

# NZ-COM

## Das Automatische Z-System

von

Joe Wright

# User's Manual

von

Jay Sage und Bridger Mitchell

Manual © Copyright 1988  
Jay Sage und Bridger Mitchell

Alpha Systems  
711 Chatsworth Place  
San Jose, CA 95128

## Copyright Vermerk

NZ-COM — Das Automatische, Dynamische Z-System für CP/M-2.2 Computer — ist © Copyright 1988 Joe Wright. Alle Rechte vorbehalten. Das NZ-COM User's Manual ist © Copyright 1988 - Jay Sage und Bridger Mitchell. Deutsche Übersetzung ist © Copyright 1991 - Helmut Jungkunz. Der ZCPR34 Commando Prozessor ist © Copyright 1988 - Jay Sage, Alle Rechte vorbehalten. JETLDR.COM ist © Copyright 1988 Bridger Mitchell, Alle Rechte vorbehalten.

Ihre Einzel-Lizenz gestattet Ihnen, NZ-COM auf Ihrem/(n) Computer(n) einzusetzen, Sicherheitskopien für eigene Verwendung anzufertigen und NZ-COM anderen vorzuführen. Von geschützten NZ-COM-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 NZ-COM Dateien sind: NZCOM.COM, NZCOM.LBR, und MKZCOM.COM. Durch Copyright geschützte ZCPR34 Dateien sind ZCPR34.LBR und NZCPR.ZRL.

Z-System Tools und andere Dateien, sind auf der NZ-COM Diskette als Zugeständnis an die NZ-COM-Lizenz. Viele dieser Dateien genießen Copyright durch deren einzelne 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 irgendeinem 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 NZ-COM 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 NZ-COM System wurde getestet und verbessert durch Anregungen von Cam Cotrill, Howard Goldstein, Bridger Mitchell, und Jay Sage.

Das Z-System ist eine sich entwickelnde Betriebs-System-Umgebung für Z80-kompatible Computer, zu der viele beigetragen haben. Die Entwicklung von NZ-COM 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
Bridger Mitchell	-	Z3PLUS, JETLDR
Bruce Morgen	-	LPUT, LGET
Paul Pomereau	-	EASE
Jay Sage	-	ZCPR v 3.3, v 3.4, ARUNZ, ZFILER, ZSHOW
Joe Wright	-	NZCOM

## Warenzeichen

Z-System, ZCPR3	Alpha Systems
Z3PLUS, DosDisk, DateStamper	Bridger Mitchell
CP/M, CP/M-Plus	Digital Research, Inc.
WordStar	MicroPro Corp.

## VORWORT

Automatisch, universell, dynamisch, die zwei neuen 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 automatisch selbst 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 Commando-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 Datei-Format und einen Lader, der es einem einzelnen File erlaubte, sich an jedes beliebige Z-System anzupassen, ungeachtet dessen Konfiguration in Bezug auf Modul-Adresse 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 Commando-Prozessors (Jay Sage), den Z3PLUS-Lader und Commando-Prozessor (Bridger Mitchell), sowie den NZ-

COM-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 NZ-COM?</b>	<b>1</b>
1.1	Die Vorteile von NZ-COM . . . . .	3
1.2	Die Eigenschaften des neuen Kommando-Prozessors . . .	3
1.3	Andere Haupt-Eigenschaften . . . . .	4
1.4	Zum Gebrauch des Handbuchs . . . . .	4
<b>2</b>	<b>Erstes Arbeiten mit NZ-COM</b>	<b>7</b>
2.1	Einleitung . . . . .	7
2.2	Das Zusammenstellen der Dateien . . . . .	8
2.3	Der Betrieb von NZ-COM . . . . .	10
2.4	Die Organisation Ihrer Dateien . . . . .	11
<b>3</b>	<b>Kennenlernen und Verstehen des Z-Systems</b>	<b>13</b>
3.1	Wie man einen Alias erzeugt . . . . .	13
3.2	Die verfügbaren Befehle . . . . .	15
3.2.1	RCP Kommandos . . . . .	16
3.2.2	FCP Kommandos . . . . .	17
3.2.3	CPR Kommandos . . . . .	21
3.2.4	Transiente Kommandos . . . . .	23
3.3	Mehr über die System-Konfiguration . . . . .	24

<b>4</b>	<b>Mehr vom Z-System</b>	<b>27</b>
4.1	Residente NZ-COM Komponenten . . . . .	27
4.2	Automatische Kommandos . . . . .	28
4.2.1	Der Erweiterte Kommando-Prozessor . . . . .	28
4.2.2	Error Handler . . . . .	30
4.2.3	Kommando Shells . . . . .	30
4.3	Andere Z-System Tools . . . . .	35
4.3.1	ARUNZ . . . . .	36
4.3.2	HELP . . . . .	38
4.3.3	Library Tools . . . . .	39
4.3.4	Datei-Kompression . . . . .	39
4.3.5	Benannte Directory Tools . . . . .	40
4.3.6	Andere Tools . . . . .	42
4.4	Kommando-Hierarchie . . . . .	42
4.4.1	Kommando Akquisition . . . . .	42
4.4.2	Kommando Auflösung . . . . .	43
<b>5</b>	<b>Wie man mehr aus NZ-COM herausholt</b>	<b>45</b>
5.1	Alternative Befehle zum Aufruf von NZ-COM . . . . .	45
5.1.1	Alternative Lademethoden . . . . .	45
5.2	Anpassen Ihres NZ-COM Systems . . . . .	46
5.2.1	Vorübergehende Änderungen . . . . .	47
5.2.2	NZ-COM Deskriptor-Dateien . . . . .	47
5.2.3	Modifizieren der Deskriptor-Dateien . . . . .	48
5.2.4	Erzeugen neuer Definitionen mit MKZCM . . . . .	51
5.3	Laden neuer Systeme . . . . .	54
5.4	Clonen des Systems . . . . .	56
5.5	Systemmodifikationen “im Flug” . . . . .	57

5.5.1	Lade-Regeln . . . . .	57
5.5.2	Warum Systeme wechseln? . . . . .	58
5.5.3	Automatisierung der Systemwechsel . . . . .	59
5.5.4	Automatische Rückkehr zu CP/M . . . . .	60
<b>6</b>	<b>Technisches Nachschlagwerk</b>	<b>63</b>
6.1	Definition der Datei-Typen . . . . .	63
6.2	Bei NZ-COM mitgelieferte Dateien . . . . .	63
6.2.1	NZ-COM System-Dateien . . . . .	64
6.2.2	Tools und Utilities . . . . .	65
6.3	NZCOM Befehlszeilen . . . . .	68
6.3.1	Hilfe-Seiten . . . . .	68
6.3.2	Laden des NZ-COM Systems . . . . .	68
6.3.3	Das Entfernen von NZ-COM . . . . .	73
6.3.4	Beispiele und Tips . . . . .	73
6.4	Patchen von NZCOM.COM . . . . .	75
6.4.1	Der Interne Suchpfad . . . . .	76
6.4.2	Die Default-Optionen . . . . .	77
6.4.3	Die Startup Befehlszeile . . . . .	77
6.5	Das Programm JetLDR . . . . .	81
6.6	Die MKZCM-Befehlszeile . . . . .	81
6.7	Funktions-Theorie . . . . .	82
<b>7</b>	<b>Bibliographie</b>	<b>85</b>
7.1	Z-Nodes . . . . .	85
7.2	<i>The Computer Journal</i> . . . . .	86
7.3	Andere Publikationen . . . . .	87

# Kapitel 1

## Was ist NZ-COM?

Dieses Handbuch muß sich an zwei Gruppen wenden: an jene, die schon eine Version besitzen von etwas, das als das ZCPR3 Betriebssystem oder Z-System bekannt ist, und an jene, für die dies alles neu ist. Da man aus unserer Sicht immer etwas über das Z-System lernen kann, haben wir keine Bedenken, das Manual an die zweite Gruppe zu adressieren. Wir haben nach besten Kräften versucht, beim Leser keine Kenntnisse des Z-Systems vorauszusetzen. Erfahrene Z-System-Anwender werden einige Passagen gerne überspringen oder überfliegen. Andererseits können sich in jedem Abschnitt kleine Leckerbissen oder neue Einblicke verbergen.

NZ-COM — früher ZCOM Version 2 — ist eine der aufregendsten und bemerkenswertesten Entwicklungen in der Geschichte der Microcomputer-Betriebssysteme. Das Betriebssystem Ihres Computer 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.

Normalerweise ist das Betriebssystem eines Computers eine Statische Ganzheit. Man "bootet" den Rechner und da ist das Betriebssystem, fixiert und unveränderlich. Wenige Computer bieten mehr als ein Betriebssystem. Bei diesen ist der einzige Weg, ein anderes Betriebssystem zu laden, neues "Booten", womit im allgemeinen das Einlegen einer neuen Bootdiskette und Drücken des Reset- Knopfes verbunden ist. Dabei haben *Sie* niemals die Möglichkeit, die Charakteristika dieses Betriebssystems zu bestimmen. Sie müssen akzeptieren was der Her-



steller anbietet.

Mit NZ-COM wird das Betriebssystem ein flexibles Werkzeug, gerade wie ein Anwendungsprogramm. Sie können schnell die Konfiguration des Betriebssystems jederzeit ändern, sogar mitten in einer automatischen Befehls-Sequenz. Das kann manuell geschehen, oder Eigenschaften des Betriebssystems können angewandt werden, um dies auf Wunsch automatisch zu tun.

Man kann das gesamte Betriebssystem oder Teile davon ändern. Sie möchten einen neuen Kommando Prozessor? Kein Problem. Mit einem einzigen Befehl wird NZCOM diesen laden — ohne Assemblierung oder Konfigurationsänderung. Der neue Kommando Prozessor bleibt im Speicher bis ein neuer geladen oder neu gebootet wird. Sie möchten andere eingebaute (residente) Befehle? Wiederum - kein Problem: NZ-COM kann diese in Sekundenschnelle laden. NZ-COM bietet Ihnen eine ganze Welt an Flexibilität und Anpaßbarkeit. Es macht es leicht für Sie zu experimentieren und zu lernen.

Das Basis-Betriebssystem, das NZ-COM zur Verfügung stellt, ist das beliebte ZCPR oder Z-System, dessen Eigenschaften weiter unten in diesem Manual noch etwas ausführlicher beschrieben werden. Bis man NZ-COM getestet hat, ist es fast unmöglich, sich vorzustellen, wie leicht es ist, das Z-System Ihrer Wahl zu designen und lauffähig zu bekommen. Vorbei die Tage des Quellcodes (es wird kein Quellcode benötigt), des Editierens von Konfigurations-Dateien (Kein Editor nötig), des Assemblierens (kein Assembler nötig), und des "Patchens" (keine Kenntnisse der archaischen Programme SYSGEN und DDT nötig). Simple, relokierbare Objekt-Code Dateien für ein bestimmtes Modul können von jedermann auf jedem System benutzt werden.

Wir hoffen, daß NZ-COM der Allgemeinheit die Welt des Z-Systems erschließt, jenen, die kein Interesse haben, ihr eigenes Betriebssystem zu assemblieren, oder die nicht das Werkzeug oder die Fähigkeiten dazu besitzen. Wenn Sie sich überhaupt für das Z-System interessieren (wie könnte es anders sein?) und Sie einen Z80-Rechner mit CP/M-2.2 besitzen,<sup>1</sup> dann ist jetzt Ihre Chance zum Experimentieren gekommen.

---

<sup>1</sup>Wenn Sie einen CP/M-Plus-Rechner besitzen, bestellen Sie Z3PLUS, das automatische Z-System für CP/M-Plus-Computer.

## 1.1 Die Vorteile von NZ-COM

Hier sind einige der allgemeinen Vorteile, die man durch NZ-COM 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.
- Das NZ-COM-System ist gebrauchsfertig ("ready-to-run"). Assemblierung oder technische Installation ist nicht notwendig.
- NZ-COM 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 NZ-COM-Systems ist die Kontrolle und die Abarbeitung von Befehlen, die das CP/M-System steuern. Es sorgt für

- bequemes Editieren falsch eingegebener Befehlszeilen
- Rückholmöglichkeit für frühere Kommandos zur Wiederverwendung, Korrektur, oder Modifikation
- konditionale 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 zu 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 NZ-COM 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 Programmsammlung profitieren können, ohne daß Sie 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 NZ-COM 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 NZ-COM, 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 Nachschlagewerk. 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 NZ-COM gelieferten Dateien. Es ist das Kapitel mit den meisten technischen Details. Es gibt jedoch Material, das nur dort auftaucht und ab und zu sollte man dieses Kapitel mal überfliegen.

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

Während Sie sich weiter voranwagen mit NZ-COM, wünschen wir Ihnen aufregende und vergnügliche Erlebnisse. Wir hoffen, daß dieses Handbuch und Ihr Ausprobieren des NZ-COM 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 NZ-COM

### 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 gestattet es Ihnen, sich damit vertraut zu machen. Ein voll ausgebautes System wie dieses reduziert Ihre TPA (Speicher, der Anwenderprogrammen zur Verfügung steht) um 5.5K. Später (S. 48) werden Sie lernen, wie man Sonderkonfigurationen erstellt, so daß Sie Ihre eigenen Kompromisse wählen können zwischen System-Eigenschaften und Spe-

icherverbrauch und daß Sie Z-Systeme erzeugen können, die gerade 1K TPA abknapsen.

