECHO Das ist der Befehl 1<cr>
DIR *.COM<cr>
ECHO Das ist der Befehl 3<cr>
LDIR Z3PLUS<cr>
```

---

<sup>5</sup>HELPLSH kann mit ESC-J zur Hilfestellung aufgerufen werden

Wenn der nächste Prompt erscheint, drücken Sie Control-E (für Rückwärts). Das vorangegangene Kommando erscheint. Geben Sie jetzt mehrfach Control-E ein. Dann versuchen Sie's mal mit Control-X (für Vorwärts); es wird Sie in der History vorwärts bewegen. Fahren Sie fort, bis Sie die Befehlszeile "ECHO Das ist der Befehl 3" am Prompt sehen.

Jetzt wollen wir das Kommando editieren. Die meisten der Editor-Tasten sind von WordStar her bereits bekannt. Control-S und Control-A springt ein Zeichen bzw. ein Wort nach Links, Control-D und Control-F entsprechend nach Rechts. Wenn Sie die "3" erreichen, geben Sie Control-G zum Löschen ein. Dann geben Sie eine "5" ein und drücken Return.

Nun versuchen wir einen Befehl zu finden. Beim Prompt geben Sie nur "ECHO" ein (ohne Return). Dann drücken Sie Control-QF. Beachten Sie, daß LSH das zuletzt benutzte Kommando findet, das mit "ECHO" angefangen hatte. Drücken Sie nochmals Control-QF. LSH findet das nächste zurückliegende Vorkommen. Diese Eigenschaft ist bemerkenswert praktisch!

Versuchen Sie jetzt den Seitenmodus von LSH. Drücken Sie zweimal kurz hintereinander ESC.<sup>6</sup>

Sie finden jetzt mehrere Eingabezeilen untereinander am linken oberen Bildschirmrand. Wie in WordStar kann man jetzt die Eingabezeilen bearbeiten und die geeignete Zeile schneller erkennen und wieder verwenden. Sehr praktisch ist auch die Hilfs-Funktion von LSH. Sorgen Sie dafür, daß die Datei HELPLSH.COM sich im Suchpfad befindet. Geben Sie jetzt ein:

ESC-J<cr>

Das erreichen Sie durch gleichzeitiges Drücken der ESCAPE-Taste<sup>7</sup> und der Taste "J".

### Die ZFILER Shell

ZFILER illustriert, wie eine Shell das Anwender-Interface vollständig verändern kann. Statt einer Befehlszeile stellt es eine "Point-and-

<sup>6</sup>Wenn Sie einen AMSTRAD Rechner benutzen, überprüfen Sie, ob Sie die ESC-Taste wirklich über SETKEYS mit dem Zeichen 27 (ESC) belegt haben. Die CPCs haben dort normalerweise das Bömbchen, das Zeichen 252

<sup>7</sup>beim PCW heißt diese Taste STOP



Shoot"- Umgebung zur Verfügung. Um nun ZFILER in Aktion zu sehen, geben Sie ein:

```
ZFILER<cr>
```

Sie brauchen LSH nicht zu "entfernen", bevor Sie ZFILER starten. Shells können unter dem Z-System "verschachtelt" werden. Als LSH lief, wurde ZFILER aufgerufen, dadurch wurde LSH inaktiv, wohingegen ZFILER läuft, ist jedoch automatisch wieder zurück, wenn ZFILER verlassen wird.

Sie werden eine ganzseitige Anzeige von Dateinamen des augenblicklichen Directorys erhalten, sowie einen Zeiger, der auf den ersten Dateinamen zeigt. Benutzen Sie die WordStar-Raute (Control-Tasten E, X, S, und D für hoch, runter, links, und rechts) um diesen zu bewegen. Positionieren Sie den Zeiger auf eine Text-Datei, wie LSH.COMD, ALIAS.COMD, oder ZFILER.COMD. Drücken Sie die "V"-Taste, um eine Datei anzusehen ("View"-Funktion). ZFILER hat viele solcher eingebauter Funktionen, und Sie können eine Liste davon sehen, indem Sie einen Slash ("/") für Hilfe eingeben. Drücken Sie nochmals diese Taste, um zum Dateimenü zurückzukehren.

Versuchen Sie jetzt mit "T" Dateien "anzukreuzen". Experimentieren Sie mit "Gruppen"-Operationen durch Drücken der "G" Taste. Wie Sie sehen können, sind Funktionen nicht nur auf eine einzelne Datei, sondern auf alle "angekreuzten" Dateien ausführbar. Geben Sie zum Probieren "Copy" ein durch Drücken von "C". Wenn Sie nach dem Ziel-Directory gefragt werden, geben Sie ein

```
A1<cr>
```

Beachten Sie, daß der Doppelpunkt nach Verzeichnisangaben innerhalb von ZFILER nicht unbedingt notwendig ist.

Überprüfen wir jetzt, ob die Dateien ordnungsgemäß kopiert wurden. "Loggen" Sie sich in das Directory A1 ein durch die "L" oder "Log" Taste. Geben Sie beim Prompt ein:

```
A1<cr>
```

Löschen Sie diese doppelten Dateien mit der "D" Taste. Kehren wir jetzt wieder ins Directory A0: zurück, aber auf eine leicht andere Art und Weise: Das Verzeichnis A0: hat den Namen: COMMAND. Drücken Sie die "L" Taste und geben Sie dann ein

COMMAND<cr>

Wie Sie sehen, können benannte Directories genauso wie Drive-User Bezeichnungen angegeben werden. Das gilt für das gesamte Z-System.<sup>8</sup>

Die eingebauten Funktionen sind nicht die einzigen, die ZFILER ausführen kann. Ein Set von "Macro"-Kommandos kann zusammen mit einer Anwender-Hilfsseite in der Datei ZFILER.CMD definiert werden. Wir haben eine ganz einfache Variante beigefügt, um die Möglichkeiten zu verdeutlichen. Bewegen Sie den Zeiger auf die Datei Z3PLUS.LBR. Drücken Sie jetzt die ESCAPE-Taste. Sie werden jetzt nach einem Makro gefragt. Später, wenn Sie wissen, welche Tasten welchen Macro-Funktionen zugeordnet sind, können Sie gleich die richtige Taste eingeben. Alternativ können Sie nochmals ESCAPE drücken (tun Sie das jetzt), und die Informations- Seite verschwindet. Sie werden sehen, daß das "L"-Makro Ihnen das Verzeichnis einer Library-Datei zeigt. Drücken Sie "L" und sehen Sie zu, was passiert.

ZFILER hat für Sie eine Befehlszeile generiert und dem Kommando Prozessor übergeben. Nachdem diese Kommando-Zeile ausgeführt ist, wird automatisch die ZFILER-Shell wiederaufgerufen. In diesem Fall, wußte sie ja, daß sie gerade ein Kommando beendet hatte, das eine Bildschirmausgabe verursacht hatte. Also wartete sie auf einen Tastendruck, bevor sie die Dateiliste restaurierte. Es ist sogar möglich, Makro-Kommandos auf Gruppen von angekreuzten Dateien anzuwenden. "Taggen" Sie ein paar Dateien und drücken Sie dann "G", um eine Gruppen-Operation zu starten. Wenn Sie nach der Operation gefragt werden, drücken Sie wie zuvor die ESCAPE-Taste, um einen Makro-Befehl auszuwählen. Wie vorher, werden Sie aufgefordert, eine Makro-Taste einzugeben. Sie können wieder sofort die Taste drücken, oder mit ESCAPE Hilfe anfordern. Geben Sie jetzt ein: "E". ZFILER wird jetzt automatisch eine Datei namens ZFILER.SUB im Directory A0: generieren und dann eine SUBMIT Stapel-Operation einleiten. Das "E"-Macro, wie Sie sehen werden, ist eine harmlose Kommando-Zeile, die ECHO benutzt, um zu zeigen, wie ZFILER den Namen der (mit Zeiger oder Tag markierten) bezeichneten Datei "parsen" kann.

Sie können jetzt ZFILER verlassen, indem Sie die "X" oder "eXit" Taste drücken<sup>9</sup>. Das wird die Shell beenden. Wenn LSH vor dem Aufruf von ZFILER aktiv war, dann wird LSH zurückkehren. Andernfalls, werden Sie den Prompt des Kommando-Prozessors selbst sehen.

<sup>8</sup>Das heißt, für den Z3PLUS Kommando Prozessor und Z-System-spezifische Utility Programme.

<sup>9</sup>AMSTRAD PCW Anwender: Es ist nicht die EXIT-Taste gemeint

Profis erkennen das Vorhandensein von LSH an der angezeigten Uhrzeit vor dem Prompt.

### Andere Shells

Es gibt eine ganze Reihe anderer Shells, die nicht im Standard Z3PLUS-Paket enthalten sind, die ebenso eine große Bandbreite von Funktionen bieten. Einige sind in der gleichen Familie wie ZFILER. Im Gegensatz zu ZFILER, der die Dateien auf dem Bildschirm zeigt und die verfügbaren Makro-Kommandos nur auf Wunsch anzeigt, präsentiert die MENU-Shell nur die Kommando-Optionen. Sie unterstützt mehrfache Verschachtelung von Befehlsmenüs. Die VMENU- und FMANAGER-Shells sind irgendwo in der Mitte zwischen MENU und ZFILER. Sie zeigen die Dateien in der oberen Hälfte des Bildschirms und die Kommando-Optionen in der unteren.

Eine andere Art Shell ist die Patch-Shell ZPATCH. Sie erlaubt Ihnen, den Maschinencode einer Datei zu editieren. Zum Beispiel möchten Sie die Konfigurations-Optionen eines Programms anpassen. Aus ZPATCH heraus können Sie das modifizierte Programm aufrufen, um das Ergebnis Ihrer Arbeit zu begutachten und dann automatisch nach ZPATCH zurückzukehren, wobei der Cursor auf exakt demselben Byte steht wie vor dem Aufruf.

Ein weiteres Set von Shells (SH, GETVAR, FOR-NEXT) gestatten das Erzeugen und Verwenden von "Shell Variablen", ziemlich ähnlich den "Environment Variablen" in MS-DOS.

## 4.3 Andere Z-System Tools

Es gibt noch viele, viele andere Z-System-Tools, bei weitem zu viele, um sie alle hier beschreiben zu können. In der Tat gibt es wahrscheinlich nicht einen Z-System User, egal welchen Expertenlevels, der alle kennt! Damit Sie damit umgehen lernen, werden wir ein paar wichtige beschreiben. Bis jetzt mußten Sie diese noch nicht alle von Ihrer Z3PLUS-Arbeitsdiskette (S. 6) in Ihr Directory A0: kopieren. Sie können diese entweder kopieren, oder von Ihrer Arbeitsdiskette ausführen.

### 4.3.1 ARUNZ

Ein Z-System Alias oder Alias-Script ist eine Sequenz eines oder mehrerer Kommandos, die mit einem einzigen Befehle aufgerufen werden kann. Wie wir in Abschnitt 3.1 (S. 11) gesehen haben, sind alleinstehende Aliase kurze COM-Dateien, die man mit SALIAS, dem Bildschirm-orientierten Alias-Editor, erzeugen kann.

ARUNZ bietet einen alternativen Weg, um Aliase zu definieren und anzuwenden. Statt jeden Alias in eine eigene Datei zu packen, erlaubt ARUNZ, die Definition zahlreicher Aliase in einer einzigen Datei namens ALIAS.CMD. ARUNZ extrahiert einen bestimmten Alias-Script aus der ALIAS.CMD Datei, expandiert eventuelle Symbolische Parameter im Script, und übergibt die resultierende Kommando-Zeile an den Kommando-Prozessor.

Die Datei ALIAS.CMD ist eine gewöhnliche Textdatei, die mit einem beliebigen Text-Editor oder Textverarbeitung (im Non-Document Mode) erstellt werden kann. Da alle Scripts in einer einzigen Datei kombiniert werden können, benötigen sie sehr wenig Disk-Speicherplatz. Auf diese Weise können Dutzende oder sogar hunderte verwendet werden und dadurch Ihren Computer-Alltag erheblich erleichtern.

ARUNZ-Aliase werden in der Form aufgerufen:

```
ARUNZ ALIASNAME BEFEHLS-ARGUMENTE<cr>
```

ALIASNAME spezifiziert, welcher der Scripts in ALIAS.CMD gemeint ist und BEFEHLS-ARGUMENTE liefern weitere Informationen, die von den Scripts erwartet werden, wie etwa Dateinamen. Die wirkliche Leistungsfähigkeit von ARUNZ kommt zutage, wenn man ihn in CMDRUN.COM umbenennt und er als Erweiterter Kommando-Prozessor (siehe S. 26) fungiert. Dann kann das Kommando einfach so eingegeben werden:

```
ALIASNAME BEFEHLS-ARGUMENTE<cr>
```

Ein Beispiel macht das möglicherweise etwas klarer. Eine Zeile in der Datei ALIAS.CMD lautet:

```
D=SD      sdz $td1$tu1:$tn1*.$tt1* $-1
```

Das mag ziemlich verboten aussehen mit all diesen Dollar-Zeichen, aber, wenn wir es Stück für Stück auseinandernehmen, wird es nicht so

schlimm. Zuerst kann der Alias als erster mit “D” oder “SD” aufgerufen werden, es ist sozusagen sein Name. ARUNZ gestattet eine Vielzahl von Namen für einen einzigen Script. Das Programm, das jetzt von diesem Alias aufgerufen wird, ist SDZ, das Super-Directory Programm.

Der interessante Teil dieses Scripts ist die Art, wie die Befehlsargumente verarbeitet werden. Um zu illustrieren, was geschieht, nehmen wir an, daß wir das folgende Kommando eingegeben haben:

```
D S.C<cr>
```

Der Parameter-Ausdruck “\$d1” steht für das im ersten Token angegebene Laufwerk. In unserem Beispiel, ist das erste Token “S.C”. Da dort kein Laufwerk angegeben wurde, wird das aktuelle Laufwerk A: angenommen. Gleichermaßen bedeutet “\$u1” die im ersten Token angegebene User-Nummer. Wiederum war auch hier keine angegeben, daher wird die aktuelle User-Nummer 0 angenommen. Als nächstes gibt es da einen Doppelpunkt. Bisher liest sich die Kommando-Zeile “SDZ A0:”.

Der nächste Parameter, “\$tn1”, indiziert, daß der Dateiname Teil des ersten Tokens ist. In unserem Beispiel, ist das “S”. Die nächsten Zeichen im Script sind ein Asterisk (\*) und danach ein Punkt. Dann haben wir den Parameter “\$tt1”, der für den Datei-Typ im ersten Token, oder “C”. Dann gibt es noch einen weiteren Asterisk (\*) im Script. Bisher liest sich also unsere Kommando-Zeile: “SDZ A0:S\*.C\*”.

Der letzte Teil des Scripts ist der Parameter Ausdruck “\$-1”. Das bedeutet das gesamte Befehls-Argument ohne das erste Token (“\$-2” würde die ersten zwei Tokens auslassen). Im Beispiel, gibt es keine anderen Tokens, aber man hätte eine Option von SDZ angeben können, wie etwa “/C”, um die Dateigröße als Record Count statt in K anzugeben. Auf diese Weise wird das Kommando, das als

```
D S.C<cr>
```

eingegeben wurde, dank ARUNZ, zu:

```
SDZ A0:S*.C* <cr>
```

Versuchen Sie den Alias-Befehl und sehen Sie, was passiert. Das Script zeigt uns einen sehr bequemen Weg, automatisch die Dateiangaben für das SDZ-Programm in eine “Wildcard”-Angabe zu verwandeln,

womit wir uns das ärgerliche Eingeben der Joker-Zeichen (\*.\*) ersparen. Im Beispiel, erhalten wir alle Dateien, deren Name mit "S" beginnt und deren Typenbezeichnung mit "C" beginnt, einschließlich SALIAS.COM, STARTZ3P.COM, und SDZ.COM.

Um mehr darüber zu lernen, wie man ARUNZ-Aliase schreibt und einsetzt, konsultieren Sie die Querverweise in der Bibliographie. Aber beachten sie, daß die Version von ARUNZ, die mit Z3PLUS geliefert wurde, neuer ist, als die dort beschriebene. Viele neue Parameter wurden hinzugefügt und ein paar alte Parameter wurden verändert. Lesen Sie daher die Update-Dokumentation, die im Z3PLUS Paket enthalten ist.

### 4.3.2 HELP

Die meisten Z-Tools geben eine herbe Erinnerung an die Befehlszeilen-Syntax aus, wenn Sie sie nur mit einem doppelten Schrägstrich (SLASH) aufrufen, wie in

```
Z3PLUS //<cr>
```

Viel ausführlichere on-line Information kann man durch das Z-System HELP Utility erhalten. Es stellt eine organisierte Methode dar, um (baumstrukturiert) nach speziellen Textdateien des Typs HLP zu suchen und die darin enthaltene Information anzuzeigen. Es ähnelt der CP/M-Plus HELP Einrichtung, ist aber ungleich leistungsfähiger. Das Programm hieß ursprünglich HELP, um aber eine Verwechslung mit dem CP/M Programm dieses Namens zu vermeiden, haben wir es für das Z3PLUS-System in ZHELP umbenannt. ZHELP ruft man wie folgt auf:

```
ZHELP<cr>
ZHELP Dateiname<cr>
```

Die erste Form benutzt eine Datei mit dem Namen HELP.HLP; die zweite verwendete Form war mit einem Namen "filename" angeandt worden. Zum Beispiel, wenn Sie dafür SALIAS einsetzen und es gibt eine Hlp-Datei dafür, geben Sie ein

```
ZHELP SALIAS<cr>
```

Help-Dateien für eine Vielzahl anderer Z-System-Programme können in den Z-Nodes gefunden werden <sup>10</sup>. Sie können auch Ihre eigenen HLP-Dateien schreiben.

### 4.3.3 Library Tools

Z-System Library Tools machen es möglich, einen ganzen Set von Dateien in eine einzige Library-Datei zu packen und dann die einzelnen Library-Mitglieder nach Bedarf zu verwenden. Das ist effektiv, um kurze und in Zusammenhang stehende Dateien zu organisieren und Directory- und Dateikapazität der Diskette zu sparen. Programme wie Z3PLUS und JETLDR sind in der Lage, auf mehrere Mitglieder einer Library schneller zuzugreifen, als individuelle Dateien gelesen werden könnten. Der Einsatz von Libraries in Mailboxen (RASs oder RCPMs) ist sehr verbreitet. Die grundlegenden Z3PLUS System-Dateien zum Beispiel, befinden sich alle in der Library Z3PLUS.LBR.

Die zwei Mehrzweck-Library-Tools sind NULU (new Library utility) und VLU (Visual Library Utility). Diese zwei Programme können eine Library erzeugen, Dateien hineinstellen und extrahieren, Textdateien auf dem Bildschirm "typen", und besitzen eine Reihe anderer Funktionen. NULU ist ein generisches CP/M Programm, es läuft also unter dem normalen CP/M ebenso und kann von Mailboxen downgeloadet werden; VLU ist ein Z-System-spezifisches Tool und arbeitet ähnlich wie die ZFILER-Shell. Wir haben dem Z3PLUS Paket eine spezielle, für CP/M Plus gepatchte Version beigefügt, die die freie Disk-Kapazität korrekt anzeigt. Andere Versionen sollten unter Z3PLUS nicht verwendet werden.

Library Tools, die eingeschränktere Funktionen besitzen, sind:

LPUT	Datei in eine Library packen
LGET	Datei aus Library holen
LDIR	Directory einer Library
LX	Ausführen von Member-Dateien
LT	Library-Text-Datei ansehen

---

<sup>10</sup>In Europa gibt es bisher nur einen Z-NODE: ZNODE 51 in Neufahrn bei München 08165/60041 Baudrate 300 1200 2400 8 N 1 SIG-OP Helmut

#### 4.3.4 Datei-Kompression

Kompressions- und Dekompressions-Tools erlauben das “Crunchen” von Files auf ein Bruchteil Ihrer Original-Größe für kompaktere und schnellere Übertragung per Modem. Gecrunchte Dateien haben ein “Z” als zweiten Buchstaben des Dateityps, der sogenannten Extension (oder ZZZ als Typ, wenn keine Extension vorhanden war). CRUNCH und UNCRUNCH machen die Arbeit. Auch sie sind im Z3PLUS-Paket enthalten.

LT.COM ist sehr vielseitig. Es wurde mit den Library Tools gelistet, weil es Textdateien in Libraries darstellen kann. Es kann auch einzelne Dateien anzeigen und in beiden Fällen kann die Datei gecruncht sein als auch unkomprimiert sein. Es erkennt sogar automatisch, ob “Un-crunchen” notwendig ist. Um ein Library-Mitglied auf dem Bildschirm auszugeben, benutzen Sie die Befehlsform:

```
LT LBRNAME FILENAME.TYP<cr>
```

Um eine einzelne Datei anzuzeigen, benutzen Sie die Form:

```
LT FILENAME.TYP<cr>
```

#### 4.3.5 Benannte Directory Tools

Die Fähigkeit des Z-Systems, Namen mit Directories zu assoziieren, machen das Leben einfacher, vor allem, wenn Ihr Computer eine Hard-disk besitzt. EDITNDR<sup>11</sup> ist ein Hilfsmittel, um die Zuordnung von Directory-Namen zu editieren. Es kann sowohl interaktiv, als auch aus der Kommando-Zeile verwendet werden. Wenden wir es zuerst interaktiv an. Geben Sie ein

```
EDITNDR<cr>
```

Sie erhalten einen Prompt, der Sie einlädt, für Hilfe ein “?” einzugeben. Warum sollten wir diese Einladung nicht akzeptieren! Geben Sie jetzt ein RETURN ein. EDITNDR zeigt Ihnen jetzt die augenblicklich zugewiesenen Namen. Geben Sie jetzt ein

---

<sup>11</sup>früher hieß es einmal EDITND. Wir haben es umbenannt, um eine einheitliche Form mit SAVENDR zu erreichen.



