

ZSDOS 1.1

Ein Ersatz des CP/M 2.2 BDOS

Programmer's Manual

© Copyright 1988, 89, 90

by

Harold F. Bower
Cameron W. Cotrill
Carson Wilson

Ein  Dokument

Inhaltsverzeichnis

1	Einleitung	1
1.1	Komponenten des Betriebssystems	1
1.2	Speicherbelegung	2
1.2.1	Speicherbereich der Segmente	2
1.2.2	Systempage	4
2	BDOS Funktionen	7
2.1	Zeicheneingabe/-ausgabe	7
2.2	Disketteneingabe/-ausgabe	8
2.3	Kontrolle und Status	8
3	ZSDOS Datenstrukturen	10
3.1	Allgemein	10
3.2	Logischer Record	10
3.3	Dateisteuerblock (FCB)	11
3.4	Directoryrecord.....	12
3.5	Diskettenbelegungsvektor	13
3.6	Diskettenparameterblock	13
3.7	Datumsangaben	14
3.8	Stempelformat	15
4	ZSDOS Programmierkonventionen	16
4.1	Allgemein	16
4.2	Wiedereintritt in ZSDOS Rufe	18
4.3	ZSDOS Konfigurationsbereich.....	19
4.3.1	Fehlervektortabelle	20
4.3.2	Pfadadresse (nur ZSDOS)	20
4.3.3	Adresse des Wheel Bytes	20
4.3.4	Konfigurationsbyte	21
4.3.5	Datumsvektoren	22
4.4	Routinen zur Unterstützung von Zeit- und Datumsstempeln	23
5	ZSDOS Funktionsrufe	24
5.1	Beschreibung der zurückgegebenen Werte	24
5.2	Funktionsbeschreibung.....	25
	Funktion 0 - Boot	25
	Funktion 1 - Konsoleneingabe	26
	Funktion 2 - Konsolenausgabe	27
	Funktion 3 - Zusatzeingabe [Leser]	28
	Funktion 4 - Zusatzausgabe [Stanzer]	29
	Funktion 5 - Listausgabe	30
	Funktion 6 - direkte Konsoleneingabe/-ausgabe	31
	Funktion 7 - IOBYTE holen	33
	Funktion 8 - IOBYTE setzen	34
	Funktion 9 - Zeichenkette ausgeben	35
	Funktion 10 - Konsolenpuffer lesen	36
	Funktion 11 - Konsolenstatus holen	38
	Funktion 12 - CP/M Versionsnummer holen	39
	Funktion 13 - Diskettensystem zurücksetzen	40
	Funktion 14 - Laufwerk auswählen	41
	Funktion 15 - Datei öffnen	42
	Funktion 16 - Datei schließen	44
	Funktion 17 - ersten Eintrag suchen	45
	Funktion 18 - nächsten Eintrag suchen	47
	Funktion 19 - Datei löschen	48
	Funktion 20 - sequentiell lesen	49

Funktion 21 - sequentiell schreiben	50
Funktion 22 - Datei erstellen.....	51
Funktion 23 - Datei umbenennen	52
Funktion 24 - Login-Vektor holen	53
Funktion 25 - aktuelles Standardlaufwerk holen	54
Funktion 26 - Adresse DMA Puffer setzen	55
Funktion 27 - Adresse des Belegungsvektors holen	56
Funktion 28 - Schreibschutz für Diskette setzen	57
Funktion 29 - Schreibschutzvektor holen	58
Funktion 30 - Dateiattribute setzen	60
Funktion 31 - Adresse des DPB holen	62
Funktion 32 - Nutzerbereich holen/setzen	63
Funktion 33 - wahlfrei lesen	64
Funktion 34 - wahlfrei schreiben	65
Funktion 35 - Dateigröße berechnen.....	66
Funktion 36 - wahlfreien Record setzen	67
Funktion 37 - mehrere Laufwerke zurücksetzen	68
Funktion 39 - Vektor fester Disketten holen	69
Funktion 40 - wahlfrei schreiben, auffüllen mit Nullen	70
Funktion 45 - BDOS Fehlermodus setzen	71
Funktion 47 - Adresse des DMA Puffers holen	74
Funktion 48 - Version des erweiterten DOS holen	75
Funktion 98 - Uhrzeit holen	76
Funktion 99 - Uhrzeit setzen.....	77
Funktion 100 - Konfigurationsflags holen	78
Funktion 101 - Konfigurationsflags setzen	79
Funktion 102 - Datumsstempel holen	80
Funktion 103 - Datumsstempel setzen	81