Diese Manual ist nur eine Einführung in das Z-System. Während Sie beginnen, von Ihrem neuen NZ-COM-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 NZCOM.COM, NZCOM.LBR, MKZCM.COM, ZCPR34.LBR, und NZ-COM 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 NZ-COM 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 NZ-COM-Distributionsdiskette** und verstauen Sie die Originaldiskette an einem sicheren Ort. Dann legen Sie die Arbeitskopie in eins Ihrer Zusatz-Laufwerke (nicht das A-Laufwerk) und loggen diese ein. Die Datei RELEASE.NOT, so vorhanden, enthält Zusatzinformation, die zum Druckzeitpunkt noch nicht vorlag. Sie sollten es 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". 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 NZ-COM 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 NZ-COM ohne Terminal-Definition verwenden, aber Sie können keine solchen Programme verwenden, die bestimmte Terminaleigenschaften voraussetzen, dazu gehören zum Beispiel SALIAS, SHOW, und ZFILER.

An diesem Punkt angelangt, wird angenommen, daß Sie Ihren Computer mit einem Standard-CP/M-2.2 Betriebssystem betreiben, womöglich mit einem CCP-Ersatz (wie etwa ZCPRI). Obwohl NZ-COM in einem manuell installierten Z-System betrieben werden kann, verschwendet das Speicher und wird nicht empfohlen.

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

NZCOM.COM	ARUNZ.COM	LSH.COM
NZCOM.LBR	ALIAS.CMD	ZFILER.CMD
MKZCM.COM	LX.COM	SHOW.COM
SALIAS.COM	IF.COM	myterm.Z3T
SDZ.COM	ZEX.COM	ZEX.RSX
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

---

<sup>1</sup>Wenn Sie die Bezeichnung "NZCOM" wählen, dann wird es später automatisch geladen, aber es wirft unseren Plan über den Haufen, Ihnen beizubringen, wie Sie das mit einem Alias (S. 14) machen können.



Ihr Terminal geeignete Version wählen. Für ein Dim-Video Terminal würde man zum Beispiel folgendes eingeben:

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

## 2.3 Der Betrieb von NZ-COM

Jetzt sind Sie bereit für eine schnelle zweistufige Prozedur des Definierens und Ladens des Standard-NZ-COM-Systems für Ihren Computer. Der Schritt der Definition wird von dem Programm MKZCM (MaKe ZCoM) ausgeführt. Einfach auf Laufwerk A: einloggen und eingeben:

```
MKZCM<cr>
```

Sie sehen ein Menu, das Sie später anwenden lernen (S. 51) um Ihre eigenen NZ-COM-Systeme zu kreieren. Im Moment werden wir lediglich die vorgegebenen Werte akzeptieren. Drücken Sie die "S"-Taste "Save", um zu speichern und geben Sie beim Prompt für einen Dateinamen "NZ-COM" ein. Das ist der Name, den NZCOM per Default definiert hat. MKZCM generiert zwei Dateien: NZCOM.ZCM und NZCOM.ENV. Die Bedeutung der Dateien wird später (S. 47) erklärt.

Geben Sie jetzt ein:

```
NZCOM<cr>
```

Ihr neues NZ-COM System wird generiert, geladen, und automatisch ausgeführt. Beim ersten Mal meldet NZ-COM

```
STARTZCM?
```

Diese Meldung können Sie für's erste ignorieren; sie bedeutet, das NZ-COM die Datei STARTZCM.COM nicht finden kann (wir haben sie noch nicht erzeugt). Damit befassen wir uns auf Seite 13.

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

```
NZCOM myterm.Z3T<cr>
```

Das war alles, damit NZ-COM läuft! Jetzt können Sie mit dem nächsten Kapitel fortfahren, um mehr über das System zu Lernen.

## 2.4 Die Organisation Ihrer Dateien

Nachdem Sie NZ-COM 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 großer Kapazität ist, können Sie fast alle NZ-COM- 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 NZCOM, 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 NZ-COM geladen ist, werden die Dateien NZCOM.COM und NZCOM.LBR erst wieder benötigt, wenn Sie die gegenwärtige System-Konfiguration ändern wollen, sie können daher auf eine separate "Boot Disk" verbannt werden.



## Kapitel 3

# Kennenlernen und Verstehen des Z-Systems

Wir empfehlen Ihnen, die folgenden Schritte wirklich auszuführen, um mit ein paar der normalerweise gebrauchten Features des NZ-COM 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 STARTZCM erzeugen, der automatisch Ihre Terminal-Datei in das NZ-COM System einbindet, wenn Sie das nächste mal NZCOM eintippen. (Sie haben diesen Namen STARTZCM schon einmal gelesen. Als Sie das erste Mal NZCOM starteten, war er auf dem Bildschirm zu sehen, gefolgt von einem Fragezeichen.)

Geben Sie ein

```
SALIAS STARTZCM<cr>
```

um das SALIAS-Programm zu starten und den STARTZCM-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

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

Wir haben hier die “/Q” Quiet Option benutzt, um die extensive Information zu unterdrücken, die NZCOM 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 eingeben von

```
STARTZCM<cr>
```

Auf dem Bildschirm sieht man, daß er tatsächlich das NZCOM 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 NZCOM diesen Alias automatisch ausführt, wenn es das nächste Mal gestartet wird. Probieren wir es soweit aus. Geben Sie ein

```
NZCPM<cr>
```

Jawohl, das NZ-COM-System ist weg und der Standard-CP/M-2.2 Kommando-Prozessor ist wieder aktiv. Geben Sie jetzt ein

```
NZCOM<cr>
```

um das NZ-COM-System zu starten. Dieses Mal wird es automatisch Ihre Z3T-Terminal-Datei laden, indem es den neuen STARTZCM-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 NZCOM.COM konfiguriert ist. Ein etwas ausgearbeitetes Beispiel finden Sie auf Seite 20. 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 Die verfügbaren Befehle

Geben Sie jetzt ein

```
H<cr>
```

Das wird Ihnen alle eingebauten Befehle des aktiven Z-Systems anzeigen. Der Bildschirm wird 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 NZ-COM 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 24, 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 20. 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 “peeked” 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. 22). 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 resettet das Disk-System; es ist wie Control-C, aber kann in eine Befehl-Sequenz 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

---

<sup>1</sup>Wenn Sie es ändern wollen, kann die Datei NZRCP.ZRL in NZCOM.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.



Verständnis, aber ist so mächtig und nützlich, daß wir glauben, daß es den Aufwand zu Lernen wert ist. In einer Nutshell ist es der Zweck der 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 vorgenommen, 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, Try einen Befehl wie “ECHO TESTING<cr>” und sehen Sie, daß es 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. Sind Sie 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 Flow Control und die FCP Befehle, um selbst zu experimentieren. Wenn Sie fertig sind, geben Sie ein "ZIF<cr>", um den Ablauf-Status zurückzusetzen.

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

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. 16). 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 NZCOM.COM<cr>
TEST NZCOM.LBR<cr>
TEST NOFILE.JNK<cr>
```

### 3.2.3 CPR Kommandos

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 verwenden Code, der schon für andere Zwecke existiert. GET und GO/JUMP sind die zwei grundlegenden Hälften des normalen Prozesses für Laden und Ausführen, 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 NZCOM.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>
```

<sup>2</sup> das Gleiche 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 oft 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

---

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

Grundgedanke hierbei ist, die Datei zu laden, indem man den GET-Befehl verwendet, sie modifiziert durch den POKE-Befehl und dann sie aufzurufen mit GO (or JUMP). 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:

```
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. 36) 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. Diese Dateien haben alle den Datei-Typ COM.

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, und type-4 angelegt.<sup>3</sup>

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, was man Z-System Environment Deskriptor nennt. Das ist ein Speicherbereich, das eine

---

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

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 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 NZ-COM 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 es mit der bestehenden Konfiguration laufen kann, lädt die Datei dorthin und ruft sie an dieser Adresse dann 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 und Fehlermanager (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 der mit NZ-COM gelieferten Programme 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 weitere Information über Ihr Sys-

tem. Die beste Art und Weise, etwas über die Konfiguration Ihres NZ-COM-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





# 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 NZ-COM-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 NZ-COM Komponenten

- Der ZCPR34-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. 16).

- 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. 17).
- Ein IOP (Input/Output Package): Es gestattet das bedarfsweise Laden von I/O Treibern um solche Möglichkeiten wie Tastatur-Makros oder Device- Umleitung von Bildschirm oder Druckerausgabe in eine Datei. <sup>1</sup>
- 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: Er 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 Der Erweiterte Kommando-Prozessor

Wenn der Kommando-Prozessor einen Befehl weder als Disk-Datei, noch als speicherresidentes Kommando finden kann, dann wird dieser an ein Programm übergeben, das als Erweiterter Kommando-Prozessor (ECP) bezeichnet wird. Die Standard-NZ-COM-Version

---

<sup>1</sup>Dieses Thema wird in diesem Handbuch nicht weiter behandelt. Fragen Sie den Händler, bei dem Sie NZ-COM gekauft haben nach Information über verfügbare IOPs.

des ZCPR34 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 NZ-COM-System beinhaltet zwei Programme, die oft als ECPs eingesetzt werden:<sup>2</sup>

- 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>3</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>4</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. 36) ARUNZ für einige Beispiele verwenden wollen. Später, wenn Sie lieber LX verwenden wollen, geben Sie ein:<sup>5</sup>

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

<sup>2</sup>Es gibt sogar einen Weg, beide als ECP zusammenwirken zu lassen.

<sup>3</sup>Diese Datei wird durch Einpacken häufig benötigter Befehle mittels LPUT selbst hergestellt

<sup>4</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>5</sup>Sie sollten das alte CMDRUN.COM entfernt haben

### 4.2.2 Error Handler

Wenn der Kommando-Prozessor einen Befehl nicht ausführen kann, weil der ECP nicht gefunden wurde, oder weil der Erweiterte 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 NZ-COM-Paket beinhaltet den ZERRLSH 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 ZEX — 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 dem NZ-COM-System beigelegt.

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 NZ-COM-Systems kann bis zu vier Shell-Befehlszeilen speichern, jede bis 32 Bytes lang, diese Werte können jedoch durch Ändern von MKZCM modifiziert werden. 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 Kommandozeile. 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 Eingabezeile in einer "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 STARTZCM Startup Alias hinzufügen. LSH wird mit ZERR als Error-Handler erst perfekt. LSH ist, so wie es mit NZ-COM geliefert wird ein Type-3 Programm (siehe S. 24), 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>6</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:

---

<sup>6</sup>HELPLSH kann mit ESC-J zur Hilfestellung aufgerufen werden

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

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>7</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 und der Taste "J".

---

<sup>7</sup>Wenn Sie einen AMSTRAD Rechner benutzen, überprüfen Sie, ob Sie die ESC-Taste wirklich über SETUP mit dem Zeichen 27 (ESC) belegt haben. Die CPCs haben dort normalerweise das Bömbchen, das Zeichen 252

### Die ZFILER Shell

ZFILER illustriert, wie eine Shell das Anwender-Interface vollständig verändern kann. Statt einer Befehlszeile stellt es eine "Point-and-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: COMMANDS. Drücken Sie die “L” Taste und geben Sie dann ein

```
COMMANDS<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 “Makro”-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 NZCOM.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,

---

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

wie ZFILER den Namen der (durch Zeiger, den sog. Tag markierten) bezeichneten Datei "parsen" kann.

Sie können jetzt ZFILER verlassen, indem Sie die "X" oder "eXit" Taste drücken. 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.

### Andere Shells

Es gibt eine ganze Reihe anderer Shells, die nicht im Standard NZ-COM-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 NZ-COM-Arbeitsdiskette (S. 8) 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. 13) 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.COMD. ARUNZ extrahiert einen bestimmten Alias-Script aus der ALIAS.COMD Datei, expandiert eventuelle Symbolische Parameter im Script, und übergibt die resultierende Kommando-Zeile an den Kommando-Prozessor.