```
A1:TEXT<cr>
```

Das weist dem User 1 auf Laufwerk A: den Namen TEXT zu. Drücken Sie <cr> um die neue Anzeige zu erhalten. Vielleicht sehen Sie die Zuweisung von A1: nicht, weil es nicht bei den anderen Zuordnungen von Laufwerk A: steht. Mit der Eingabe von "S" sortieren Sie die Liste und geben Sie noch einmal ein <cr>. Jetzt sind die Namen in geordneter Reihenfolge. Wenn Sie den eingebauten Hilfe-Bildschirm ansehen, werden Sie sehen, daß EDITNDR sehr flexibel ist in der Syntax, die es akzeptiert.

Die Datei ALIAS.CMD enthält einen Alias, um die Zuweisung von Directory-Namen zu vereinfachen. Sie geben dazu nur ein

```
NAME DU:DIRNAME<cr>
```

wobei DU den Userbereich (drive/user) angibt, der den Namen DIRNAME erhalten soll. Wenn "DU:" weggelassen wird, wird dem aktuellen Verzeichnis der Name zugewiesen. Versuchen Sie die Eingabe

```
NAME SYS<cr>
```

und beobachten Sie, wie der Kommando-Prompt sich verändert. Sie können den Inhalt von ALIAS.CMD betrachten, um zu sehen, wie dieser Alias funktioniert.

Die mit EDITNDR gemachten Directory-Zuweisungen sind temporär. Sie können das SAVENDR Utility verwenden, um die Zuweisungen in eine Datei zu speichern. Diese Datei kann durch den STARTZ3P-Alias geladen werden, um Ihre Namen wieder zu installieren. Das Kommando

```
SAVENDR SYS<cr>
```

zum Beispiel, erzeugt im aktuellen Verzeichnis eine Datei namens SYS.NDR. Diese Datei kann von Z3PLUS geladen werden durch den Befehl

```
Z3PLUS SYS.NDR<cr>
```

Z3PLUS kann viele Dateien gleichzeitig laden, daher können Sie das kombinieren mit dem Laden der Z3T-Datei wie folgt:

```
Z3PLUS myterm.Z3T SYS.NDR<cr>
```

Wenn Sie diese zwei Dateien mit LPUT in Z3PLUS.LBR stellen, können Sie diese sogar noch schneller laden mit

```
Z3PLUS Z3PLUS.LBR myterm.Z3T SYS.NDR<cr>
```

Am besten benennen Sie die beiden Dateien in DEFAULT.Z3T und DEFAULT.NDR um, bevor Sie sie in Z3PLUS.LBR packen. Dann lädt Z3PLUS sie automatisch beim Start und Sie müssen im STARTZ3P-Alias kein neues Kommando einfügen.

### 4.3.6 Andere Tools

Eine Anzahl anderer Utility-Programme können auf Ihrer Z3PLUS-Diskette enthalten sein. Sie sollten selbst mit Ihnen experimentieren. Geben Sie nach dem Namen “//” ein, um die eingebaute Hilfe zu bekommen. Wenn es eine (HLP)-Datei mit demselben Namen wie die Befehlsdatei gibt, können Sie ZHELP benutzen, um mehr über dieses Kommando zu lernen.

## 4.4 Kommando-Hierarchie

Der Z3PLUS Kommando-Prozessor ist hochentwickelt in der Art, wie er Befehle abarbeitet. Zwei Schritte sind es bei der Befehlsabarbeitung. Zuerst muß von irgend einer Kommandoquelle ein Kommando kommen. Dann muß das Kommando in entsprechender Weise bearbeitet werden. Wir wollen jetzt diese zwei Aspekte der Befehlsbearbeitung kurz beschreiben.

### 4.4.1 Kommando Akquisition

Der Z3PLUS Kommando-Prozessor erhält sein nächstes Kommando nach der folgenden Hierarchie:

1. aus dem Mehrfach-Kommandozeilen-Puffer
2. aus einer SUBMIT-Datei
3. durch ein Shell-Kommando
4. durch Anwender-Eingaben

Kommandos im Mehrfach-Kommandozeilen-Puffer haben die höchste Priorität. Ist der Kommandozeilen-Puffer leer, sieht der Kommando-Prozessor nach, ob ein SUBMIT läuft, in welchem Fall er seine nächste Kommandozeile von der SUBMIT-Datei erhält. Das bedeutet, daß ein SUBMIT-Job sich tatsächlich wie eine sehr lange Kommandozeile verhält. Ist der Befehlsstrom erschöpft, kehrt der Kommando Prozessor zu einer eventuell gewählten Shell zurück. Nur wenn es keine Shell gibt, kehrt der Kommando Prozessor als letzte Anlaufstelle zum Anwender zurück!

#### 4.4.2 Kommando Auflösung

Wenn Z3PLUS das nächste Kommando erhält, löst es den Befehl nach folgender Hierarchie auf:

1. Durchsuchen der FCP Kommandos
2. Durchsuchen der RCP Kommandos
3. Durchsuchen der eingebauten Kommando-Prozessor Befehle
4. Suchen einer Datei (COM, SUB, PRL) entlang des Suchpfades
5. Aufrufen des Erweiterten Kommando-Prozessors
6. Aufrufen des Error Handlers

Die Dateitypen, die entlang des Suchpfades gesucht werden, werden von der CP/M-Plus Suchordnung kontrolliert.<sup>12</sup> Wenn die Reihenfolge zum Beispiel erst COM, dann SUB ist, dann wird erst der gesamte Pfad erst nach einer COM-Datei abgesucht. Wird keine passende COM-Datei gefunden, wird der Pfad nochmals nach einer SUB-Datei durchsucht.

Einige Faktoren können die Auflösungshierarchie beeinflussen. Ist der gegenwärtige Flow State "False", werden nur FCP und Shell Kommandos gescannt. RCP, CPR-residente Kommandos, transiente Kommandos und der Erweiterte Kommando Prozessor (ECP) werden ignoriert. Shells werden immer noch ablaufen.

Verschiedene Voranstellungen können die Hierarchie der Auflösung verändern. Ein Slash Prefix "/" weist einen Befehl unmittelbar dem

---

<sup>12</sup>Diese Reihenfolge und welche Dateitypen das beinhaltet, kann mit dem CP/M-Plus Utility SETDEF geändert werden.

Erweiterten Kommando Prozessor zu. Wenn Sie wissen, daß Ihr Kommando vom ECP ausgeführt werden muß, dann können Sie sich die Zeit sparen, die andernfalls bei der Durchsuchung des Pfades nach einem transienten Kommando verloren geht. Der Slash Prefix kann auch verwendet werden, wenn sowohl transiente, als auch ECP-Versionen ein und desselben Kommandos existieren, und Sie die ECP Version haben möchte.

Ein Prefix, der nur aus einem Doppelpunkt, oder aus einem Punkt besteht, veranlaßt den Z3PLUS Kommando-Prozessor, residente Kommandos zu überspringen.<sup>13</sup> Außerdem fügt es das aktuelle Verzeichnis auch dem Suchpfad hinzu. Wird ein transientes Kommando nicht gefunden, wird immer noch der ECP will aufgerufen. Diese Prefixe können benutzt werden, wenn der Suchpfad das aktuelle Verzeichnis (\$\$) nicht beinhaltet und Sie wissen, daß es das Kommando gibt. Man kann sie auch benutzen, um die Ausführung eines transienten Programms mit demselben Namen wie ein residentes im RCP oder CPR zu erzwingen.

Ein expliziter Directory Prefix der Form D:, U:, DU:, oder DIR: hat denselben Effekt wie der Doppelpunkt oder der Punkt, außer daß das Kommando *nur* im angegebenen Directory gesucht wird und der ECP nicht aufgerufen wird, wenn das Kommando nicht gefunden wurde.

Tritt ein Fehler auf, versucht der Kommando Prozessor erst, einen Error-Handler Wurde keiner angegeben oder keiner gefunden wurde, schreibt es die verbleibende Kommando-Zeile in den Mehrfach-Kommandozeilen-Puffer, gefolgt von einem "?".

---

<sup>13</sup>Das gilt nicht für residente Kommandos im FCP. Bei *jedem* Kommando werden Flow Control Kommandos mitgeprüft, egal, welcher Prefix vorliegt (ein Doppelpunkt, ein Slash, DU:, oder DIR:).

# Kapitel 5

## Wie man mehr aus Z3PLUS herausholt

In den vorangegangenen Kapiteln haben Sie die grundlegenden Funktionen des Z-Systems und der besonderen Implementation in Z3PLUS kennengelernt. Was Sie gesehen haben, ist jedoch nur die Spitze eines Eisbergs! Z3PLUS bietet eine Unzahl an Möglichkeiten und wir werden versuchen, sie Ihnen in diesem Kapitel vorzustellen. Tatsächlich gibt es sicher Anwendungen für Z3PLUS, die nicht einmal uns eingefallen wären, daher können wir hier nicht Anspruch auf erschöpfende Behandlung erheben.

### 5.1 Alternativer Aufruf von Z3PLUS

Bisher haben Sie die einfachste Methode kennengelernt, Z3PLUS zu starten. In diesem Abschnitt werden wir einige andere Wege diskutieren, das Z-System zu laden und einige andere Versionen des Systems, die fast ebenso leicht geladen werden können, wie die bereits Gesehene.

#### 5.1.1 Alternative Lademethoden

Sie haben die Standardmethode, Z3PLUS zu starten, kennengelernt, die in der Eingabe des einfachen Befehls besteht

```
Z3PLUS<cr>
```

Z3PLUS kann auch auf folgende Weise geladen werden:

- aus einem SUBMIT-Script
- aus einer CP/M-Plus Mehrfach-Kommandozeile

Eine Z3PLUS-Kommandozeile kann in einer SUBMIT-Datei enthalten sein, aber es muß die letzte Zeile der SUB-Datei sein. Andernfalls wird bei dem Versuch, Z3PLUS zu laden, die Fehlermeldung “First remove: GET” ausgegeben.<sup>1</sup> Das Kommando Z3PLUS in einer SUB-Datei kann nützlich sein für den automatischen Aufruf von Z3PLUS als Teil Ihrer CP/M-Plus Kaltstart-Prozedur.

Z3PLUS kann unter CP/M-Plus auch als Teil einer Mehrfach-Kommandozeile gestartet werden. Zum Beispiel:

```
Z3PLUS!MYSTART<cr>
```

startet Z3PLUS und dann das Kommando MYSTART. In diesem Fall wird das Default Start-Kommando (normalerweise STARTZ3P) nicht ausgeführt. Statt dessen werden die in der CP/M-Plus Mehrfach-Kommandozeile verbliebenen Befehle in das Z-System-Format konvertiert und in den Z-System Puffer für die Mehrfach-Kommandozeile kopiert, von wo aus sie ausgeführt werden. Diese Technik kann sehr nützlich sein, wenn Sie mit einer Sonderanpassung starten wollen. Beachten Sie, daß der Kommandozeilen-Separator von CP/M-Plus, das Ausrufezeichen (“!”), nur in diesem speziellen Fall verwendet werden sollte, das heißt, wenn die Kommandozeile vom CP/M-Plus Kommando-Prozessor bearbeitet wird, bevor Z3PLUS aktiv wird. Beachten Sie auch, daß diese Lade-Technik nicht mit der SUBMIT Lade-Technik kombinierbar ist.

### 5.1.2 Alternative Default Systeme

Alle Diskussionen in diesem Handbuch haben bisher das “Standard” Default System behandelt. Es gibt ja zwei andere Default Systeme, die

---

<sup>1</sup>Das kommt von der CP/M-Plus Resident System Extension (RSX). GET bleibt geladen, bis der letzte Befehl einer SUBMIT-Abarbeitung gelesen wurde. Z3PLUS kann nicht geladen werden, während die RSXe GET, PUT, oder DosDisk aktiv sind. Andere RSXe werden automatisch entfernt.

fast genauso einfach geladen werden können. Eine dieser Versionen ist eine größere Version von Z3PLUS, die mit dem Befehl

```
Z3PLUS LARGE<cr>
```

geladen wird. Dieses System benutzt mehr Speicher des Betriebssystems und bietet entsprechend mehr Möglichkeiten. Es ist fast das gleiche, wie das Standard-System und unterscheidet sich nur durch einen 25% größeren RCP-Puffer (vier zusätzliche Records oder 512 weitere Bytes), so daß mehr residente Kommandos unterstützt werden können. Anwender mit Disketten-Laufwerken geringer Kapazität oder langsamen Laufwerken werden dieses System vorziehen. Auf Seite 60 finden Sie detailliertere Information darüber, wie dieses System definiert ist und welche Moduln geladen werden.

Ein zweites alternatives Default-System ist eine kleinere Version von Z3PLUS, die mit dem Befehl

```
Z3PLUS SMALL<cr>
```

geladen wird. Dieses System verwendet weniger TPA und hat entsprechend weniger Features. In der Tat ist es so spartanisch, wie ein Z-System nur werden kann. Es besitzt keine Puffer für einen RCP, FCP, oder ein NDR. Es gibt dem Anwender fast 3K mehr TPA für Anwendungsprogramme und bietet immer noch viele wertvolle Z-System-Features: Shells, Error-Handler, Erweiterten Kommando-Prozessor, Aliase, einen dynamischen Suchpfad, TCAP-Unterstützung. Siehe auch Seite 60 wegen detaillierterer Information darüber, wie dieses System definiert ist und welche Moduln geladen werden.

## 5.2 Fliegender Wechsel der Systeme

Wir haben die Möglichkeiten diskutiert zwischen verschiedenen Z3PLUS Konfigurations-Optionen, als ob man nur auswählen und für die Dauer der Sitzung verwenden könnte. Offensichtlich könnte man jederzeit ins CP/M gehen durch Eingabe von

```
Z3PLUS OFF<cr>
```

und dann wieder ganz von vorne anfangen mit einer neuen Version von Z3PLUS. Das ist jedoch nicht notwendig. Neue Z3PLUS-Konfigurationen können geladen werden, während Z3PLUS schon läuft.

Wenn Z3PLUS vom CP/M-Plus Eingabe-Prompt geladen wird (ein "Kalt-"start), nehmen alle diese Werte, die in der Z3P Deskriptordatei beschrieben sind, Gestalt an <sup>2</sup>. Auf der anderen Seite, wenn Z3PLUS aus einem bereits laufenden Z3PLUS-System ("Warm-"Start) gestartet wird, werden alle System-Informationen mit Ausnahme der Adressen und Größen der System-Moduln bewahrt. Shells und Error-Handler laufen im neuen System weiter; der Suchpfad, Werte in den User-Registern, System-Dateinamen, etc. ändern sich nicht. Die Inhalte eines Puffers der Mehrfach-Kommandozeile werden in das neue System übernommen. Das bedeutet, die neue Konfiguration kann als Teil einer komplexen Befehls-Sequenz geladen werden, sogar eine mit aktiven Fluß-Kontroll-Operationen.<sup>3</sup> Das System kommt dabei nicht aus dem Takt!

Warum würden Sie Systeme so auf diese fliegende Art wechseln wollen? Der übliche Grund ist der Speicherbedarf. Gäbe es Speicher unbegrenzt, würden Sie die größte, leistungsfähigste Version von Z3PLUS, die Sie kriegen könnten, laden und dort für alle Zeiten lassen. Im wirklichen Leben, besonders auf einem CP/M-System mit seinem Adressraum von nur 64K, müssen Zugeständnisse gemacht werden.

Die meisten Leute möchten wahrscheinlich die meiste Zeit eine relativ leistungsfähige Version des Z-Systems (so wie das Standard Default System) benutzen. Es wird jedoch Gelegenheiten geben, wenn ein Anwendungs-Programm aufgerufen wird, das mehr Speicher benötigt, als diese Konfiguration erlaubt. Dann wird man ein kleineres System laden wollen.

Nehmen wir zum Beispiel an, wir hätten ein Anwendungsprogramm namens "BIGPROG", das viel mehr Speicher benötigt (vielleicht ist es eine Datenbank). Wir könnten dann den Befehl eingeben

```
Z3PLUS SMALL;BIGPROG . . . ;Z3PLUS<cr>
```

Diese Befehlszeile installiert erst das "Small" Z3PLUS-System mit seiner größeren TPA. Dann führt es BIGPROG aus. Schließlich, nachdem BIGPROG beendet wurde, wird das Standard Z3PLUS-System wieder geladen. Außer dem Zeitverlust beim Systemwechsel, gibt es eigentlich keinen Nachteil, wenn man vorübergehend Z-System Features aufgibt, die von dem zu startenden Anwendungsprogramm nicht

---

<sup>2</sup>Diese Dateien werden im nächsten Abschnitt beschrieben.

<sup>3</sup>Vorausgesetzt natürlich, daß keine Fluß-Kommandos bearbeitet werden müssen, während ein System ohne FCP aktiv ist.



benötigt werden.<sup>4</sup> Im obigen Beispiel ist die Vollversion des Z-Systems wieder zurück, bevor der Prompt den Anwender nach seiner nächsten Eingabe fragt. Für den Anwender war er jedenfalls immer da!

Der Vorgang des Systemwechsels kann in verschiedenen Stufen automatisiert werden indem man den Vorteil der Z-System-Möglichkeiten nutzt. Nehmen wir zum Beispiel an, BIGPROG.COM sei im Directory BIGDIR: und wir würden es in BGPRG.COM umbenennen. Dann, wenn wir die folgenden Script-Definition in unser ALIAS.CMD eingeben

```
BIGPROG      z3plus small;bigdir:bgprg $*;z3plus
```

können wir unser Kommando wie gewohnt eingeben:

```
BIGPROG FILENAME ... <cr>
```

Der Erweiterte Kommando-Prozessor ARUNZ übernimmt die automatische System-Umschaltung für uns. Noch geschliffenere Scripts können sogar prüfen, ob die TPA schon groß genug für ein Programm ist und nur dann einen Systemwechsel einleiten, wenn es notwendig ist. Es gibt keine Grenze, was Genialität mit Hilfe der Flexibilität erreichen kann, die Z3PLUS bietet!

Es gibt zwei kleine Punkte, die wir hier erwähnen sollten. Der erste ist, sich zu merken, daß Z3PLUS bei dem Versuch, das System zu wechseln, nicht einfach aufgibt, wenn es auf einen Fehler trifft. Es gibt eine Fehlermeldung auf dem Bildschirm aus, der die Natur des Problems angibt und *dann ruft es den Error-Handler auf*. Auf diese Weise, können Sie alle Fehler korrigieren, bevor die Befehlssequenz weitermachen und möglicherweise eine Katastrophe (oder zumindest große Unannehmlichkeiten) verursachen kann.

Als zweiter Punkt ist zu bemerken, daß , wenn ein neues Z3PLUS-System geladen werden soll, der Puffer der neuen Mehrfach-Befehlszeile groß genug sein muß , die übriggebliebenen Kommandos aus dem Puffer der original Mehrfach-Befehlszeile zu übernehmen. Wenn die schwebenden Kommandos nicht reinpassen, wird das neue System nicht geladen; statt dessen wird der Error-Handler aufgerufen. Sie können sich dann

---

<sup>4</sup>Wenige Anwendungsprogramme machen von Z-System-Features Gebrauch. WordStar Release 4 ist die Ausnahme unter den Anwendungen aus Großen Software-Häusern; es kann benannte Verzeichnisse verwenden. In diesem Fall, möchten Sie vielleicht ein Z3PLUS System generieren, das nur einen NDR-Puffer besitzt.

entscheiden, wie Sie die Situation behandeln wollen (z.B. einige der schwebenden Kommandos oder die ganze Operation abubrechen).

## 5.3 Anpassen Ihres Z3PLUS Systems

Es gibt viele Möglichkeiten, Ihr Z3PLUS-System anzupassen und es auf Ihre Bedürfnisse und Ihren Geschmack einzustellen. Wir wollen zwei Arten von Änderungen in Betracht ziehen, jene, welche Sie mehr oder weniger vorübergehend anlegen und jene, die Sie eher permanent machen, etwa durch Eingabe einer komplett neuen Konfiguration.

### 5.3.1 Vorübergehende Änderungen

Nachdem Sie sich mit den Grundlagen des Z3PLUS-Systems etwas vertrauter gemacht haben, möchten Sie zweifellos eine Reihe der System-Charakteristika ändern. Einige dieser Änderungen werden vorübergehender Natur sein. Sie möchten zum Beispiel den Command Search Path (Suchpfad) ändern, um ein Directory mit solchen Dateien zu erfassen, die Sie nicht ständig benützen — und die daher normalerweise nicht im Suchpfad liegen — aber die Sie für die momentane Aufgabe brauchen. Dazu benützen Sie das PATH-Utility. Haben Sie ihre Arbeit damit beendet, benützen Sie wiederum dieses Utility, um den Pfad auf seine ursprüngliche Einstellung zurückzusetzen.

Andere Parameter, die oft temporär geändert werden, sind die Charakteristika von Drucker und CRT (Cathode Ray Tube= Kathodenstrahlröhre=Bildschirm). Viele Z-System-Programme werden sich automatisch auf diese "Environment" Werte einstellen. Zum Beispiel, wenn Sie mit einem Zeilenabstand von 6 Zeilen pro Zoll drucken, könnten Sie die Printerdaten auf 66 Zeilen pro Seite und 58 Textzeilen einstellen. Wenn Sie mit 8 Zeilen pro Zoll drucken, benützen Sie statt dessen Werte von 88 und 78. Man verwendet dafür das CPSET-Utility.

Manchmal möchten Sie derartige Veränderungen über einen gewissen Zeitraum beibehalten, aber nicht fest einbinden. In diesem Fall, können Sie die benötigte Befehlszeile Ihrem Startup Alias beifügen. Die Veränderungen treten dann automatisch in Kraft, wenn Sie das Z3PLUS-System erstmals starten (und nach jedem Kaltstart).

### 5.3.2 Sonderanpassungen

Obwohl wir, wie wir eben beschrieben haben, leicht vorübergehende Veränderungen in einigen der System-Charakteristika durchführen können, wird wahrscheinlich der Wunsch nach einer dauerhaften Veränderung auftauchen. Es ist auch wahrscheinlich, daß Sie irgendwann System-Charakteristika verändern wollen, die nicht mit den Utility-Programmen verändert werden können, wie zum Beispiel die Größe der Systempuffer (RCP, FCP, und NDR). Für sehr viele Anwender werden die Default System-Konfigurationen nicht völlig Ihren Vorstellungen entsprechen. Zum Beispiel wenn Sie eine Harddisk besitzen, wird der Standard NDR Puffer mit seiner Kapazität von 21 Namen manchmal nicht reichen und Sie werden die Kapazität auf 28, 42, oder sogar mehr Namens-Einträge erhöhen wollen. Wir schätzen uns glücklich, Ihnen mitteilen zu können, daß *alle diese Dinge machbar sind, und noch dazu ziemlich einfach*.

Die Charakteristika des Z3PLUS-Systems sind definiert durch "Descriptor"-Files mit dem Dateityp Z3P. Diese Files sind gewöhnliche Textfiles, die mit jedem x-beliebigen Text-Editor bearbeitet werden können (im NON-DOKUMENT Modus = ASCII). Die Z3P-Files sind aufgebaut wie Assembler Quelldateien und enthalten Definitionen von Werten für die konfigurierbaren Z-System-Environment-Variablen, zusammen mit optionalen Kommentaren.

Eine Definitionszeile hat das Format:

```
VARIABLENNAME [ = | EQU ] WERT [;Kommentar]
```

Die eingeklammerten Angaben sind optional. Das Symbol kommt als erstes in der Zeile. Dann kann da entweder ein Gleichheitszeichen stehen (für die, die an Hochsprachen gewöhnt sind), oder ein EQU (für die, die Assembler gewöhnt sind), oder weder noch (für die, die keine von beiden Kenntnissen besitzen). Als nächstes kommt der Wert, der dem Symbol zugewiesen werden soll. Schließlich kann dann ein Strichpunkt stehen, gefolgt von einem Kommentar. Leerzeilen und weitere Kommentarzeilen können ebenfalls nach Belieben beigefügt werden.

Sie sollten sich die Datei DEFAULT.Z3P in der Z3PLUS.LBR ansehen, um eine komplette Liste der reservierten Variablenamen zu sehen, die konfiguriert werden können. Durch Editieren einer Kopie(!) dieser Datei (oder einer der anderen Z3P-Files), können Sie das System anpassen, um andere Puffergrößen, andere Laufwerkstabellen,

Bildschirm- und Druckergrößen etc. einzustellen. Die Variablennamen können in beliebiger Reihenfolge stehen und Kommentare im DEFAULT.Z3P-File können die Bedeutung dieser Variablen erläutern.

Numerische Werte können in dezimaler, hexadezimaler, oder binärer Form eingegeben werden. Dezimale Werte werden als Default angenommen. Für Hexadezimal hängen Sie “H” an oder ein “B” für Binär. Einige Variablen erlauben String-Werte; diese Werte müssen als zusammenhängender String eingegeben werden, also ohne Leerzeichen.

Es gibt ein paar Werte, die fast alle Anwender anpassen sollten. Zum Beispiel gibt es da eine Variable namens DRVEC, die dem System mitteilt, welche Laufwerke (A, B, etc.) auf Ihrem Computer existieren. Diese Variable hat ein Bit für jedes der möglichen 16 Laufwerke, die von CP/M unterstützt werden. Die Default-Konfigurationen haben alle 16 Laufwerke aktiv, aber nur sehr wenige Anwender werden tatsächlich so viele (an einem Rechner!) haben. Durch “Abschalten” der Bits für nicht-existente Laufwerke werden der Kommando-Prozessor und einige Z-System-Programme gnädigerweise Zugriffsversuche auf diese nicht-existenten Laufwerke unterbinden.

Der Z3PLUS Kommando-Suchpfad ist das Beispiel einer Stringvariablen. Sein Wert ist eine Sequenz von Buchstaben (wohlgemerkt, ohne Leerzeichen!), die ein bis fünf Drive/User-Angaben repräsentieren. Ein “\$”-Zeichen anstelle der Laufwerksangabe steht für das momentan eingeloggte Laufwerk; in der Position der Usernummer repräsentiert den momentan eingeloggtten Userbereich. Zum Beispiel,

```
PATH = $$B0A15
```

kennzeichnet den Suchpfad:

- Zuerst momentanes Laufwerk und Userbereich
- danach Laufwerk B:, User 0
- schließlich Laufwerk A:, User 15

Die letzte Drive/User-Angabe im Suchpfad (in diesem Beispiel A15:) wird Root Directory genannt.<sup>5</sup> Viele Z-System-Abläufe beziehen sich automatisch auf dieses Directory. Zum Beispiel sucht der Z3PLUS

---

<sup>5</sup>Verwechseln Sie dieses Root Directory nicht mit einem Directory, das zufällig ROOT heißt. Das letztere ist nur ein Name und beinhaltet keinerlei Bedeutung für das Directory.

Kommando-Prozessor im Root Directory nach dem Erweiterten Kommando Prozessor. ARUNZ und ZFILER suchen dort ihre CMD-Files.<sup>6</sup>

Eine andere Stringvariable, die Sie wahrscheinlich verändern möchten, ist der Name des Startup Kommandos. Alle drei Default Systeme benutzen den Namen STARTZ3P. Sie möchten eventuell unterschiedliche Startup-Operationen für verschiedene Sonderanpassungen ausführen. Sollte das der Fall sein, geben Sie den Startup Kommandos unterschiedliche Namen und erzeugen Sie individuelle Aliase.

Es gibt zwei Arten, Sonderanpassungen durchzuführen. Die eine Art besteht darin, komplett neue Definitionen in Z3P-Files mit komplett neuen Namen zu erzeugen. Die andere besteht darin, die Default-Definitionen zu modifizieren.

### Erzeugen neuer Definitionen

Vollständig neue Definitionen bieten die größte Flexibilität: alle Variablen-Definitionen können geändert werden und, welche System-Moduln Sie auch immer möchten, können in das Spezialesystem geladen werden. Die Default Systeme arbeiten mit vorbestimmten System-Moduln, wie wir in wenigen Augenblicken sehen werden.

Der Nachteil totaler Spezialanpassungen ist, daß der Befehl zu dessen Ladung jedes zu ladende Modul explizit angeben muß . Wurde ein Modul nicht angegeben, so wird dafür *kein* passendes Default-Modul substituiert. Vergessen Sie zum Beispiel den Namen eines CPR-Moduls, entsteht eine Fehlersituation, da das System ohne Kommando-Prozessor nicht laufen kann.

Das folgende Beispiel sollte helfen, klarzumachen, was benötigt wird. Nehmen wir an, Sie haben ein Spezialesystem durch SUPER.Z3P definiert, das den Default Kommando-Prozessor (DEF3.ZRL), aber zwei spezielle Moduln namens SUPERFCP.ZRL und SUPERRCP.ZRL als FCP- respektive RCP-Moduln benutzt. Von der Tatsache ausgehend, daß alle Files sich in der Z3PLUS.LBR befinden, lautet die Ladeanweisung wie folgt:

```
Z3PLUS Z3PLUS.LBR SUPER.Z3P DEF3.ZRL <cr>
```

---

<sup>6</sup>Die bei Z3PLUS mitgelieferten Versionen von ARUNZ und ZFILER sind konfiguriert, ihre CMD-Files im Root Directory zu suchen. Diese Programme können auf Wunsch konfiguriert werden, ein anderes Directory zu durchsuchen.