Die Datei ALIAS.COMD 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.COMD 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. 28) 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.COMD 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 “\$td1” 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. Gleichmaßen bedeutet “\$tu1” 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, wodurch wir uns das ärgerliche Eingeben der Joker-Zeichen (\*.\*) sparen. Im Beispiel, erhalten wir alle Dateien, deren Name mit “S” beginnt und deren Typenbezeichnung mit “C” beginnt, einschließlich SALIAS.COM, STARTZCM.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 NZ-COM 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 NZ-COM-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

```
NZCOM //<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.

HELP ruft man wie folgt auf:

```
HELP<cr>
HELP 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

```
HELP SALIAS<cr>
```

Help-Dateien für eine Vielzahl anderer Z-System-Programme können in den Z-Nodes gefunden werden<sup>9</sup>. Sie können auch Ihre eigenen HLP-

---

<sup>9</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

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 NZCOM 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 NZ-COM System-Dateien zum Beispiel, befinden sich alle in der Library NZCOM.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 NZ-COM Paket eine spezielle, für CP/M Plus gepatchte Version beigefügt, die auch dann die freie Disk-Kapazität korrekt anzeigt. Andere Versionen sollten nicht mehr 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

### 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 NZ-COM-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>10</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

```
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

---

<sup>10</sup>früher hieß es einmal EDITND. Wir haben es umbenannt, um eine einheitliche Form mit SAVENDR zu erreichen.

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 STARTZCM-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 NZCOM geladen werden durch den Befehl

```
NZCOM SYS.NDR<cr>
```

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

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

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

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



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

### 4.3.6 Andere Tools

Eine Anzahl anderer Utility-Programme können auf Ihrer NZ-COM-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 HELP benutzen, um mehr über dieses Kommando zu lernen.

## 4.4 Kommando-Hierarchie

Der ZCPR34-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 NZ-COM 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 NZCOM 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 Suchpfads
5. Aufrufen des Erweiterten Kommando-Prozessors
6. Aufrufen des Error Handlers

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 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 NZ-COM Kommando-Prozessor, residente Kommandos zu überspringen.<sup>11</sup> Außerdem fügt es das aktuelle Verzeich-

---

<sup>11</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: ).

nis 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 angegeben 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 "?".

# Kapitel 5

## Wie man mehr aus NZ-COM herausholt

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

### 5.1 Alternativer Aufruf von NZ-COM

Bisher haben Sie die einfachste Methode kennengelernt, NZ-COM 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, NZCOM zu starten, kennengelernt, die in der Eingabe des einfachen Befehls besteht

```
NZCOM<cr>
```

NZCOM kann auch auf folgende Weise geladen werden:

- aus einem SUBMIT-Script
- aus einer speziellen Mehrfach-Kommandozeile

Eine NZCOM-Kommandozeile kann in einer SUBMIT-Datei enthalten sein, die entweder unter CP/M oder unter dem Z-System abläuft. Das Kommando NZCOM in einer SUB-Datei kann nützlich sein für den automatischen Aufruf von NZCOM als Teil Ihrer CP/M Kaltstart-Prozedur. Siehe dazu auch Seite 60 für Information über ausgefeiltere Verwendungsarten von SUBMIT mit NZ-COM.

NZCOM kann auch als Teil einer Mehrfach-Kommandozeile gestartet werden. Zum Beispiel:

```
NZCOM nzcom-parameter;Befehl 1;Befehl 2;... <cr>
```

startet NZCOM mit dem Parameter und dann das Kommando Befehl 1 nach dem “; usw. In diesem Fall wird das Default Start-Kommando (normalerweise STARTZCM) nicht ausgeführt. Zum Beispiel wird

```
NZCOM ;MYSTART<cr>
```

die Standard NZ-COM-Konfiguration laden und dann das alternative Startup Kommando MYSTART.

Bedenken Sie auch, daß diese Lade-Technik nicht mit der SUBMIT Lade-Technik kombinierbar ist.

## 5.2 Anpassen Ihres NZ-COM Systems

Es gibt viele Möglichkeiten, Ihr NZ-COM-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.

### 5.2.1 Vorübergehende Änderungen

Nachdem Sie sich mit den Grundlagen des NZ-COM-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 Druckerdaten 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 NZ-COM-System erstmals starten (und nach jedem Kaltstart).

### 5.2.2 NZ-COM Deskriptor-Dateien

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 NZ-COM-Systems sind definiert durch "Deskriptor"-Files in zwei alternativen Formaten, durch den Dateityp ZCM und den anderen vom Typ ENV. Dateien beider Typen werden durch das MKZCM-Utility (siehe S. 51). Sie können mit dem Format arbeiten, das Ihnen mehr zusagt; die andere Datei können Sie löschen, wenn Sie wollen.

Die ZCM-Dateien sind gewöhnliche Textfiles in Form einer Sysmboltable, die mit jedem x-beliebigen Text-Editor bearbeitet werden können (im NON-DOKUMENT Modus = ASCII).

An diesem Punkt empfehlen wir Ihnen, die Default Deskriptor-Dateien durch Eingabe von

```
TYPE NZCOM.ZCM<cr>
```

zu untersuchen.

Die ENV-Dateien sind Binär-Dateien in Form von Z-System "Environment-Deskriptoren". Sie können den Default ENV-Deskriptor untersuchen durch Eingabe des ARUNZ-Alias-Befehls

```
LOOK NZCOM.ENV<cr>
```

Die Bildschirmanzeige wird Ihnen nicht viel sagen, es sei denn, Sie sind schon vertraut mit den Z-System Programmier-Interns. Der einzige Vorteil der ENV-Form ist, daß die Datei nur einen Record (128 Bytes) lang ist, wohingegen die ZCM-Datei fünf Records lang ist. Es gibt ein paar Gelegenheiten, wo eine ENV-Datei benötigt wird, um ein anderes Z-System-Programm zu installieren. Im allgemeinen empfehlen wir das Arbeiten mit der ZCM-Datei.

### 5.2.3 Modifizieren der Deskriptor-Dateien

Das System-Definitionen-Utility MKZCM kann Änderungen nur in der Speicher-Zuweisungstabelle machen; viele andere Charakteristika können durch Editieren der System-Deskriptor-Datei geändert werden. Einige dieser Änderungen sollten Sie durchführen zur Optimierung des System-Deskriptors mit den Charakteristika Ihrer Hardware. Es ist eine gute Idee, diese Änderungen anfangs in der Datei NZCOM.ZCM

durchzuführen. Dadurch müssen Sie im allgemeinen keine weiteren Änderungen, außer durch Verwendung von MKZCM.<sup>1</sup>

Einige Charakteristika, die Sie höchstwahrscheinlich ändern wollen, damit Sie zur Hardware passen, werden nachfolgend behandelt.

### Laufwerke und Userbereiche

Der Laufwerk Vector (DRVEC) ist ein 16-Bit Wort, das dem System mitteilt, welches logische Laufwerk zulässig ist. Das niederwertigste Bit entspricht Laufwerk A. Wenn Sie also Laufwerk Eine und B, sowie Festplatten-Partitionen F, G, H, I, und RAM-Disk M, sähe der Drive Vector in Binärform so aus:

Laufwerk:	PONM LKJI HGFE DCBA
Bit:	0001 0000 1111 0011

In hexadezimaler Notation, wie es die ZCM-Datei erfordert, wäre dies 10F3 für dieses Beispiel. Der Defaultwert, den MKZCM von CP/M aus erzeugt ist FFFF (alle 16 CP/M-Laufwerke erlaubt).

Es gibt ein zweites Symbol, das das höchste logische Laufwerk in einem Z-System bestimmt. MAXDRV ist die Zahl des höchsten Laufwerks innerhalb des Systems, beginnend mit A=1. Im oberen Beispiel, würde dieses Symbol den Wert 13 dezimal oder 000D Hex erhalten. Der Defaultwert, den MKZCM von CP/M aus erzeugt ist 0010 (16 dezimal, alle Laufwerke erlaubt).<sup>2</sup>

Das Symbol MAXUSR spezifiziert die höchste Usernummer, die im Z-System erlaubt ist. Obwohl nur Userbereiche von 0 bis 15 eingeloggt werden können, weist CP/M-2.2 tatsächlich Dateien im Disk-Directory Usernummern 0 bis 31 zu. Daher ist der von MKZCM unter CP/M erzeugte Defaultwert 001F (31 dezimal), und es gibt selten Gründe, diesen Wert zu ändern.

---

<sup>1</sup>Das hängt damit zusammen, daß MKZCM immer die Werte des aktiven Systems ließt (wenn ein Z-System läuft) und diese Werte in die neue ZCM-Dateien und ENV-Dateien integriert.

<sup>2</sup>Die meisten Utilities wissen nichts vom Drive Vector und verwenden MAXDRV um das anzusprechende Laufwerk zu erkennen. Ebenso gibt es Situationen, in denen man eine Unterscheidungsmöglichkeit braucht zwischen Laufwerken, die für benannte Directory-Referenzen erlaubt sind (Drive Vector) und dem höchsten Laufwerk, auf das ein User mittels der Drive/User (DU:)-Form in Bezug auf MAXDRV zugreifen darf).



### Drucker und CRT Charakteristika

Das Z-System-Environment definiert einige physikalische Charakteristika von Drucker- und CRT-Devices. Vor ZCPR34 speicherte das Environment zwei CRT-Definitionen und vier Drucker-Definitionen. Die aktiven Definitionen wurden selektiert durch die Werte der Symbole CRT und PRT. Nun gibt es nur eine Definition für jedes Device (daher sollten die Symbole CRT und PRT nicht von ihrem Defaultwert 0000) abgeändert werden, aber das Utility CPSET kann benutzt werden, um eine beliebige Zahl von Definitionen zu erzeugen.

Die Default-Device-Charakteristika können durch die folgenden Symbole eingestellt werden:

- Für CRT, COLS für die Bildschirmbreite (Default 80 dezimal, 0050 hex), ROWS für die maximale Zahl an Zeilen (default 24 dezimal, 0018 hex), und LINS für die Zahl der Zeilen, die für die Textdarstellung verwendet werden sollen (default 22 dezimal, 0016hex)
- Für den Drucker, PCOL für die Zahl der zu druckenden Spalten (default 80 dezimal, 0050 hex), PROW für die Seitenlänge in Zeilen (default 66 dezimal, 0042 hex), PLIN für die Zahl der zu druckenden Textzeilen pro Seite, ohne Ränder (default 58 dezimal, 003A hex), und FORM, um die Fähigkeit des Druckers anzugeben, Formfeed zum Papiertransport verwenden zu können. wenn ja [default] (0001 oder 0000 wenn nicht)

### System Laufzeit Charakteristika

Manche Z-System-Software Zeitschleifen basieren auf dem Wert des Symbols SPEED. Setzen Sie es auf die Taktfrequenz der CPU in Ihrem Computer, gerundet auf ganze Megahertz. Wenn Ihre Machine auf 2.5 MHz getaktet ist, müssen Sie sich entscheiden, ob Sie lieber tiefstapeln (0002) oder übertreiben (0003); die Definition dieses Symbols gestattet keine "ehrliche" Angabe! Der Default Wert von MKZCM ist 0004.

Eine Zahl von Z-System-Utilities zeigen weniger oder gar keine Bildschirm-Information, wenn das System "Quiet Flag" gesetzt ist. Der Zustand dieses Flags beim Systemstart wird durch das Symbol QUIET bestimmt. Normalerweise ist sein Wert 0000 oder 0001 für "stille" Operation.

Z-System-Utilities und der Kommandoprozessor können Directory-Angaben der Form DU: erlauben oder zurückweisen. Wenn das Symbol DUOK auf den (Default-) Wert 0001 gesetzt ist, können Directories im Drive/User-Format eingegeben werden. Wenn der Wert 0000 ist, dann werden nur benannte Directories akzeptiert. Die letztere Möglichkeit wird selten benutzt und ist nicht zu empfehlen.

### Public Directory Spezifikation

Man kann die initiale Konfiguration von ZRDOS Public Directories angeben, die die Symbole PUBDRV und PUBUSR verwenden. Die acht niederwertigsten Bits jedes Worts werden benutzt. Für PUBDRV repräsentiert das niederwertigste Bit (LSB-Bit 0) Laufwerk A und das höchste (Bit 7) Laufwerk H. Setzen Sie die Bits für jedes Laufwerk, das Sie beim Systemstart zum Public-Laufwerk erklären wollen. Für PUBUSR repräsentiert das niederwertigste Bit (Bit 0) User 1 und das höchste (Bit 7) User 8. Setzen Sie die Bits für jeden Userbereich, den Sie als Public definieren wollen. Die Defaultwerte sind 0000 für beide Symbole, so daß kein Verzeichnis von Haus aus "Public" ist.

#### 5.2.4 Erzeugen neuer Definitionen mit MKZCM

Das Utility MKZCM stellt einen bequemen Weg dar, die Speicherzuordnung des Systems neu zu verteilen und völlig neue System-Deskriptoren zu beschreiben. Es erlaubt die Größenveränderung der System-Haupt-Moduln, einschließlich eines speziellen "User-"Puffers, der sehr gelegen kommen kann für Systemerweiterungen "oberhalb des BIOS", wie DateStamper oder RAM-Disks bzw. Festplatten-Laufwerke.

Sie können den Befehl

```
MKZCM //<cr>
```

wie gewohnt eingeben, um eine Hilfsseite mit Erklärung der Funktionen und Syntax von MKZCM zu erhalten. Aus der Hilfsseite, können Sie entweder zurück ins Betriebssystem, oder ins Programm gehen. Sie können auch jede der zwei folgenden Formen

```
MKZCM<cr>  
MKZCM name<cr>
```

eingeben, um direkt ins Programm zu gelangen:

In der zweiten Form ist "name" der Dateiname, der den ZCM- und ENV-Deskriptor-Dateien für die neudefinierten Systeme zugewiesen wird. In der ersten Form erscheint ein Prompt für einen Namen aus dem Programm heraus, wenn einer benötigt wird.

Wenn MKZCM läuft, präsentiert es ein Menü, das fast den gesamten Schirm ausfüllt und einen Prompt an der Fußzeile. Sie können auf zahlreiche Art antworten. Sie können jede der Zahlen (oder den Buchstaben "U") eingeben, die auf der linken Bildschirmseite erscheinen, um dasjenige System-Modul auszuwählen, dessen Größe Sie ändern möchten. In diesem Fall erscheint ein Prompt für einen neuen Wert der Modulgröße in Records (128 dezimal oder 80 hex).<sup>3</sup>

Die Spanne der akzeptierten Werte und der Wert, der bei einem simplen Return übernommen wird, ist abhängig vom zu definierenden Modul. Für viele der Moduln selektiert Return den Wert 0 (Null). Aufgrund des Zwangs, daß das BIOS immer am Anfang einer "page" (einem geradzahligem Vielfachen von 256 Bytes dezimal bzw. 100 HEX) beginnen muß, wird MKZCM manchmal automatisch einen zusätzlichen Record an ein Modul anfügt. Aus diesem Grund erscheint es zweckmäßig, sich von unten nach oben durch das Menü zu arbeiten.

Obwohl MKZCM es erlaubt, CPR- und DOS-Größen zu definieren, die nicht den CP/M-Standards entsprechen, empfehlen wir, daß der Durchschnittsuser die Moduln 1 bis 3 nicht verändern sollten. Fortgeschrittene User können interessante und nützliche Anwendungsfälle für Systeme mit größeren (oder kleineren) Kommandoprozessoren oder BDOS-Systemen finden.

Die Absicht beim Entwurf von NZ-COM war, daß der User einige verschiedene Systeme entwirft und das momentan passende jeweils lädt. Der nächste Abschnitt des Handbuchs befaßt sich damit, wie man zu gegebener Zeit von einem auf das andere System umschalten kann. Bevor wir auf dieses Thema kommen, möchten wir vorschlagen, einige Konfigurationen herzustellen, die Sie in Betracht ziehen sollten.

Reden wir zuerst über die "User's memory area" -den reservierbaren persönlichen Speicherpuffer des Anwenders. Als Beispiel, wollen wir beschreiben, wie man diesen Puffer nutzen könnte, um etwa DateStamper zu betreiben.<sup>4</sup> Die Version von DateStamper, die mit einem

---

<sup>3</sup>Die Definition des Shell Stacks ist eine Ausnahme. Einfach auf die Prompts antworten.

<sup>4</sup>Das ist eine Betriebssystemerweiterung für CP/M-2.2, die Zeit- und Datum-

allgemeinen BDOS läuft, erfordert 10 Records TPA. Sie kann automatisch konfiguriert werden und lädt sich selbst unterhalb des CCP, aber diese Methode schützt den CCP vor Überschreiben und verbraucht so zusätzlich 2K Speicher. Es ist wesentlich effizienter, DateStamper oberhalb des BIOS abzulegen.

Beim *echten* BIOS wird das Verlagern des BIOS nach unten erforderlich, um Platz zu schaffen für für DateStamper. Mancher Anwender fühlt sich hierbei überfordert und es bedeutet, daß der für DateStamper verbrauchte Speicher nie für Anwenderprogramme zur Verfügung steht. Mit NZ-COM können Sie sich eine System Definition schaffen, die einen Puffer für DateStamper und besitzt und eine andere ohne dieses Puffer.

Jetzt sollten Sie MKZCM aufrufen und die Beispiele durcharbeiten. Beachten Sie die Adressen der verschiedenen System Moduln. Jetzt definieren Sie einen Puffer für DateStamper durch Eingabe von "U" beim Prompt aus MKZCM und dann geben Sie ein

```
10<cr>
```

Beachten Sie, wie die Adressen für die System Moduln sich automatisch ändern, um Platz für den Puffer zu schaffen. An diesem Punkt sollten Sie die Start-Adresse des User-Puffers notieren. Dann beginnen wir die Prozedur für das Schaffen einer speziellen "über dem BIOS"-Version von DateStamper.<sup>5</sup> Dann können Sie DateStamper wie gewohnt laden. Sie sollten beachten, daß aber bei jedem Laden eines neuen NZ-COM-Systems das DOS und das BIOS neugeladen werden. Dadurch verlieren Sie die Einbindung von DateStamper und Sie müssen DateStamper neu starten. Dies kann automatisch erfolgen, indem man die Systeme durch Verwendung von Alias Scripts ändert (sehen Sie sich hierzu die Beispiele in ALIAS.CMD an).<sup>6</sup>

Manche allgemeine Wahl für die anderen Systemkomponenten könnte wie folgt aussehen.

Zum Beispiel kann man sowohl Systeme mit oder ohne IOP (Input-Output-Package=Ein-Ausgabe-Paket). Wenn man gewöhnlich ein IOP wie etwa das ausgezeichnete NuKey Tastatur-Macro IOP benützt, dann würde das Default-System einen IOP-Puffer beinhalten und die Defini-

---

seinträge verwaltet, um den Zeitpunkt der Erstellung, des letzten Zugriffs und der letzten Änderung zu indizieren.

<sup>5</sup>Siehe dazu auch das DateStamper Handbuch.

<sup>6</sup>Siehe S. 78 für weitere Möglichkeiten.

tion ohne Puffer könnte etwa NOIOP heißen. Benützen Sie meist kein IOP, dann würde die Definition ohne IOP Puffer den Defaultnamen NZCOM (ZCM und ENV) erhalten und die Version mit IOP könnte WITHIOP heißen.

Manchmal sucht man krampfhaft nach Möglichkeiten, um Arbeitsspeicher zu gewinnen. Für solche Anlässe ist ein System zweckmässig, vielleicht heißt es SMALL, das sowohl das IOP und das RCP entfernt, die zwei größten Hilfsmoduln. Wenn die Speichernot größer wird, kann ein System (vielleicht heißt es MIN) sogar das FCP und /oder das NDR, das Register für die benannten Verzeichnisse, entfernen.

Während Sie Erfahrung mit NZ-COM sammeln, werden Sie Verständnis für die System Definitionen entwickeln, die sich am besten für Ihre Zwecke eignen. Wenn sich Ihre Bedürfnisse ändern, können Sie leicht neu beginnen und bestehende Systeme löschen, oder noch weitere zu schaffen. Sie können sogar temporäre Definitionen aus dem Stand erzeugen und in weniger als einer Minute laden.

### 5.3 Laden neuer Systeme

Bisher haben das Erzeugen neuer Systeme behandelt; wir haben nicht beschrieben, wie andere Systeme als das Default-System namens NZCOM geladen werden. Erinnern wir uns, daß das Default-System geladen wird, wenn man den Befehl

```
NZCOM<cr>
```

eingibt. Dieser Befehl verwendet den System-Deskriptor NZCOM.ZCM (nur die ZCM-Version). Er lädt den Code und andere Moduln, die es braucht, entweder aus der Library NZCOM.LBR oder aus individuellen Dateien, die entlang des Suchpfades liegen können.

Der Kommandoprozessor (CPR), das DOS, und das Spezial NZCOM-BIOS werden geladen unter Verwendung der Default-Dateien NZCPR.ZRL, NZDOS.ZRL, und NZBIO.ZRL, respektive. Zusätzlich werden der RCP, (NZRCP.ZRL), das FCP (NZFCP.ZRL), und das IOP (NZIOP.ZRL) geladen, wenn die betreffenden Moduln in der System-Definition vorgesehen sind. Das letztere ist nur ein "Dummy IOP", das die grundlegenden Zeichen- Ein- und Ausgabenfunktionen übernimmt. Werden benannte Directories unterstützt, wird eine Datei namens NZCOM.NDR geladen und, so vorhanden, die Terminal-Deskriptordatei

NZCOM.Z3T ebenfalls.

Andere Systeme als die Default-Definition können auf vielfältige Art geladen werden. Zuerst, wollen wir eine Quickmethode beschreiben, die einfach anzuwenden ist und im allgemeinen das tut, was man erwartet. Bei dieser Methode verwenden wir die Befehlszeile:

```
NZCOM name<cr>
```

wobei "name" der Name einer ZCM- oder einer ENV-Datei ist, Sie mit MKZCM erzeugt haben.<sup>7</sup> Die System-Deskriptordatei dieses Namens wird benutzt, um das System zu definieren. Moduln werden genauso geladen, wie oben beschrieben.

Das Quick-Kommando kann nicht benutzt werden, um Systeme zu laden, die kleinere Moduln enthalten (abgesehen von Null-Files, die nur aus dem Namen bestehen). Zum Beispiel, wenn Sie ein System mit einem kleinen RCP definieren, (z.B., 10 records), dann wird ein Fehler entstehen, wenn NZCOM versucht, NZRCP.ZRL zu laden, da der aus dieser Datei erzeugte capsRCP nicht in den für den RCP reservierten Puffer paßt. Daher sollte die Quickmethode nur zum Laden von Systemen verwendet werden, aus denen Puffer entweder vollständig ausgelassen wurden, oder in denen die Puffer zumindest die Standardgröße der Defaults aufweisen.

Die allgemeinste Form der Ladeanweisung für NZCOM erlaubt enorme Flexibilität, viel mehr, als die meisten Anwender je brauchen. Siehe hierzu Seite 70 für vollständige Beschreibung. Jetzt werden wir lediglich die Befehlsform anmerken, die man zum Laden eines anderen als des Default-Systems verwenden sollte. Sie lautet

```
NZCOM name modul(n)<cr>
```

wobei "name" wie zuvor, den Namen der Deskriptor-Datei bedeutet und wo ein oder mehrere System Moduln zum Laden explicit angegeben werden können. Zum Beispiel,

```
NZCOM SMALL SMALLRCP.ZRL<cr>
```

würde ein System laden, das durch SMALL.ZCM (oder SMALL.ENV) beschrieben wurde und würde SMALLRCP.ZRL statt NZRCP.ZRL als RCP-Modul laden.

---

<sup>7</sup>Oder eine ZCI-Datei, die von NZCOM erzeugt wurde, wie im nachfolgenden Abschnitt beschrieben wird.

## 5.4 Clonen des Systems

Es gibt eine zusätzliche Methode, NZ-COM-Systeme zu laden, die den Vorteil größerer Ladegeschwindigkeit bietet, auf Kosten von Diskettenkapazität in Form größerer Dateien. NZCOM verfügt über eine "Clone"-Option, die die normale Systemerzeugungs-Prozedur (wie vom CP/M-Prompt) erlauben, aber mit dem Unterschied, daß das resultierende Environment in eine Datei geschrieben werden kann, statt es im Speicher zu installieren. Das Ergebnis ist eine Datei mit dem Namen der ZCM- oder ENV-Datei (oder NZCOM, wenn keiner angegeben wurde) vom Dateityp ZCI (für Z-Com Image). Zum Beispiel, erzeugt der Befehl

```
NZCOM /C<cr>
```

NZCOM.ZCI mit dem Default System Abbild (Image). Der Befehl

```
NZCOM SMALL SMALLRCP.ZRL /C<cr>
```

erzeugt SMALL.ZCI mit dem Abbild des Small Systems, einschließlich seines speziellen RCP-Moduls.

Systeme, die durch ZCI-Dateien definiert wurden, werden genauso geladen, wie jene, die durch ZCM- und ENV-Deskriptoren beschrieben wurden, durch hinzufügen ihrer Namen auf der NZCOM Befehlszeile. Zum Beispiel,

```
NZCOM SMALL.ZCI<cr>
```

lädt das Small-System in einer einzigen Operation, ohne daß man all die individuellen Dateien suchen und laden muß, die beim Erzeugen des SMALL.ZCI. Dies erlaubt das Laden des Systems in wesentlicher kürzerer Zeit. Es bedeutet auch, daß man die zum Erzeugen notwendigen Moduln (z.B. SMALLRCP.ZRL) nicht weiter auf der Diskette benötigt.

Wenn die Kurzform des NZCOM-Ladebefehls wie etwa