```
SUPERFCP.ZRL SUPERRCP.ZRL DEFAULT.NDR
DEFAULT.Z3T<cr>
```

Das ist eine Menge Holz zu tippen! Es geht viel einfacher, wenn man diese Befehlszeile als Script in das ALIAS.COMD-File einträgt, vielleicht unter dem Namen SUPER. Dann ist alles, was Sie tippen müssen, wenn ein Z3PLUS System aktiv ist, (vorausgesetzt ARUNZ ist der Erweiterte Kommando- Prozessor)

```
SUPER<cr>
```

und das System wird gewechselt. Bei einem "Kalt"-Start aus CP/M können Sie folgenden Trick anwenden. Rufen Sie das System als zweiten Befehl auf durch eine Multiple Befehlszeile des CP/M-Plus wie folgt:

```
Z3PLUS!SUPER<cr>
```

Das wird das Standard Default Z3PLUS System laden, aber sofort das "SUPER"-System durch Chaining nachladen. Sie müssen dadurch zwar zwei Systeme laden, aber Z3PLUS ist so schnell, daß Sie es kaum bemerken werden. Es geht schneller, als die lange Befehlszeile zu tippen und ist vermutlich schneller als ein SUBMIT-Stapeljob (man könnte die lange Befehlszeile in eine Datei namens SUPER.SUB packen).

### Modifikation der Default-Definitionen

Ein anderer Weg, Spezial-Konfigurationen durchzuführen, ist es, die existierenden Default Z3P-Files zu verändern. Dieser Weg hat den Vorteil, daß die Default Systeme durch Eingabe einer simplen Befehlszeile geladen werden können. Z3PLUS erkennt automatisch, welche Moduln für diese Konfigurationen geladen werden müssen, so daß wir uns darum nicht zu kümmern brauchen.

Das impliziert einige Notwendigkeiten zu Veränderungen der Default-Definitionen, die leicht durchgeführt werden können. Es spricht alles dafür, den Drive Vector (DRVEC), maximale Laufwerke und Userbereiche, den Ur-Pfad, den Namen des Startup-Kommandos und Bildschirm-/Drucker-Charakteristika an Ihr System anzupassen.

Die Größe der Moduln in der Default-Konfiguration *kann* geändert werden, aber nur mit Vorsicht und in bestimmten Grenzen. Das

DEFAULT System lädt zum Beispiel immer ein RCP-Modul namens DEFRCP.ZRL. Wenn Sie DEFAULT.Z3P modifizieren, um einen kleineren RCP anzugeben, wird es ein Problem geben. Wenn Z3PLUS versucht, DEFRCP.ZRL zu laden, stellt es fest, daß das Modul nicht in den reservierten Puffer paßt. Es ist intelligent genug, die Operation an diesem Punkt mit einer Fehlermeldung abubrechen. Um dieses Problem zu umgehen, müßten Sie das Default RCP-Modul durch ein neues, kleineres ersetzen (aber immer noch mit dem Namen DEFRCP.ZRL). Bei entsprechendem Geschick können alle solchen Probleme auf diese Art beseitigt werden (wir haben es so gemacht), aber wir glauben nicht, daß es den Aufwand wert ist und empfehlen diese Methode daher nicht. Es gibt einfachere Lösungen.





# Kapitel 6

## Technisches Nachschlagwerk

### 6.1 Definition von Datei-Typen

Das Z3PLUS-Betriebssystem identifiziert die Funktionen gewisser besonderer Dateien durch ihren Dateityp. Diese Typen sind in der Tabelle 6.1 auf Seite 56 aufgeführt. ZRL-Files sind besondere, relokatable Dateien, die an jede beliebige Z3PLUS-System Konfiguration anpaßbar sind. Sie können sowohl das Format von SLR Systems (SLR-Ass.) oder das von Microsoft (M80) besitzen. Die Moduln müssen in Übereinstimmung mit Tabelle 6.1 benannt werden durch Beifügen einer Zeile im Source-Code mithilfe des "NAME"-Pseudo-Opcodes. Es gibt keine Einschränkungen in Bezug auf die Namen, die für die Dateien selbst verwendet werden können. Zum Zeitpunkt des Ladens bestimmt Z3PLUS die Art des Moduls, das in einer Datei kodiert ist, indem es den darin eingebetteten Namen liest.

### 6.2 Bei Z3PLUS mitgelieferte Dateien

Viele Dateien sind im Lieferumfang des Z3PLUS-Systems enthalten. Die meisten davon sind in diesem Abschnitt zusammen mit Erklärungen ihrer Funktionen aufgeführt. Beachten Sie die Datei RELEASE.NOT in Bezug auf Informationen über eventuelle Änderungen.

DATEI-TYP	INHALT
LBR	Library File
Z3P	Z3PLUS System Descriptor
Z3T	Z3 Terminal Capability Descriptor (TCAP)
NDR	Named Directory Register Datei
ZRL	Benannter Common Bereich, Z-system ReLozierbar
	Datei mit dem REL-Modulnamen :
	CP3xxx    Z3PLUS Kommando Prozessor
	RCPxxx    Residentes Kommando-Paket
	FCPxxx    Fluß Kommando Paket
	wobei "x" irgendein Zeichen ist

Table 6.1: Tabelle der Dateitypen, die das Z3PLUS-System verwendet.

### 6.2.1 Z3PLUS System-Dateien

Die Dateien, die das Z3PLUS-Produkt bilden, sind in der Tabelle 6.2 auf Seite 57 aufgelistet. Wir möchten Sie daran erinnern, daß diese Dateien durch Copyright geschützt und auf einen einzigen Anwender lizenziert sind. Es ist ungesetzlich, diese Dateien zu kopieren oder an Dritte weiterzugeben. Details hierzu entnehmen Sie bitte der Copyright-Notiz-Seite am Anfang dieses Handbuchs.

### 6.2.2 Tools und Utilities

Viele Dateien, die nicht Teil des Z3PLUS-Systems sind, werden *als Zugeständnis an den Anwender* mitgeliefert. <sup>1</sup> Einige dieser Dateien sind Public Domain. Andere unterliegen dem Copyright ihrer Autoren oder der ZSIG, der Z-System Interest Group. Für alle gilt jedoch die Erlaubnis der Autoren, diese Dateien zu Kopieren und ohne Berechnung für *nichtkommerzielle Zwecke* an andere Anwender weiterzugeben. Folgende Dateien unterstützen das TCAP des Z-Systems:

TCSELECT	.COM	Terminal Deskriptor auswählen
Z3TCAP	.TCP	Database der Terminal-Deskriptoren

Die folgenden Dateien unterstützen Z-System-Aliase:

<sup>1</sup>Deshalb können wir für diese Programme keinen extra Support anbieten.

Z3PLUS	.COM	Z3PLUS Systemlader
Z3PLUS	.LBR	Library der Z3PLUS System Moduln
DEFAULT	.Z3P	Default System Deskriptor
DEFAULT	.Z3T	Default Terminal Deskriptor (adm3a)
DEFAULT	.NDR	Default Benannte Directory Registerdatei
DEFPCP3	.ZRL	Default Kommando-Prozessor
DEFRCP	.ZRL	Default Residentes Befehls-Paket
DEFPCP	.ZRL	Default Fluß Kommando-Paket
SMALL	.Z3P	SMALL System Deskriptor
LARGE	.Z3P	LARGE System Deskriptor
LARGERCP	.ZRL	RCP für das Large System
TIMECP3	.ZRL	CPR mit Stunde:Minute-Prompt
JETLDR	.COM	Z-System Paket-Lader

Table 6.2: Liste der Z3PLUS-Systemdateien.

SALIAS	.COM	Generiert alleinstehende Aliase
ARUNZ	.COM	Alias Befehls-Prozessor
ALIAS	.CMD	Alias Script-Datei-Beispiel für ARUNZ

Die folgende Datei wird vom FCP verwendet, um erweiterte konditionelle Prüfungen auszuführen:

IF	.COM	Erweiterter Fluß-Bedingungsprüfer
----	------	-----------------------------------

Die folgenden Utilities definieren oder zeigen diverse Z-System Environment Variablen und System Möglichkeiten:

CPSET	.COM	definiert und zeigt CRT/PRT Charakteristika
PATH	.COM	setzt und zeigt den Befehls-Suchpfad <sup>2</sup>
ZSHOW	.COM	zeigt Konfigurations-Information an <sup>3</sup>
EDITNDR	.COM	Editiert NDR im Speicher <sup>4</sup>
SAVENDR	.COM	Speichert Named Directory Register in Datei

<sup>2</sup>Der Name hat sich von ursprünglich SETPATH.COM geändert

<sup>3</sup>Der Name hat sich von ursprünglich SHOW.COM geändert

<sup>4</sup>Der Name wurde von ursprünglich EDITND.COM geändert

Folgende Programme sind Utilities von allgemeinem Interesse:

FF	.COM	File Finder
CRUNCH	.COM	Datei-Kompressor Tool
UNCRUNCH	.COM	Datei-Dekompressor Tool

Folgende Programme unterstützen Library Files:

LDIR	.COM	Library DIrectory Programm
LGET	.COM	Library Extraktor
LPUT	.COM	Library Datei-Insertor
VLU	.COM	Video-orientiertes Library Utility
LX	.COM	Library-Programm Exekutor

Die folgenden Dateien gehören zur Gruppe der Shells und Error Handler:

ZF-REV	.COM	ZFILER Shell für Reverse-Video Terminals
ZF-DIM	.COM	ZFILER Shell für Dim-Video Terminals
ZFILER	.CMD	Macro Script File für ZFILER
LSH	.COM	Befehls-History-LOG-Shell
ZERRLSH	.COM	Error Handler für LSH

Folgende Dateien unterstützen die Z-System Help-Einrichtung:

ZHELP	.COM	Zeigt Help-Dateien menügesteuert an <sup>5</sup>
xxx	.HLP	diverse Help-Menü-Dateien

Die folgenden Programme sind Transiente "Type-4"-Programme (siehe S. 22), die am obersten Ende der verfügbaren TPA ablaufen und den unteren Teil nicht berühren. Diese sind besonders nützlich in einem "Minimum"-System, wo transiente Programme für Funktionen verwendet werden, die normalerweise durch RCP-Befehle ausgeführt werden. Zur Verwendung benennen Sie diese einfach um, indem Sie das vorangestellte TY4 weglassen.

TY4SP	.COM	Freien Diskettenspeicher anzeigen
TY4SAVE	.COM	Speicherbereich in Datei schreiben
TY4REN	.COM	Rename — Umbenennen
TY4ERA	.COM	Erase — Löschen

---

<sup>5</sup>Der Name hat sich vom ursprünglichen HELP.COM geändert

Es gibt verschiedene Wege, an die geballte Ladung der stets expandierenden Z-System User-Group Dateien zu gelangen. Die Telefonnummern der Z-Node Mailboxen sind aufgeführt in:

```
ZNODES .LST
```

Eine Gruppe von Personen hat sich bereitgefunden, anderen bei der Installation und Anwendung von Z-Systemen zu helfen. Ihre Namen, Adressen und Telefonnummern sind in der Datei

```
ZHELPERS .LST
```

aufgeführt.

## 6.3 Z3PLUS Befehls-Zeilen

### 6.3.1 Help Seiten

Eine eingebaute Help-Seite für die Syntax der Z3PLUS Befehle wird angezeigt durch Eingabe von:

```
Z3PLUS //<cr>
Z3PLUS ?<cr>
```

Diese Hilfs-Seiten geben auch Auskunft über die Konfiguration der Quiet-Option und des System-Directorys (siehe unten).

### 6.3.2 Laden des Default-Systems

Es gibt drei Default-Systeme im Lieferumfang von Z3PLUS. Sie werden durch folgende Befehls-Zeile geladen :