```
NZCOM SMALL<cr>
```

benutzt wird, wird als erster zu ladender Dateityp für SMALL ZCI, dann ZCM, und schließlich ENV. Der einfache Ladebefehl

NZCOM<cr>

ist jedoch ein Sonderfall und entspricht

NZCOM NZCOM.ZCM<cr>

Wenn Sie NZCOM.ZCI laden möchten, sollten Sie eine der folgenden Befehlsfolgen wählen:

NZCOM NZCOM.ZCI<cr>

NZCOM NZCOM.LBR NZCOM<cr>

Die erste Form lädt nur eine ZCI-Datei; die zweite Form zwingt lediglich NZCOM, seine normale Hierarchie ZCI, ZCM, und schließlich ENV einzuhalten.

## 5.5 Systemmodifikationen “im Flug”

Die vorangegangene Diskussion hat vielleicht den Eindruck vermittelt, daß Sie ein bestimmtes NZ-COM-System laden und dann für die Dauer der Sitzung damit leben müssen. Dies ist ganz und gar nicht der Fall. Im Gegenteil, eins der grundlegenden Design-Features von NZ-COM ist seine Fähigkeit, neue Versionen des Betriebssystems jederzeit nachzuladen, sogar inmitten einer Mehrfach-Kommandozeile.

### 5.5.1 Lade-Regeln

Wenn NZCOM vom CP/M-Befehlsprompt aus geladen wird (“Cold”), treten alle im Deskriptorfile angegebenen Werte in Kraft und alle System-Moduln werden geladen, wie oben beschrieben (S. 54). Auf der anderen Seite, wenn NZCOM aus einem bereits aktiven NZ-COM-System aufgerufen wird (“Warm”), werden nur die echten Veränderungen geladen.

Zuerst wird die ganze System-Information mit Ausnahme der Adressen und Größen der System-Moduln bewahrt, wann immer möglich. Shells und Error Handler laufen im neuen System weiter: der Suchpfad, Werte in Anwender-Registern, System-Dateinamen, Drucker-



und CRT-Charakteristika, etc. ändern sich nicht. Der Inhalt der Mehrfach-Kommandozeile wird in das neue System hinübergerettet. Das bedeutet, daß neue Konfigurationen als Teil einer komplexen Befehlssequenz geladen werden können, sogar mit konditioneller Ausführung.<sup>8</sup> Das System kommt dabei kein bißchen aus dem Tritt!

Zweitens werden die Inhalte der System Moduln (RCP, NDR, etc.) nicht verändert, außer die Adresse des Moduls hat sich geändert, oder die Größe des Moduls sich vermindert hat. Sogar die ausdrückliche Angabe eines Moduls auf der NZCOM-Befehlszeile wird das Laden nicht erzwingen; es wird nur geladen, wenn eine der eben genannten Bedingungen ein Laden erfordert.<sup>9</sup>.

### 5.5.2 Warum Systeme wechseln?

Welche Gründe gibt es für das "fliegende" Wechseln des Systems? Gewöhnlich heißt der Grund Speicherbedarf. Gäbe es unbegrenzten Speicher, würde man die größte, leistungsfähigste Version von NZ-COM laden, die man kriegen kann und mit dieser ständig arbeiten. Im wirklichen Leben, besonders bei einem CP/M-System mit seinem 64K-Speicher, müssen Zugeständnisse gemacht werden.

Die meiste Zeit würde man wahrscheinlich eine recht leistungsfähige Version des Z-System bevorzugen, eine, die alle System-Eigenschaften besitzt mit vielleicht der einzigen Ausnahme eines I/O-Pakets (IOP). Es wird jedoch vorkommen, daß ein Anwenderprogramm mehr Speicher benötigt, als diese Konfiguration erlaubt. Dann wäre es besser, ein kleineres System zu laden. Entsprechend würde man, wenn das Standard-System kein IOP besitzt, ein größeres System laden.

Zum Beispiel nehmen wir an, wir hätten ein Anwenderprogramm namens "BIGPROG", das viel Speicher benötigt (vielleicht eine Datenbank?). Wir könnten dann den Befehl

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

eingeben. Diese Befehlszeile installiert erst das "Small" NZ-COM-System mit seiner größeren TPA. Dann wird das Programm BIGPROG aufgerufen und schließlich, nach Beendigung von BIGPROG, wird das

<sup>8</sup>Vorausgesetzt natürlich, daß keine Prüfbedingungen abgearbeitet werden müssen, während ein System ohne FCP aktiv ist.

<sup>9</sup>Sie können ein Laden erzwingen, durch Aufruf von NZCOM im Modul-Lademodus, siehe S. 70

Standard NZ-COM-System wieder geladen. Abgesehen von der Zeiteinbuße, die damit verbunden ist, gibt es keinen Grund, vom Anwenderprogramm nicht benötigte Z-System-Features abzulegen, während dieses läuft.<sup>10</sup> Im obigen Beispiel ist die leistungsfähige Version des Z-System zurück an Ort und Stelle, wenn der Prompt für den nächsten Befehl erscheint. Soweit der Anwender merkt, war es eigentlich immer da.

### 5.5.3 Automatisierung der Systemwechsel

Der Vorgang des Systemwechsels kann in verschiedenen Stufen automatisiert werden, indem man sich die Fähigkeiten des Z-Systems zunutze macht. Die System-Ladebefehle wie im Abschnitt 5.3 beschrieben, können alle als Aliase ausgeführt werden oder man schließt sie in ALIAS.COM ein. Dann kann man in ein anders System, wie SMALL, wechseln durch die simple Befehlszeile

```
SMALL<cr>
```

Die Automation kann auf Situationen ausgedehnt werden, wo bestimmte Programme bestimmte Versionen des NZ-COM-Systems erfordern. Nehmen wir zum Beispiel an, daß BIGPROG.COM im Verzeichnis BIGDIR: steht und daß wir es in BGPRG.COM umbenennen. Wenn wir die folgende Script-Definition in unsere ALIAS.COM-Datei aufnehmen

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

dann können wir unseren Befehl wie gewohnt eingeben:

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

Der Extended Command Processor ARUNZ erledigt das Umschalten für uns automatisch. Verzinktere Scripts könnten sogar prüfen, ob die TPA schon groß genug für unser System ist und nur dann das System wechseln, wenn nötig. Es gibt kaum eine Grenze, was Einfallsreichtum aus der Flexibilität von NZ-COM machen kann!

<sup>10</sup>Wenige Anwenderprogramme machen von Z-System-Features Gebrauch. WordStar Release 4 ist die Ausnahme unter den Anwendungen aus den größeren Software-Häusern; es kann benannte Directories benutzen. In diesem Fall möchte man ein NZ-COM-System erzeugen, das lediglich einen NDR-Puffer enthält.

Hier sollten wir zwei wichtige Punkte erwähnen. Zuerst einmal ist NZCOM schlau genug, keine Lade-Operation auszuführen, wenn es erkennt, daß ein zu ladendes System sich vom gegenwärtigen nicht unterscheidet. Zum zweiten, wenn NZCOM beim Ladeversuch auf einen Fehler trifft, gibt es nicht einfach auf. Es gibt eine Fehlermeldung aus und weist auf die Art des Fehlers hin, *dann* erst ruft es den Error Handler auf. Auf diese Weise haben Sie Gelegenheit, irgendwelche Fehler zu korrigieren, bevor die Kommandozeile “weiterpflügt” und möglicherweise in eine Katastrophe (oder zumindest in große Unannehmlichkeiten) steuert.

#### 5.5.4 Automatische Rückkehr zu CP/M

Unter bestimmten außergewöhnlichen Umständen muß ein Programm im normalen CP/M ablaufen. Zum Beispiel braucht man jedes einzelne Bit TPA, das man kriegen kann. Oder man muß ein Konfigurationsprogramm laufen lassen, das direkte BIOS-Modifikationen ausführt, indem es Adressen mit einem Offset vom Warmboot-Vektor verwendet. Da unter NZ-COM der Warmboot-Vektor auf das virtuelle BIOS von NZ-COM zeigt, würden solche Utilities nicht korrekt arbeiten.

Um eine Stapelverarbeitung auszuführen, kann man SUBMIT verwenden, wobei das Z-System vollständig verlassen, unter dem puren CP/M ein Programm ausgeführt, und dann ins Z-System zurückgekehrt wird! Man muß jedoch sehr vorsichtig sein. Zum Beispiel muß man offensichtlich darauf achten, daß alle Kommandozeilen, die dann ausgeführt werden sollen, wenn das Z-System nicht in Betrieb ist, gültige CP/M-Befehle sind. Das bedeutet, nur ein Befehl pro Zeile und keine type-3 oder type-4 Programme. Wenn der Stapel-Script das Z-System wieder geladen hat, können wieder angemessene Z-System-Befehle, einschließlich Mehrfach-Kommandozeilen verwendet werden.

Ein weiterer Faktor, den man berücksichtigen sollte, ist, daß das NZCPM Sie im CP/M auf Laufwerk A User 0 zurückliefert, egal von wo es gestartet wurde. Da ZCPR3 (beginnend mit Version 3.3) seine SUBMIT-Datei ins Verzeichnis A0: statt ins aktuelle schreibt, gibt es normalerweise kein Problem bei der durchgehenden Abarbeitung unter CP/M<sup>11</sup>. Wenn Sie NZ-COM wieder laden, (es ist ein Cold-Load, einschließlich Ausführung von STARTZCM), werden Sie nicht automatisch

---

<sup>11</sup>Wenn das Bezugslaufwerk durch BIOS-“Verbiegung” verändert wurde, können hier Probleme auftauchen. In diesem Fall kontaktieren Sie bitte Ihren Z-System Händler

im ursprünglichen Directory landen. Sie müssen schon einen expliziten Befehl eingeben, um zurückzukehren.

Sie müssen außerdem sichergehen, daß Sie nur Versionen des ZCPR34 verwenden, die die Standard-SUBMIT-Verarbeitung unterstützen. Die Form "LONGSUB" der SUBMIT-Bearbeitung ist inkompatibel mit der SUBMIT-Bearbeitung von CP/M.

Hier ist eine Sequenz von Befehlszeilen, die in einer Datei namens CONFIG.SUB stehen könnte, um den automatischen Abkauf eines Utilities zu gewährleisten, das ursprünglich unter CP/M CONFIG.COM hieß .

```
A0:NZCPM
CONFIG0
NZCOM
$1:
```

Die erste Befehlszeile entfernt NZ-COM und restauriert CP/M auf User A0:. Die nächsten zwei Befehlszeilen laufen unter CP/M ab. Die erste ruft das Konfigurations-Utility auf, das wir in CONFIG0.COM umbenannt haben, die Gründe dafür werden wir gleich erläutern. Der zweite CP/M Befehl lädt die Standard Konfiguration von NZ-COM. Nach Beendigung des Befehls läuft das Z-System wieder, wir befinden uns aber noch immer im Verzeichnis A0:. Die letzte Zeile läuft im Z-System ab und ist dazu gedacht, uns in das Directory zurückzubringen, von wo wir gestartet waren.

Der Ausdruck "\$1" im letzten Befehl ist ein SUBMIT-Parameter. Wir haben angenommen, daß dieser SUBMIT-Script durch einen Alias namens CONFIG in der Datei ALIAS.COM ausgeführt wird. Der Alias-Script müßte die Form besitzen:

```
CONFIG    sub config $hb
```

Der ARUNZ-Parameter "\$hb" gibt uns das Ursprungs-Directory zurück, wenn der Alias in der Form "DU" aufgerufen wird. Dies wird als Parameter an den SUBMIT-Job übergeben.



# Kapitel 6

## Technisches Nachschlagwerk

### 6.1 Definition der Datei-Typen

Das NZ-COM-Betriebssystem identifiziert die Funktionen bestimmter Spezial-Dateien durch ihren Datei-Typ. Diese sind aufgeführt in der Tabelle 6.1 auf Seite 64. ZRL-Dateien sind spezielle relozierbare Dateien, die an eine beliebige NZ-COM-Systemkonfiguration angepaßt werden können. Diese können sowohl im SLR- als auch im Microsoft M-80-Format vorliegen. Die Moduln müssen gemäß der Tabelle 6.1 benannt werden, einschließlich einer Zeile im Quell-Code, die den NAME-Pseudo-op-Code enthält. Es gibt keinerlei Einschränkungen hinsichtlich der Namen, die für die Dateien selbst benutzt werden dürfen. Zum Zeitpunkt des Ladens bestimmt NZCOM die Art des Moduls durch Prüfung des im Code eingebetteten Modul-Namens.

### 6.2 Bei NZ-COM mitgelieferte Dateien

Viele Dateien werden beim NZ-COM-System mitgeliefert. Die meisten sind in diesem Abschnitt aufgeführt, zusammen mit Erklärungen ihrer Funktionen. Siehe dazu auch die Datei RELEASE.NOT wegen Information über kurzfristige Änderungen.

DATEI-TYP	INHALT
LBR	Library Datei
ZCM	NZ-COM System-Deskriptor
ENV	Z-System Environment und NZ-COM System-Deskriptor
ZCI	NZ-COM System Speicher Belegung
Z3T	Z3 Terminal Capability Deskriptor (TCAP)
NDR	Register-Datei der benannten Directories
ZRL	Z-system ReLozierbare allgemeine Code-Datei
	Datei mit REL-Modul namens:
	CCPxxx NZ-COM Kommando Prozessor
	DOSxxx NZ-COM Disk Operating System Modul
	BIOxxx NZ-COM BIOS-Modul
	RCPxxx Resident Command Package (eingeb. Befehle)
	FCPxxx Flow Command Package
	IOPxxx Input/Output Package
	wobei "x" irgendein Zeichen sein kann

Table 6.1: Tabelle der Datei-Typen, die vom NZ-COM-System benutzt werden.

### 6.2.1 NZ-COM System-Dateien

Die Dateien, die NZ-COM ausmachen, sind aufgeführt in der Tabelle 6.2 auf Seite 65. Wir möchten Sie erinnern, daß diese Dateien einem Copyright unterliegen und nur auf einen einzigen Anwender lizenziert sind. Es ist daher illegal, diese Dateien für andere Personen zu kopieren und an diese weiterzugeben. Einzelheiten dazu siehe Copyright-Notiz am Anfang des Handbuchs.

Der alternative RCP ist kleiner als der Standard-RCP und bietet ein sehr verschiedenes Befehls-Set, die im Hinblick auf Arbeitsgeschwindigkeit zusammengestellt wurden. Mit Ausnahme des R-Befehls, der fast keinen Platz im RCP beansprucht, beinhaltet er lediglich Befehle, die völlig speicherbezogen arbeiten und ohne Bezug auf eine Diskettenfunktion. Hier kann die Geschwindigkeits-Optimierung eines residenten Befehls voll ausgeschöpft werden. Diskettenorientierte Befehle, wie SP, TYPE, und ERA, werden transienten Befehlen überlassen.

Der alternative FCP kommt ohne IFQ-Befehl aus und ist daher einen

NZCOM	.COM	NZCOM-Systemlader
NZCOM	.LBR	Library mit NZCOM-System-Moduln
NZCPR	.ZRL	Default Kommando-Processor
NZDOS	.ZRL	Default Disk Operating System
NZBIO	.ZRL	Default virtuelles BIOS
NZRCP	.ZRL	Default Resident Command Package
NZRCP1	.ZRL	Alternatives Resident Command Package
NZFCP	.ZRL	Default Flow Command Package
NZFCP1	.ZRL	Alternatives Flow Command Package
NZCOM	.NDR	Default Definition benannter Directories
MKZCM	.COM	NZ-COM System-Definition-Utility
ZCPR34	.LBR	Alternative Kommando-Prozessor Moduln
JETLDR	.COM	Z-System Package-Lader

Table 6.2: Liste der NZ-COM System-Dateien.

Record kleiner. Wenn Sie den Gebrauch der Z-System-Ablaufsteuerung mit IF erst einmal gelernt haben, werden Sie vielleicht diese kleinere FCP-Version wählen. Wenn das der Fall sein sollte, können Sie einfach NZFCP1.ZRL in NZFCP.ZRL umbenennen (und den Standard-NZFCP.ZRL in etwa NZFCP2.ZRL) so daß dieser als Default geladen wird. Die gleiche Art der Umbenennung kann auch für den weiter unten beschriebenen RCP vorgenommen werden. Sie können die Library "Member-"Files umbenennen mit NULU oder VLU. Ist keins dieser Programme zur Hand, entpacken Sie alle Dateien aus der Library mit (LGET), nehmen die Umbenennung vor und erstellen eine neue Library mit (LPUT). Geübte können Dateien auch direkt in der LBR-Datei durch Patchen umbenennen.

### 6.2.2 Tools und Utilities

Viele Dateien, die nicht zum NZ-COM-System gehören, sind im Lieferumfang enthalten, *als Zugeständnis an die Anwender*.<sup>1</sup> Einige dieser Dateien sind Public Domain. Andere stehen unter Copyright ihrer Autoren oder der ZSIG, der Z-System Interest Group. Für alle gilt jedoch,

---

<sup>1</sup>Daher können wir keine Verantwortung zur Unterstützung für diese Programme übernehmen.



daß die Autoren ihre Zustimmung gaben zum Kopieren und kostenlosen Weitergeben an dritte, für *nicht-kommerzielle* Zwecke wohlgemerkt.

Die folgenden Dateien unterstützen die Z-System TCAP Einrichtung:

TCSELECT.COM	Auswahl des Terminal Deskriptors
Z3TCAP .TCP	Datenbank von Terminal Deskriptoren

Die folgenden Dateien unterstützen Z-System Aliase:

SALIAS .COM	Alias Comdatei-Generator
ARUNZ .COM	Alias Befehlsprozessor
ALIAS .CMD	Alias Script Datei für ARUNZ

Die folgende Datei wird vom FCP benutzt, um erweiterte konditionelle Prüfungen durchzuführen:

IF .COM	Erweiterter Flußzustands-Tester
---------	---------------------------------

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

CPSET .COM	zeigt/definiert CRT/PRT Charakteristika
PATH .COM	setzt/zeigt Befehls-Suchpfad <sup>2</sup>
SHOW .COM	Zeigt Z-System Konfigurations-Information
EDITNDR.COM	Editiert Named Directory Register i. Speicher <sup>3</sup>
SAVENDR.COM	“Save Named Directory Register” in Datei

Die folgenden Programme sind Utilities von allgemeinem Interesse:

FF .COM	Datei-Sucher
CRUNCH .COM	Datei-Komprimierer
UNCR .COM	Datei-Dekomprimierer

Folgende Programme unterstützen Library-Dateien:

LDIR .COM	Library Directory Programm
LGET .COM	Library Member Extraktor
LPUT .COM	Library Datei Inserter
VLU .COM	Video-orientiertes Library Utility

---

<sup>2</sup>Name aus dem ursprünglichen SETPATH.COM abgeändert

<sup>3</sup>Name vom ursprünglichen EDITND.COM geändert

LX .COM Library Datei Executor

Die folgenden Dateien beziehen sich auf Shells und Error Handler:

ZF-REV	.COM	ZFILER f. Terminals m. Inversdarstellung
ZF-DIM	.COM	ZFILER f. Terminals m. Helligkeiststufen
ZFILER	.CMD	Makro Script Datei für ZFILER
EASE	.COM	Kommando History-Shell und Error Handler
EASECMDCOM		erzeugt Datei mit EASE-Befehlstasten
VARPACKCOM		komprimiert EASE-History-Datei
LSH	.COM	Log-Shell - alternative History-Shell

Folgende Dateien unterstützen die Z-System-Help-Einrichtung:

HELP	.COM	zeigt menügesteuert Help-Dateien
xxx	.HLP	diverse Help-Dateien

Folgende Programme sind TYP-4 transiente Programme (siehe S. 24), die am oberen Ende der verfügbaren TPA laufen und die so den Speicher darunter nicht stören. Sie sind besonders nützlich in einem "Minimum-"System, wo transiente Programme für Funktionen benutzt werden, die andernfalls von RCP Befehlen übernommen werden müßten. Zum Gebrauch ist es zweckmäßig, sie umzubenennen, um die vorangestellte TY4-Kennzeichnung zu eliminieren.

TY4SP	.COM	Speicherplatz auf Diskette
TY4SAVE	.COM	SAVE-Befehl wie bei CP/M
TY4REN	.COM	REName
TY4ERA	.COM	ERAsE

Es gibt verschiedene Arten, wie man an die Unmengen von ständig zunehmenden Z- System User-Group Dateien kommt. Die Telefonnummern der Z-Node Mailboxen sind aufgelistet in:

ZNODES .LST

Eine Reihe von einzelnen Leuten haben mitgeholfen, bei anderen das Z-System zu installieren und die Anwendung zu unterstützen. Ihre Namen, Adressen, und Telefonnummern sind aufgeführt in der Datei

ZHELPER.LST

## 6.3 NZCOM Befehlszeilen

Dieser Abschnitt beschreibt die verschiedenen Arten der Befehlszeile, die zum Aufruf von NZCOM.COM eingesetzt werden können.

### 6.3.1 Hilfe-Seiten

Eine eingebaute Hilfeseite zur Erinnerung an die Syntax der NZCOM Befehle wird dargestellt, wenn einer der folgenden Befehle eingegeben wird:

```
NZCOM //<cr>  
NZCOM ?<cr>
```

Diese Seite wird auch anzeigen, wie die momentane Version durch Patchen der Datei (siehe S. 75) konfiguriert wurde. .

### 6.3.2 Laden des NZ-COM Systems

Die allgemeine Form der NZCOM Befehlszeile ist

```
NZCOM [Library] [Deskriptor] [Dateiliste] [/Optionen]  
[:Befehle]<cr>
```

Alle Elemente auf der Befehlszeile sind optional. Wenn mehrere Eingaben gemacht werden, kann man diese trennen durch Leerzeichen, (wie dargestellt), Kommata, oder Kombinationen beider Formen. Für verschiedene Teile der Befehlszeile können verschiedene Separatoren benutzt werden.

### Allgemeine Regeln

Bevor wir die einzelnen Bestandteile der Befehlszeile detailliert beschreiben, wollen wir zwei allgemeine Regeln definieren. Erstens: Ein Präfix namens "dir:" in einem Syntax-Ausdruck bedeutet eine geeignete Directory-Angabe. Wenn NZCOM von CP/M aus aufgerufen wird, kann der Directory-Präfix jede der folgenden Drive/User Formate haben: Doppelpunkt allein für das gegenwärtige Directory, D:, U:, oder

DU: Ist das Z-System bereits aktiv, dann dürfen auch benannte Directories benutzt werden.

Zweitens: Dateien, die auf der NZCOM-Befehlszeile angegeben werden, werden nach folgender Prozedur gesucht. Hat die Datei einen spezifischen Directory-Präfix, wird nur dieses Directory durchsucht. Wurde kein expliziter Directory-Präfix verwendet, wird zuerst die momentan ausgewählte Library-Datei (mehr darüber gleich) durchsucht. Wird die Datei dort nicht gefunden, wird der "Suchpfad" abgesucht. Ist das Z-System bereits aktiv, dann steht der Ausdruck "Path" für den Z-System-Suchpfad, einschließlich des momentanen Directories. Unter CP/M bedeutet "Path" einen internen Suchpfad, der in NZCOM.COM verankert wurde (siehe S. 75). Diesen Pfad kann man sehen, indem man NZCOM mit der Hilfe-Seite aufruft.

### **Der Ausdruck Library**

Der erste Teil der Befehlszeile ist optional die Angabe einer Library-Datei. Er ist von der Form

```
[dir:]lbrname.LBR
```

Der Datei-Typ LBR wurde angegeben. Wird dabei kein Library-Name explizit angegeben, wird der Ausdruck "NZCOM.LBR" für diesen Teil der Befehlszeile angenommen. Directory-Präfixe sind optional.

Libraries können ebenfalls als Teil einer Dateiliste angegeben werden, wie wir bald sehen werden. In allen Fällen sucht NZCOM nach Library Dateien wie nach jeder anderen Datei, außer daß es natürlich nicht als "Member" einer anderen Library gesucht wird. Der zuletzt angegebene (oder eingebaute) Library-Name wird in der Such-Hierarchie für andere Dateien benutzt. Eine Library, die am Beginn einer Parameterzeile steht, wird nach allen Deskriptor-Dateien durchsucht.

### **Der Ausdruck Deskriptor**

Der zweite Teil ist eine optionale NZ-COM-System Deskriptor-Datei. Auch diese kann optional einen Directory-Präfix beinhalten. Sie wird dadurch identifiziert, daß sie entweder keinen Datei-Typ hat, oder einen der drei System Deskriptor-Datei Typen ZCI, ZCM, oder ENV. Wurde kein Datei-Typ explizit genannt, dann findet die Such-Prozedur in der

Reihenfolge, in der die drei Datei-Typen gerade aufgeführt wurden sequentiell Anwendung. Erscheint kein System-Deskriptor in der Befehlszeile, geht NZCOM in den “*System-building mode*” über. Andernfalls findet der *Modul-Lade-Modus* statt.

Wenn keine Dateiliste (siehe unten) angegeben wurde, dann wird dafür im allgemeinen der Ausdruck “NZCOM” als System-Deskriptor eingesetzt. Eine Ausnahme tritt dann ein, wenn gar keine Dateien als Parameter angegeben wurden. Dann wird die Dateispezifikation als

```
NZCOM NZCOM.LBR NZCOM.ZCM
```

angenommen, wobei der ZCM-Datei-Typ statt ZCI benutzt wird, wie in der oben angegebenen allgemeinen Regel.

### Der Ausdruck Dateiliste

Der dritte Teil ist eine optionale Liste eines oder mehrerer Moduln oder Library-Dateien der Form

```
Dateiname [Dateiname [...]]
```

Jeder Dateiname hat die Form

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

wobei die erlaubten Dateitypen LBR, ZRL, REL, Z3T, NDR, FCP, RCP, oder IOP sind.

Wurde eine Library-Datei benannt, wird sie zur aktiven Library und wird für die Suche nach Modul-Dateien in der Folge eingesetzt.

Die ZRL-Datei ist eine spezielle Form relozierbaren Codes, der für den Gebrauch im Z-System konfiguriert werden kann, wenn die Datei verwendet werden soll (im Gegensatz zum Entstehungszeitpunkt). Es ist die einzige Dateiart, die zum Laden von Kommando-Prozessor, Disk Operating System, oder virtuellen BIOS-Moduln verwendet werden kann. Die ZRL-Datei ist ein bequemer Weg, andere Code-Moduln, wie FCP- und RCP-Moduln zu versorgen, da eine einzelne ZRL-Datei in jedem Z-System verwendet werden kann, das dafür Speicher reserviert hat. Wir ermutigen Entwickler, Dateien des Typs ZRL die Endung ZRL zu geben. Es können jedoch Dateien des ZRL-Typs mit der Endung REL geladen werdend.

NZCOM kann gleichwohl verschiedene Formen nicht-relozierbarer Dateien laden. Z3T Terminal Capability Deskriptor-Dateien und NDR Named Directory Register Dateien stellen nie ein Problem dar, da diese Moduln keinen Code enthalten; sie enthalten nur Daten. NZCOM kann auch absolute Speicher-Abbilder von FCP-, RCP-, und IOP-Code Moduln laden. Das ist eine bleibende, riskante Prozedur in einem dynamischen System wie NZ-COM, wo die Adressen von System-Komponenten sich ändern können. Aus diesem Grund empfehlen wir nachhaltig den Gebrauch von ZRL-Dateien. NZCOM unternimmt einige Anstrengungen, um diese absoluten Moduln auf Kompatibilität mit dem aktiven oder dem zu konstruierenden NZ-COM-System zu prüfen, aber kein Test kann diese Kompatibilität garantieren.

NZCOM bearbeitet die Dateiliste auf zwei verschiedene Arten, je nachdem, in welchem Modus es sich gerade befindet. Im Modul-Lade-Modus wird jede in der Liste genannte Datei direkt in das momentan aktive System geladen. Im "System-building"-Modus baut es daraus ein Abbild des neuen Systems.

Im System-building Modus, wenn ein NZ-COM-System bereits aktiv ist, werden Moduln aus der Liste nur geladen, wenn sich das neudefinierte NZ-COM-System vom aktiven NZ-COM-System auf eine Weise unterscheidet, die das Laden des neuen Moduls erforderlich macht, besonders, wenn dies Adresse des Puffers, an die das Modul geladen werden soll, sich geändert hat, oder wenn dieser Puffer kleiner geworden ist. Werden irgendwelche Moduln benötigt, die nicht angegeben wurden, benutzt NZCOM seine Standard-Default-Modulnamen, und sucht es diese Dateien, als ob sie explizit am Ende der Dateiliste angegeben wären. Diese Moduln sind wie folgt:

- NZCPR.ZRL als Kommando-Prozessor
- NZDOS.ZRL als Disk Operating System
- NZBIO.ZRL als virtuelles NZ-COM-BIOS
- NZFCP.ZRL als Flow Command Package
- NZRCP.ZRL als Resident Command Package
- NZIOP.ZRL als Input/Output-Package
- NZCOM.NDR als Named Directory Register
- NZCOM.Z3T als Terminal-Capabilities-Deskriptor

Wurde die Option “C” (siehe unten) angegeben, dann ist es NZCOM egal, ob gerade ein Z-System aktiv ist. Es verhält sich, als ob CP/M aktiv wäre und lädt alle Modul-Dateien in den Speicher.

### Der Ausdruck Optionen

Der nächste Teil der NZCOM-Befehlszeile ist ein optionaler “Slash” (Vorwärts-Schrägstrich), gefolgt von einem oder mehreren Buchstaben. Die Schalter “Q” oder “V” setzen den “quiet” oder “verbose” Modus. Im Verbose-Modus (=wortreich), zeigt NZCOM Information über die Moduln, die geladen werden, einschließlich des Namens und, wo angebracht, der Zieladresse der verwendeten Datei. Im Quiet-Modus wird diese Anzeige unterdrückt und nur eine Reihe von Punkten wird dargestellt. Dies teilt Ihnen mit, daß etwas passiert und daß etwas vorangeht. Die NZCOM-Hilfs-Seite zeigt den Default-Modus für diesen Schalter. Der Quiet-Modus ist in der Vertriebsversion von NZCOM.COM Default, kann aber vom Anwender geändert werden (siehe S. 75).

Ein anderer Options-Schalter ist “C” (für “Clonen”). Wenn dieser Schalter angegeben wurde, verhält sich NZCOM als ob es aus der CP/M-Befehlszeile aufgerufen worden wäre und schafft ein komplettes System. Es lädt dieses System jedoch nicht. Statt dessen, erstellt es eine Datei mit der Bezeichnung der auf der Befehlszeile angegebenen NZCOM-Deskriptor-Datei (oder per Default NZCOM). Diese Option kann natürlich nur im “System-building”-Modus benutzt werden.

Eine dritte Option ist eine der Kombination “Z” und “R”. Damit wird festgelegt, ob NZCOM eine Datei-Extension ZRL oder REL für Dateien des Typs ZRL als *Default*-Dateien lädt (siehe obige Liste der Default-Dateien), wenn es ein Modul ist, das Sie nicht ausdrücklich angegeben haben. Wir empfehlen den Gebrauch dieser Option nicht sondern raten Ihnen, die Dateien umzubenennen, so daß die Extension ZRL wird und damit den Datei-Typ eindeutig als ZRL festlegen.

### Der Ausdruck Befehlszeile

Die letzte Option in der NZCOM-Befehlszeile ist ein Strichpunkt (Semikolon), gefolgt von einem Ausdruck, der aus einer Multiplen Z-System Befehlszeile besteht. Da ein Semikolon unter dem Z-System eine Befehlszeile abschließt, ist diese Option bedeutungslos, wenn das

Z-System bereits läuft.<sup>4</sup> Dies soll dazu dienen, dem Anwender den Gebrauch Multipler Befehlszeilen beim Aufruf von NZCOM aus CP/M direkt anzuhängen.

### 6.3.3 Das Entfernen von NZ-COM

Wenn NZCOM aus CP/M aufgerufen wird, wird die Existenz einer Datei namens NZCPM.COM im Root Directory des intern in NZCOM.COM (siehe S. 75) konfigurierten Pfads abgeprüft. Wurde keine solche gefunden, wird ein Speicherabbild des CP/M-Systems zusammen mit einem Lader in eine Datei gespeichert. Diese Datei heißt NZCPM.COM. Das NZ-COM Betriebssystem kann jederzeit aus dem System entfernt und das CP/M Betriebssystem wiederhergestellt werden durch Eingabe von<sup>5</sup>

```
NZCPM<cr>
```

### 6.3.4 Beispiele und Tips

Wir wollen Ihnen jetzt einige Beispiele von NZCOM-Befehlszeilen geben, um Ihnen das Verständnis der Syntax zu erleichtern. Während wir diese Beispiele diskutieren, wollen wir einige Tricks zeigen, die das Arbeiten mit NZCOM vereinfachen oder beschleunigen.

#### Laden einer speziellen NZ-COM-Konfiguration

Die folgende Befehlszeile kann benutzt werden, um eine besondere, mit SMALL.ENV erstellte Konfiguration zu laden. Es wird angenommen, daß sich die Definition und die Moduln SMALLFCP.ZRL und SMALLRCP.ZRL, die es benutzt, in der Library SPECIAL.LBR befinden, die sich im Directory A15: befindet. Außer diesen speziellen Moduln, sind alle anderen Moduln die Default-Dateien, die sich wie gewohnt in der Datei NZCOM.LBR auf A0: befinden. Hier die Befehlszeile, um dieses System zu laden:

---

<sup>4</sup>Wann immer NZCOM eine neue Version des Z-Systems lädt, werden schwebende Befehle im Befehlszeilen-Puffer des laufenden Systems in den Befehlszeilen-Puffer des neuen Systems übergeben. Als Resultat scheint die hier beschriebene Option genauso unter einem laufenden Z-System zu funktionieren.

<sup>5</sup>Hierbei wird davon ausgegangen, daß NZCPM sich in einem Directory entlang des Suchpfads befindet. Wenn nicht, müssen Sie eine Directory-Angabe davorstellen.



```
NZCOM A15:SPECIAL.LBR SMALL.ENV SMALL-
FCP.ZRL SMALLRCP.ZRL A0:NZCOM.LBR /V<cr>
```

Wir haben Laufwerksangaben für beide Libraries angegeben, damit NZCOM nicht Zeit verliert bei der Suche nach diesen entlang des Pfads.

Durch Angabe eines expliziten Datei-Typs im Ausdruck SMALL.ENV, erhalten wir zwei Vorteile. Zuerst versichern wir uns, daß Dateien mit den Namen SMALL.ZCI oder SMALL.ZCM nicht geladen werden (hier entgegen unserer Absicht). Zum Zweiten, für den Fall, daß die Datei unglücklicherweise nicht in der Library gefunden wird, hindern wir NZCOM daran, eine langwierige Suche nach Dateien mit allen drei Deskriptor-Datei-Typen über den gesamten Suchbereich durchzuführen, bevor der Fehler gefunden und gemeldet wird.

*Das Hinzufügen des Ausdrucks A0:NZCOM.LBR am Ende der Dateiliste ist sehr wichtig.* Ohne das, wäre die Library am Ende der Liste A15:SMALL.LBR gewesen. Die Default-Dateien für die verbleibenden Moduln würden nicht gefunden, da NZCOM nicht automatisch in NZCOM.LBR nachsehen würde.

Die Option “/V” am Ende stellt sicher, daß NZCOM eine detaillierte Lade-Anzeige ausgibt, die das generierte System zeigt.

Wenn man die System-Konfiguration, die durch diese Befehlszeile definiert wurde, mehr als nur ein- zweimal benutzt, wäre es sinnvoll, diese Zeile in die Datei ALIAS.CMD einzubinden, etwa unter einem Alias-Namen wie SMALL. Das würde uns eine Menge Tipperei ersparen und eine Menge Fehler vermeiden helfen, wie etwa, zu vergessen, den Ausdruck NZCOM.LBR anzugeben, oder einen Dateinamen falsch anzugeben.

### Laden Spezieller Moduln

Der folgende Befehl wird benutzt, um eine neue Gruppe von Kommando-Moduln zu laden. Er setzt voraus, daß wir unsere RCPs in einem Directory namens RCP entwickeln und daß wir die fertig gestellten in einer Library namens RCPS.LBR ablegen. Er setzt voraus, daß wir nur ein paar Versionen von FCPs haben und daß wir es vorziehen, sie als einzelne Dateien in unserem Hauptverzeichnis A0: zu lagern.

```
NZCOM          RCP:RCPS.LBR          RCP-D.ZRL
A0:FCP-1.ZRL<cr>
```

Durch gezielte Angabe von A0: vor dem Ausdruck FCP-1.ZRL erreichen wir einen Geschwindigkeitsvorteil. NZCOM würde normalerweise die Datei zuerst in RCPS.LBR suchen und nur dann entlang des Pfads suchen. Durch Angabe des korrekten Directory-Pfads bringen wir NZCOM dazu, die Datei sofort zu finden.

### Laden eines neuen Kommando-Prozessors

Benutzer des Z-Systems sind an die Vorstellung gewöhnt, neue RCP und FCP Moduln zu laden. Es mag überraschen, daß NZCOM genauso einfach einen neuen Kommando-Prozessor laden kann. Das NZCOM-Vertriebspaket beinhaltet eine Library namens ZCPR34.LBR, die diverse alternative Kommando-Prozessor Konfigurationen enthält. Einer davon unterstützt eine nicht standardisierte Form der SUBMIT-Bearbeitung, die willkürliche Längen von SUBMIT- Dateien erlaubt.<sup>6</sup> Um diese spezielle Datei namens Z34LONG.ZRL zu benutzen, geben wir die Befehlszeile ein