```
Z3PLUS<cr>
Z3PLUS ON<cr>
Z3PLUS LARGE<cr>
Z3PLUS SMALL<cr>
```

Die ersten zwei Formen laden das "Standard" Default system. Die anderen Formen laden das LARGE- und das SMALL- Default-System.

Options-Schalter, wie "/Q" oder "/V" können ans Ende einer Befehlszeile gepackt werden, um "Quiet" oder "Verbose" Modus zu

wählen. Im Verbose-Modus zeigt das Z3PLUS eine ganze Seite Information über das ladende System an. Im Quiet-Modus wird diese Ausgabe unterdrückt. Die eingebaute Hilfe-Seite von Z3PLUS.COM zeigt an, welcher dieser zwei Modi benutzt wird, wenn keine Option in der Befehls-Zeile angegeben ist.

Die System Moduln werden aus der Library-Datei Z3PLUS.LBR geladen, die in einem speziellen Verzeichnis, dem "System Directory" stehen muß. In der Lieferversion des Z3PLUS.COM ist dies das Directory A0:. Siehe Abschnitt ?? (S. ??) wegen Information, wie man das ändern kann. Das momentan konfigurierte System Directory wird von der Z3PLUS Help-Seite angezeigt.

Das Standard Default-System braucht 6.88K Speicher und lädt die folgenden Moduln aus Z3PLUS.LBR:

DEFAULT	.Z3P	System-Deskriptor
DEFPCP3	.ZRL	Kommando-Prozessor
DEFRCP	.ZRL	Residentes Befehls-Paket
DEFFCP	.ZRL	Fluß -Befehls-Paket
DEFAULT	.Z3T	Terminal Deskriptor
DEFAULT	.NDR	Benannte Directory Zuweisungen

Das LARGE System braucht 7.38K Speicher. Die gleichen Moduln wie beim Standard System werden geladen, außer:

- LARGE.Z3P statt DEFAULT.Z3P
- LARGERCP.ZRL statt DEFRCP.ZRL.

Das SMALL System hat keine Puffer für RCP, FCP, oder NDR und benötigt nur 4K. Das Laden der Moduln weicht auf folgende Weise vom Standard ab:

- SMALL.Z3P statt DEFAULT.Z3P
- DEFRCP.ZRL, DEFFCP.ZRL, und DEFAULT.NDR werden nicht geladen.

### 6.3.3 Das Entfernen von Z3PLUS

Das Z3PLUS Betriebssystem kann aus dem System entfernt und das CP/M-Plus Betriebssystem wiederhergestellt werden durch Eingabe des Befehls:

```
Z3PLUS OFF<cr>
```

### 6.3.4 Das Laden spezifischer System-Moduln

Z3PLUS kann nicht nur zum Laden ganzer Betriebssysteme, sondern auch zum Laden spezifischer System-Moduln verwendet werden. Diese Modul können entweder individuelle Dateien oder Teile von Library-Dateien sein. Mehrere Moduln können gleichzeitig geladen werden. Die folgenden Befehlsformen werden verwendet:

```
Z3PLUS Dateiliste [/Option]<cr>
Z3PLUS Lbrfile LBR-Dateiliste [/Option]<cr>
```

Dabei bedeutet "Dateiliste" eine Liste mit einem oder mehreren Dateinamen, alle in der Form

```
[dir:]name.typ
```

Der Ausdruck "Lbrfile" hat die Form

```
[DIR:]lbrname[.LBR]
```

"LBR-Dateiliste" schließlich, ist die Liste einer oder mehrerer Dateien, die in der bezeichneten Library enthalten sind. Das Format für einen solchen "Member"-Dateinamen ist gleich mit den Namen in "Dateiliste", außer daß naturgemäß keine Directoryangabe dabeistehen kann.

Zwei beliebige Dateinamen können in einer Befehlszeile entweder durch Leerzeichen, Kommata oder beides getrennt werden. Die Options-Schalter /Q und /V können bei diesen Befehlen ebenfalls eingesetzt werden.

Directories können für jede einzelne Datei in der ersten Form und für die Library-Datei in der zweiten Form angegeben werden. Wann immer es keine expliziten Directory-Angaben für eine Datei gibt, wird das System Directory (siehe Abschnitt 6.3.2) verwendet. Wenn Directories angegeben sind, können diese immer die Form DU:, (also Laufwerksbuchstabe Usernummer) haben. Ist ein Z3PLUS-System aktiv, kann auch die Form DIR: (Benanntes Directory) verwendet werden.

Der Dateityp LBR ist optional in der zweiten Form. Wenn bei der ersten Datei keinen Dateityp angegeben war, wird der Typ LBR

angenommen. Für die zweite Befehlsform, müssen alle Moduln, die nach einem Library-Namen angegeben werden, in dieser Library sein. Man kann nicht zugleich individuelle Dateien und Library-Dateien in einem einzigen Befehl laden.

## 6.4 Das Programm JetLDR

Im Lieferumfang des Z3PLUS-Pakets ist ein spezielles Programm namens JetLDR enthalten. Dieses Programm ist ein extrem mächtiger Allzweck-Modul-Lader. Alle der üblichen Modul-Ladevorgänge, die für das Z3PLUS-System notwendig sind, können jedoch mit Hilfe von Z3PLUS.COM ausgeführt werden, das diese Fähigkeit durch "Ausleihen" von JetLDR-Code besitzt.

Es gibt einige wichtige Unterschiede zwischen JetLDR.COM und Z3PLUS.COM. JetLDR kann nur unter dem Z-System gestartet werden und kann nicht verwendet werden, um ein neues System aufzubauen, so wie Z3PLUS das kann. Es wird nur zum Laden von Moduln verwendet. Es können jedoch viel mehr Modul-Typen geladen werden als Z3PLUS das kann und es führt ausgefeilte Gültigkeitsprüfungen für Moduln durch und versucht sicherzustellen, daß ein Modul für den Zielpuffer geeignet ist.

Das Programm JetLDR macht ausgiebig Gebrauch von speziellen Konfigurations- oder CFG-Dateien. Diese Moduln werden von JetLDR selbst geladen und bestimmen die Art und Weise, wie andere Moduln geladen werden. Diese Einrichtung kann JetLDR die Fähigkeit verleihen, spezielle Moduln wie Resident System Extensions (RSXe), BIOSe, und DOSe zu laden.

JetLDR hat eine eingebaute Help-Seite, der mit der Standard-Syntax

```
JETLDR //<cr>
```

aufgerufen werden kann. Sie werden sehen, daß seine Befehls-Syntax nahezu identisch zu der von Z3PLUS ist.



## 6.5 Patchen von Z3PLUS.COM

Es gibt diverse Optionen in Z3PLUS.COM, die von Konfigurations-Bytes in der ersten Code-Page <sup>6</sup> bestimmt werden. Diese Bytefolgen sind alle durch Text-Strings gekennzeichnet, so daß sie mit einem Debugger (etwa SID, der beim CP/M-Plus enthalten ist) oder einem Patching-Utility, am besten dem Z-System-Tool ZPATCH leicht zu finden sind.

Das erste Bytepaar ist durch den String "SYSUD" gekennzeichnet. Die zwei darauffolgenden Bytes definieren den Userbereich (00H bis 0FH) und Laufwerk (A=00H bis P=1FH), die als System Directory (siehe S. 60) benutzt werden sollen.

Das nächste Byte ist mit dem String "QUIET" bezeichnet. Es entscheidet, ob der /Q oder der /V Modus als Default für die Bildschirmausgabe gilt. Ist das Byte 00H, ist der Quiet-Modus ausgeschaltet (der Verbose-Modus ist selektiert). Wenn das Byte FFH ist, ist der Quiet-Modus Default.

Das Byte hinter dem String "TEST" ist für Entwicklungs- und Debug-Zwecke reserviert. Anwender müssen normalerweise an diesem Byte nichts ändern, aber natürlich können Sie gerne damit experimentieren, wenn Ihnen Ihre Neugierde keine Ruhe läßt.

## 6.6 Unterschiede zwischen Z3PLUS und CP/M-Plus

Es gibt offensichtlich viele Unterschiede zwischen Z3PLUS und CP/M-Plus. In diesem Abschnitt wollen wir einige herausstellen, die Verwirrung oder Schwierigkeiten bereiten könnten, wenn Sie nicht darauf hingewiesen würden.

### 6.6.1 Das GET Kommando

Unter Z3PLUS, ist GET ein (eingebauter) CPR-Befehl, der eine Datei an eine bestimmte Speicheradresse lädt. Um das CP/M-

---

<sup>6</sup> 128 Bytes werden im Maschinencode als eine Code-Page bezeichnet, hier also zwischen Hex 100 und Hex 17F

Plus GET.COM Programm zu benützen, stellen Sie dem Befehl irgendwelche Laufwerksangaben voran wie in den folgenden Beispielen:

```
:GET ...<cr>  
A:GET ...<cr>  
BASE:GET ...<cr>
```

Alternativ dazu können Sie es umbenennen etwa GETCPM.COM.

### 6.6.2 IOP — Input/Output Package

In der Z3PLUS-Version des Z-Systems gibt es keinen Support für das Input/Output Package (IOP) des ZCPR3. Statt dessen, wird das schon im CP/M-Plus vorhandene benutzt. Terminal- und Druckersteuerung, sowie Umleitung aus und in eine Datei oder ein anderes Device werden unterstützt durch die Utilities GET.COM und PUT.COM<sup>7</sup>, als auch durch RSXe. Die Device-Auswahl wird ebenfalls durch CP/M-Plus-Utilities unterstützt<sup>8</sup>. Sehen Sie dazu bitte auch in Ihren CP/M-Plus-Handbüchern nach, um weitere Information zu erhalten.

### 6.6.3 Public Files

CP/M-Plus unterstützt ein “Public”-Dateisystem, wobei ausführbare Dateien im User 0 mit gesetztem “System”- oder \$SYS-Attribut aus allen Userbereichen dieses Laufwerks aufgerufen werden können. Dieses Feature wird von Z3PLUS nicht erkannt, da hier stattdessen der Befehls-Suchpfad verwendet wird. Im Gegensatz zu dem mit SETDEF in CP/M-Plus definierten Suchpfad, sind im Z-System-Suchpfad nicht nur die Laufwerke, sondern auch vollständige Directories angegeben, einschließlich Userbereichen.

### 6.6.4 Paßwörter

Das Z-System erlaubt Directory-Paßwörter und Laufwerk/User-Begrenzungen um alle Dateien in angegeben benannten Directories

---

<sup>7</sup>diese finden Sie auf Ihrer CP/M-Plus System Diskette, eventuell auch auf einer UTILITY Seite, wenn Sie mehrere Systemdisketten besitzen

<sup>8</sup>DEVICE.COM oder ä.

und Laufwerk/User-Bereichen zu schützen. Der Z3PLUS Kommando-Prozessor verbietet nicht den Zugriff auf Dateien, die von CP/M-Plus-Datei-Paßwörtern geschützt werden. Um sensible Dateien vor der Ausführung oder dem Betrachten zu schützen, verstecken Sie sie in ein paßwortgeschütztes Directory oberhalb der von Ihnen erlaubten Laufwerks- und User-Maxima. Bücher und Druckwerke, die in der Bibliographie aufgeführt sind, bieten Information, wie man die Z-System Sicherheits-Features einsetzt.

Alle CP/M-Plus-Dateipaßwörter bleiben aktiv, wenn ein Programm versucht, die betreffende Datei zu öffnen.

### 6.6.5 Der Kommando-Suchpfad

Der Z3PLUS Kommando-Prozessor sucht nach Befehlsdateien entlang des Z-System-Pfads. Er wird in der Z3P-Deskriptor-Datei initiiert und kann jederzeit mit Hilfe von PATH.COM geändert werden. Z3PLUS deaktiviert den CP/M-Plus Suchpfad.

Der letzte Eintrag in Ihrem Pfad wird als Root Directory bezeichnet; es sollte dasjenige sein, wo sich Ihr CMDRUN.COM, der Erweiterte Kommando-Prozessor befindet.

### 6.6.6 Mehrfach-Kommandos

Unter Z3PLUS werden Befehle innerhalb einer Zeile durch einen Semikolon (;) getrennt. Der Z3PLUS Kommando-Prozessor unterstützt keine Dateipaßwörter, so daß der Semikolon, der unter CP/M-Plus einem Paßwort vorangestellt wird, dafür nicht benötigt wird. Der CP/M-Plus Befehls-Separator, das Ausrufezeichen (!), hat keine besondere Bedeutung unter Z3PLUS, außer wenn Z3PLUS das erste Mal geladen wird (siehe S. 44).

### 6.6.7 Konditionelle Ausführung

Z3PLUS stellt sehr ausführliche Möglichkeiten zum Testen und konditionellen Ausführen eines Befehls, basierend auf Flow States und den Befehlen im Flow Command Package. Unter dem Z-System veranlaßt ein vorangestellter Doppelpunkt (:) in einer Zeile einer Submitdatei nicht, daß die Zeile übersprungen wird, wenn ein CP/M-Plus-

Programm Return Code einen Fehler meldet. Statt dessen wird er interpretiert als Anweisung, das nachfolgende Wort als ausführbare Diskettendatei zu suchen. Bevor Sie daher Ihre herkömmlichen Submit-Dateien unter Z3PLUS verwenden, müssen Sie vorangestellte Doppelpunkte daraus entfernen.

## 6.7 Funktionale Theorie

Z3PLUS ist eine spezielle CP/M-Plus Resident System Extension (RSX),

die sich an die höchste Speicheradresse direkt unterhalb des CP/M-Plus Common Bereichs des BDOS "einklinkt". Sie enthält diese Hauptsegmente:

- Warm-Boot Umleitung
- Programm- und RSX-Lader
- Kommando-Prozessor
- Z-System-immanente Segmente:
  - Externe Environment Deskriptoren
  - Terminal Capabilities Puffer
  - Message Puffer
  - Pfad
  - Das Wheel Byte
  - Externer File Control Block
  - Mehrfach-Kommandozeile
  - Shell-Stack
  - Externer Kommando-Prozessor-Stack
- optionale Z-System Segment Puffer
  - Benannte-Directory Register
  - Resident Command Package
  - Flow Command Package

Z3PLUS.COM liest Z-System-Dateien in den Arbeitsspeicher aus einer Library (z.B. Z3PLUS.LBR), oder als individuelle Dateien, erstellt die angegebenen System-Adressen und Parameter und reloziert dann dieses System an die Speicher-Obergrenze. Z3PLUS schützt sich selbst durch Abfangen der normalen BIOS Warm-Boot Funktion, indem es die Kontrolle an den residenten

Z3PLUS Kommando-Prozessor übergibt, um auf den nächsten Befehl zu reagieren.

Ist das Z-System schon aktiv, rettet Z3PLUS den Status des gegenwärtigen Z-System-Environments, determiniert die notwendigen Änderungen (z.B. einen

Puffer mit anderer Größe), reloziert und installiert neue Pakete und restauriert

die ungeänderten Komponenten des Environments (wie die Message-Puffer Werte und im Befehlszeilen-Puffer verbliebene Befehle).

Wenn das System entfernt wird, deaktiviert Z3PLUS die Warmboot-Umleitung, entfernt die Z3PLUS RSX, und veranlaßt das CP/M-Plus Host-System, seinen Standard Kommando-Prozessor nachzuladen.

Sowohl der Kommando-Prozessor, als auch der Z3PLUS Lader setzen zahlreiche

Parameter im CP/M-Plus System Control Block. Darüberhinaus, mit Ausnahme des Patches für die BIOS Warmboot Sprungleiste, verändert Z3PLUS das CP/M-Plus Host-System intern nicht.



# Kapitel 7

## Bibliographie

In einer sich entwickelnden und verbessernden Umgebung eilt die Dokumentation immer hinter der Praxis her. Hierzu empfehlen wir einige Quellen zusätzlicher Information über das Z-System.

### 7.1 Z-Nodes

Die aktuellste Information über Z-System-Tools, Standards und die neusten Versionen der frei verteilten Programme können auf den Z-System Remote Access Systems (RASs), genannt Z-Nodes, gefunden werden. Diese "Bulletin Boards" oder "Mailboxen" sind auch gute Orte, um freundliche Unterstützung durch Anwenderkollegen zu erhalten. Die gecrunchte Datei ZNODESxx.LZT enthält eine Liste der Z-Nodes, die derzeit aktiv sind.

Von all diesen vielen Z-Nodes wollen wir Ihre spezielle Aufmerksamkeit auf vier lenken (drei davon sind über Telenet's PC-Pursuit Service erreichbar). Z-Node #51, erreichbar als Outdial-Server über die NBBS-Mailbox der Firma Nord-Computer. Man muß registrierter Benutzer dieser Box sein, um die volle Leitung in Anspruch nehmen zu können. Zugang besteht von 9:00 Uhr am Vormittag bis 2:00 Uhr nachts mit 300/1200 und 2400 Baud. Dort stehen immer die neuesten Dateien zum Ansehen bereit. Einiges kann dort direkt ausgeführt werden.

Z-Node #2 in Los Angeles ZNODE CENTRAL ist der nächste zum Z3PLUS-Author Bridger Mitchell und derjenige, wo dieser am schnell-

sten und direktesten erreicht werden kann. Die Rufnummer dort ist (213)-670-9465.

Jay Sage, einer der "Architekten" des Z3PLUS und Author des ZCPR34 (worauf der Kommando-Prozessor von Z3PLUS basiert), ist der Sysop des Z-Node #3 im Bostoner Gebiet. Die Rufnummer ist (617)-965-7259. Anders als andere Remote Access Systeme, ist es ein offenes System, ohne individuelle User Registration oder Paßwörter und nur mit öffentlichen Messages zwischen Anwendern. Es gibt jedoch ein allgemeines System-Paßwort, das dazu dient, nur den Anwendern Z-System-kompatibler Computer Zugang zu gewähren. Dieses Paßwort ist zur Zeit "DDT", der Name des Debugging Utilitys, das mit CP/M-2.2 geliefert wird. Jetzt, wo Z3PLUS CP/M-Plus Computer zumindest Z-System-kompatibel gemacht hat, ist vielleicht das alternative Paßwort "SID" bald gestattet!

Zum Schluß wollen wir den Z-Node #32 erwähnen, auch bekannt als Socrates Z-NODE. Chris McEwen in S. Plainfield sorgt für stets aktuelle Dateien durch den sogenannten ZSUS-Vertrag, Z-System Upgrade Service, der für 100 US\$ abonniert werden kann. Die Rufnummer dieser Box ist 201-754-9067.

## 7.2 *The Computer Journal*

Ein exzellente, weiterführende Quelle für Z-System-Material ist *The Computer Journal* (TCJ), vielleicht das letzte der großen Hobby Computer Magazine mit signifikanter Themenbehandlung von 8-bit Systemen. Darin gibt es regelmäßige Serien und besondere Artikel von Z-System-Experten, einschließlich sowohl Jay Sage und Bridger Mitchell. Ein Abonnement wird sehr empfohlen! Adresse: P.O. Box 1697, Kalispell, MT 59903.

Im Z3PLUS-Paket sind Diskettenfiles mit einer Reihe von Jay Sage's Serie aus TCJ. Die Dateien sind alle gecruncht um Platz zu sparen; siehe dazu auch Seite 38 mit einer Diskussion über den Umgang mit gecrunchten Dateien. Hier eine Liste dieser Dateien mit einer Kurzanzeige des Inhalts:

- TCJ26.MZG Optimierung reiner Floppy-Disk-Systeme
- TCJ27.MZG Aliase und Shells
- TCJ28.MZG Rekursive Aliase



- TCJ29.MZG Der ZCPR33 Kommando-Prozessor
- TCJ30.MZG SALIAS und VLU
- TCJ31.MZG ARUNZ Dokumentation
- TCJ32.MZG NZ-COM/Z3PLUS/ZCPR34 Information

## 7.3 Andere Publikationen

In Bezug auf andere gedruckte Information über das Z-System, empfehlen wir folgende Bücher.

Lesen Sie zuerst:

*The Z-System User's Guide* (Bruce Morgen, Richard Jacobson). Eine Einführung ins Z-System, die versucht, dem weniger technisch begabten Anwender von Z-Systemen verständlich zu sein.

Dann lesen Sie:

*The ZCPR 3.3 User's Guide* (Jay Sage). Dieses Manual begleitet den ZCPR 3.3 Kommando-Prozessor. Es beinhaltet viele Beispiele, wie die Features des Z-Systems zu Ihrem Vorteil genutzt werden können. Erweiterte Befehlsverarbeitung und Security Features im Besonderen werden behandelt. Fast die gesamte Information gilt auch für ZCPR in der Version 3.4 und Z3PLUS.

Ein älterer Querverweis, mit Information, die nicht mehr so ganz auf dem Laufenden ist:

*ZCPR3: The Manual* (Richard Conn). Das war die "Bibel" für ZCPR3, aber vieles davon ist inzwischen überholt. Die Behandlung der Z-System HELP-Einrichtung, die Menu-Shells und TCAPs (Terminal CAPability Deskriptoren, inklusive der Utilities TCSELECT, TCMAKE, und TC-CHECK) sind immer noch sehr nützlich.

Für die technisch Versierten, die sich ihre eigenen Z-System Programme schreiben wollen, ist das folgende Buch, zusammen mit dem relozierbaren Subroutinen-Libraries-Code extrem nützlich.

*ZCPR3: The Libraries* (Richard Conn). Dieses Buch liefert die vollständige Dokumentation der hunderten von vorgefertigten (und debuggten) Subroutinen, die das Schreiben von Z-System-Programmen in Assembler fast so einfach machen, wie das Programmieren in einer Hochsprache.