```
NZCOM A0:ZCPR34.LBR Z34LONG.ZRL<cr>
```

Auf lange Sicht gesehen wäre es vielleicht nützlich, einen Alias in ALIAS.CMD unter einem Namen wie LSUBCPR zu definieren, der diese Befehlszeile für uns erzeugen könnte.

## 6.4 Patchen von NZCOM.COM

Es gibt verschiedene Optionen in NZCOM.COM, die von den Konfigurations- Bytes im Code kontrolliert werden. Man kann Änderungen auf vielerlei Arten installieren. Die bequemste Methode ist durch ein Patch-Utility wie ZPATCH. Man kann auch die zur Verfügung stehenden Gegebenheiten des Z-System verwenden. NZCOM kann mit dem GET Befehl in den Speicher laden, mit dem POKE-Befehl ändern und dann das modifizierte Speicherabbild mit SAVE speichern. Bevor Sie das tun, sollten Sie jedoch wissen, wie groß die Datei NZCOM.COM in Records ist. Das Directory-Programm SDZ mit der "/C" Option wird diese Information liefern.

---

<sup>6</sup> Normale CP/M SUBMIT-Dateien dürfen nicht mehr als 128 Befehlszeilen enthalten, die Anzahl, die man in einem einzigen Datei-"Extent" unterbringen kann.

Wir werden jetzt die momentanen Konfigurations-Bereiche en detail betrachten. Diese Information ist korrekt für die ursprünglich veröffentlichte Version von NZ-COM und es ist sehr leicht möglich, daß zukünftige Versionen neue oder zusätzliche Konfigurations-Optionen bieten. Sie sollten stets die Datei RELEASE.NOT, sofern vorhanden, auf aktuellere Information durchforsten.

### 6.4.1 Der Interne Suchpfad

Der "interne" Suchpfad, der dann benutzt wird, wenn NZCOM von CP/M aus operiert, der initiale Z-SystemSuchpfad, der entscheidet, wohin die CCP "Image Datei" NZCOM.CCP und der NZ-COM System-Rücklader NZCPM.COM gespeichert werden, werden an die Adresse 0680H geladen.<sup>7</sup> Jedes Element im Suchpfad wird durch ein Bytepaar ausgedrückt. Das erste Byte des Paares ist das Laufwerk, wobei der Wert 1 für Laufwerk A steht. Das zweite Byte ist die User-Nummer und muß einen Wert zwischen 00 und 1F (31 decimal) aufweisen.<sup>8</sup>

Es gibt zwei besondere Symbole, die in einem Pfad Ausdruck Anwendung finden. Ein Dollar-Zeichen (24 Hex) für ein Laufwerk wird zum Zeitpunkt der Anwendung des Pfad-Ausdrucks ersetzt durch das momentan eingeloggte Laufwerk. Gleichermaßen repräsentiert ein Dollar-Zeichen an der Position des Usereintrags den momentan eingeloggten Userbereich.

Es können maximal fünf Elemente im Pfad enthalten sein. Er wird terminiert durch ein einzelnes Null-Bytes (value 00). Das Ende der ganzen Pfadsequenz im NZCOM-Code ist markiert durch FF. Der Suchpfad in der Vertriebsversion von NZCOM.COM enthält die Sequenz A0 \$\$ A0. Die verbleibenden zwei möglichen Pfad-Elemente werden mit Nullen aufgefüllt.

Sie werden sich fragen, warum das Directory A0 zweimal im Pfad enthalten ist. Das letzte Drive/User Paar heißt Root Directory.<sup>9</sup> Viele Z-System-Operationen beziehen sich automatisch auf dieses Directory.

<sup>7</sup>Alle Adressen sind in hexadezimaler Schreibweise und erwarten einen Programmbeginn bei 0100H.

<sup>8</sup>Bedenken Sie, daß Sie unter CP/M Plus nur 15 User ansprechen können. Userbereiche darüber sind daher nur in reinen CP/M 2.2 Systemen mit speziellen Utilities erreichbar.

<sup>9</sup>Bitte dieses Root Directory nicht mit einem Directory verwechseln, das zufällig ROOT heißt. Letzteres ist lediglich ein Name und hat keine besondere Bedeutung für das Directory.

Der ZCPR34-Kommando-Prozessor ist gewöhnlich so konfiguriert, daß er im Root Directory nach dem Erweiterten Kommando-Prozessor sucht, und ARUNZ und ZFILER suchen dort nach ihren CMD-Dateien.<sup>10</sup> NZCOM benutzt den "Root" dieses internen Pfads um seine Dateien NZCOM.CCP und NZCPM.COM abzulegen.

Auf der anderen Seite enthält das als Root eingestellte Directory oft die meistbenötigten Programme und im allgemeinen möchte man ein Programm dort zuerst suchen lassen. So ist NZCOM eingestellt, zuerst auf A0 zu suchen *und* es als Root zu benutzen. Pfad-Ausdrücke mit Dollar-Zeichen kann auch Doppelseinträge im Pfad hervorrufen. In der Standardkonfiguration hat der Pfad *drei* Einträge von A0 wenn man auf A0 eingeloggt ist. Wird bei der ersten Suche nach einer Datei nichts gefunden, wird eine zweite Suche sicher nichts fruchten. Sowohl NZCOM, als auch der ZCPR34-Kommando-Prozessor sind clever genug, eine zweifache Suche zu unterbinden durch einen sogenannten "Minpath". Minpath ist der verbleibende Suchpfad, nachdem alle Doppelseinträge entfernt worden sind.

### 6.4.2 Die Default-Optionen

NZCOM kann eingestellt werden, entweder Q oder V und sowohl Z oder R als Default Option für die beiden Optionspaare anzunehmen. Der "Verbose"-Options-Schalter VOPT<sup>11</sup> wird an der Adresse 068D gespeichert. Der Wert 00 selektiert den "Quiet"-Modus als Default; FF selektiert den "Verbose"-Modus.

Die für ZRL-Dateien von NZCOM benutzte Datei-Extension wird von dem ZOPT Schalter an der Adresse 068E bestimmt. 00 selektiert die R Option; der Wert FF selektiert die Z Option.

### 6.4.3 Die Startup Befehlszeile

Die Default Startup Befehlszeile, die immer dann benutzt wird, wann immer das NZ-COM-System aus CP/M geladen wird, bleibt an der Adresse 0690 gespeichert. Sie besteht aus den Zeichen, die die Befehlszeile formen, gefolgt von einem Null-Byte (00), um

---

<sup>10</sup>Die Lieferversionen von ARUNZ und ZFILER sind konfiguriert, ihre CMD-Dateien im Root Directory zu suchen. Diese Programme können, wenn gewünscht, konfiguriert werden, ein anderes Directory zu benutzen.

<sup>11</sup>verbose=geschwätzig

den Kommando-String abzuschließen, und dann einem FF, um den gesamten String abzuschließen.

Diese Befehlszeile ist normalerweise so eingestellt, daß sie den String STARTZCM enthält. Dies wird dann ein ALIAS sein, der viele andere Befehle enthält. Das macht es leichter, die Startup-Befehle zu ändern, als wenn eine komplexe Befehlsfolge starr in NZCOM.COM verankert wäre. Es ist gerade genug Platz in NZCOM.COM — ohne die Copyright Notiz zu überschreiben — für einen Befehlsnamen mit acht Zeichen. Ein Teil der Copyright-Notiz kann durch die Befehlszeile überschrieben werden, sollten Sie darauf bestehen, einen längeren Befehl fest einzubinden (oder wenn Sie beim Verstümmeln von Copyright-Notizen perverse Lust verspüren sollten).

### **Patchen des Konfigurationsprogramms**

NZCOM.COM hat einen speziellen Patchbereich zur Programmkonfiguration in den fortgeschrittene Anwender Code einbinden kann, um besondere Abläufe auszuführen, zusätzlich zu denen von NZCOM. Dieser Patchbereich beginnt bei 0280H und erstreckt sich über 1KB bis zur Adresse 067FH. Unzweifelhaft werden sich viele kreative Anwendungen für diese Einrichtung finden.

Wir haben dabei an die Verwendung für spezielle Patches gedacht, wenn “above-BIOS” System-Erweiterungen, wie etwa DateStamper und/oder BYE, in einem NZ-COM-System laufen sollen. Bei diesen Programmen ist es nötig, in das laufende System Information zu extrahieren, wie sie “eingehängt” sind, um dann diese “Haken” zu restaurieren, nachdem ein neues System geladen wurde (aber noch bevor es zu laufen beginnt). Wir wollen nicht versuchen, Ihnen hier zu erklären, wie das gemacht wird; wir wollen versuchen, Ihnen genug Information über diesen Patchbereich zu geben, daß Sie mit der notwendigen Programmiererfahrung Gebrauch davon machen können.

NZCOM gibt während seines Laufs zu unterschiedlichen Zeitpunkten CALL-Anweisungen an den Code aus, der bei 0280H steht, indem es Z80-Register benutzt, um Information an die Konfigurationsroutine weiterzuleiten. Besonders der Wert im A-Register zeigt an, von welchem Punkt in der Sequenz der auszuführenden Anweisungen NZCOM den Call auslöste. Die folgenden Calls sind momentan definiert:

- A=0 NZCOM ist gerade gestartet und hat festgestellt, daß *kein* NZ-COM-System momentan läuft. Keine Dateien wurden zum gegenwärtigen Zeitpunkt geladen.
- A=1 NZCOM ist gerade gestartet und hat festgestellt, daß ein NZ-COM system momentan läuft. Keine Dateien wurden zum gegenwärtigen Zeitpunkt geladen.
- A=2 Alle zugewiesenen oder erforderlichen Default Moduln wurden in einen Arbeitspuffer geladen und sind bereit für das Laden an ihre Run-Time Adressen. Keine Modifikationen wurden bisher vom laufenden Betriebssystem gemacht.
- A=3 Die Moduln wurden wie benötigt aus dem Arbeitspuffer an ihre Run-Time Adressen kopiert, und NZCOM ist bereit, einen Kaltstart des neuen Systems auszuführen.
- A=0FFH NZCOM hat festgestellt, daß kein neues System geladen werden muß. Dieser Call tritt ein, wenn das angegebene neue System sich vom gegenwärtigen in keiner Weise derart unterscheidet, daß etwas nachgeladen werden müßte. Er wird ebenfalls eintreten, wenn ein Fehler auftritt, der den Abbruch einer NZCOM-Operation erfordert.

Die ersten zwei Calls stehen zur Verfügung für den Fall, daß Code im Konfigurations-Patch initialisiert werden muß, bevor andere Calls auftreten oder wenn andere Operationen der Konfigurations-Routine wissen müssen, welche Art System momentan aktiv ist.

Zusätzliche Information steht durch andere Z80-Register zur Verfügung. Das HL-Registerpaar zeigt auf den Arbeitspuffer WRT-BUF, wo NZCOM das neue System zusammenbaut. Der Environment Deskriptor für das neue System steht im Code-Block von 0100H bis 017FH. Die Information, die dort zusammen mit der Adresse des WRT-BUF steht, kann benutzt werden, um die Position jedes Moduls im Arbeitspuffer zu berechnen.

Das DE-Registerpaar zeigt auf einen String von Bytes mit Information über den Ladestatus. Gegenwärtig enthält der String zwei Bytes. Das erste heißt MODLST (Modul-Liste). Jedes seiner acht Bits wird dazu verwendet, anzuzeigen, ob Veränderungen in der System Kon-

0	CCP
1	DOS
2	BIOS
3	IOP
4	RCP
5	FCP
6	NDR
7	Z3T

Table 6.3: System-Segment, das zu jedem Bit im MODLST-Byte korrespondiert.

figuration das Laden eines korrespondierenden Moduls erforderlich machen. Die Funktion jedes Bits ist definiert in der Tabelle 6.3. Das zweite Byte heißt SEGLST (Segment-Liste). Jedes Bit indiziert, ob die Veränderung eines der in der Tabelle 6.4 verzeichneten System-Segmente den Einsatz des NZCOM-Laders erforderlich macht.

Außer wenn ein CALL mit dem Wert 3 im A-Register ausgeführt wird, bleibt das alte Betriebssystem an Ort und Stelle und funktional. Der Konfigurationscode kann sich daher auf Betriebssystem-Aufrufe stützen, um seine Aufgaben zu erledigen. Wenn er zum Beispiel entscheidet, daß die Ausführung des NZCOM-Befehls abgebrochen werden soll, kann er das tun durch einen Jump an die Adresse 0000H, um einen Warm Boot zu initiieren.

Man sollte sehr sorgfältig auf jeden Code achten, mit dem ein Call auf die Konfigurationsroutine mit dem Inhalt A=3 ausgeführt wird. An diesem Punkt wurde das neue System an seine Laufzeit-Adresse geladen, aber *es wurde nicht initialisiert und darf in keinster Weise verwendet werden.*<sup>12</sup>

---

<sup>12</sup>Mit Geschick, können Sie die Adresse des echten BIOS (im Gegensatz zum virtuellen BIOS) von NZ-COM und können entsprechende Calls darauf ausführen. Wenn NZCOM unter einem existierenden NZ-COM-System läuft, ist das Wort an der Adresse 0101H die Adresse der CONST-Routine des echten BIOS.

0	Shell Stack
1	Message Puffer
2	Externer FCB
3	Kommando Suchpfad
4	Wheel byte
5	Befehlszeilenpuffer-Adresse
6	reserviert
7	reserviert

Table 6.4: System-Segment, das zu jedem Bit im SEGLST-Byte korrespondiert.

## 6.5 Das Programm JetLDR

Im NZ-COM-Lieferumfang ist ein spezielles Programm namens JetLDR enthalten. Diese Programm ist ein extrem mächtiger Allzweck-Modullader. All die herkömmlichen Modul-Ladefunktionen für das NZ-COM-System können von NZCOM.COM ausgeführt werden. JetLDR jedoch, ist erweiterbar durch Verwendung spezieller Konfigurations- oder auch CFG-Dateien. Das sind Moduln, die JetLDR in sich selbst lädt und die die Art kontrollieren, wie es andere Moduln lädt. Diese Eigenschaft kann JetLDR die Fähigkeit geben, speziellen Moduln zu laden, wie etwa residente System-Erweiterungen (RSXe).

JetLDR hat eine eingebaute Hilfs-Seite, die mit der Standardeingabe standard command

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

aufgerufen werden kann.

## 6.6 Die MKZCM-Befehlszeile

Glücklicherweise muß über die MKZCM Befehlssyntax wenig gesagt werden, da sie so einfach ist. Das Format der Befehlszeile ist

```
MKZCM [name] <cr>
```



Das optionale Token “name” — dient als Name für die ZCM- und ENV-Deskriptor Dateien, die von MKZCM erzeugt werden sollen— kann auf der Befehlszeile hinzugefügt werden, wenn man im Voraus weiß, wie der neue System-Deskriptor heißen soll. Es liegt jedoch nur ein geringer Vorteil in dieser Option. Wird kein Name auf der Befehlszeile angegeben, wird MKZCM einen Prompt dafür ausgeben, wenn es aufgefordert wird, die System-Deskriptoren zu speichern. Entschließen Sie sich dazu, den Systemdefinitionsprozeß abzubrechen und nicht zu speichern, haben Sie sich eine Menge Tipperei erspart! Viel wichtiger aber ist, daß Ihre Optionen immer noch offen sind, wenn Sie sich mitten im Definitionprozess für einen anderen Namen entscheiden sollten. Haben Sie einen Namen auf der Kommandozeile angegeben, bleibt er Ihnen!

## 6.7 Funktions-Theorie

NZ-COM ist die spezielle Form einer Resident System Extension (RSX), die an eine hohe Speicheradresse des CP/M-BIOS geladen wird. Es beinhaltet folgende Moduln:

- Kommando Prozessor
- Disk Operating System
- Warm-Boot Umleitung
- Z-System-residente Segmente:
  - externe Environment Beschreibung
  - Terminal Capabilities Puffer
  - Message Puffer
  - Path
  - Wheel Byte
  - externer Datei-Kontroll-Block
  - multiple Befehlszeile
  - Shell Stack
  - externer Kommandoprozessor-Stack
  - optionale Z-System-Segment-Puffer
  - Named-Directory Register
  - Residentes Kommando Paket
  - Fluß Kommando Paket
  - Input/Output Paket

Der Code zum Verbiegen des Warmboots initialisiert das Diskssystem auf normale Art, lädt aber den Kommandoprozessor nicht aus den

Systemspuren nach, sondern aus einer Datei namens NZCOM.CCP. Wenn das System durch NZCPM.COM entfernt wurde, werden das ursprüngliche DOS-Modul und der ursprüngliche Warmboot-Vektor wieder restauriert. Das CP/M-Host-System übernimmt dann und lädt seinen Standard Kommando-Prozessor aus den Systemspuren.

Wird ein neues NZ-COM-System geladen, während ein anderes System bereits aktiv ist, speichert NZCOM den Status des gegenwärtigen Z-System-Environments, entscheidet, welche Änderungen erforderlich sind (z.B. ein Puffer mit veränderter Größe), reloziert und installiert die neuen Pakete und restauriert die unveränderten Komponenten des Environments.



# 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 39 mit einer Diskussion über den Umgang mit gecrunchten Dateien. Hier eine Liste dieser Dateien mit einer Kurzangabe 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.