Schnellübersicht der Funktionen von ZSDOS

Übersicht der BDOS Fehlercodes

Kurzübersicht der BIOS Funktionen

1 Einleitung

ZSDOS ist ein mächtiger Ersatz für das Basic Disk Operating System (BDOS) von CP/M 2.2- oder ZRDOS 1.x-Systemen. Es verfügt über verschiedene neue und zahlreiche verbesserte Funktionen, wobei jedoch die Kompatibilität zu bestehenden CP/M 2.2 Programmen weitestgehend erhalten bleibt. Der Erhalt dieser Kompatibilität war auch das Hauptziel bei der Entwicklung von ZSDOS. In einigen Fällen erschienen uns die Möglichkeiten durch Einbindung von Elementen aus CP/M Plus (auch als CP/M 3 bekannt) und von ZRDOS jedoch wichtiger als der hundertprozentige Erhalt der Kompatibilität. Desweiteren wurden erweiterbare Datenstrukturen entworfen und in ZSDOS integriert, um die Basis für ein mächtigeres Betriebssystem zu schaffen.

Dieses Handbuch beschreibt ausführlich die Funktionen, Schnittstellen und Datenstrukturen von ZSDOS. Besonders die neuen Merkmale von ZSDOS stehen dabei im Vordergrund, aber auch die weniger bekannten oder häufig fehlverstandenen Eigenschaften von CP/M. Kurze Beispiele in Z80 Assemblersprache sollen den Gebrauch der ZSDOS Funktionsrufe verdeutlichen. In den Anhängen finden Sie Kurzübersichten der Funktionen von ZSDOS.

Dieses Programmierer's Manual wurde nicht als vollständige Dokumentation des CP/M-Betriebssystems oder der Z80 Assemblersprache konzipiert. Es wird eine gewisse Kenntnis der Konventionen bei der Benutzung von CP/M-Funktionsrufen vorausgesetzt. Für umfassendere Informationen über CP/M und die Z80 Assemblersprache empfehlen wir Ihnen die in der Bibliographie des *ZSDOS User's Guide* aufgelisteten Werke. Sofern nicht gesondert auf ein spezielles DOS hingewiesen wird, gelten die über ZSDOS gemachten Aussagen in diesem Handbuch ebenso für ZDDOS.

1.1 Komponenten des Betriebssystems

Das Betriebssystem CP/M 2.2 (bzw. kompatible) besteht aus drei separaten Segmenten; dem Basic Input/Output System (BIOS), dem Basic Disk Operating System (BDOS) und dem Console Command Processor (CCP). Jedes dieser Segmente ist relativ unabhängig von den anderen und kann recht einfach ersetzt werden. Voraussetzung dafür ist allerdings die Einhaltung festgelegter Schnittstellenparameter.

Für jeden Computer muß das BIOS in irgendeiner Form existieren. Es erledigt alle hardwarenahen Aufgaben. Alle angeschlossenen Geräte werden ebenso vom BIOS gesteuert wie die interne Hardware. Aufgrund der sehr unterschiedlichen Hardware verschiedener Computertypen ist auch das BIOS sehr verschieden. Im Normalfall wurde das BIOS vom Computerhersteller geschrieben. Es gibt aber auch einige recht verbreitete BIOSe von Drittanbietern für spezielle Computer. ZSDOS wurde so gestaltet, daß es auf nahezu jedem Computer läuft, dessen BIOS zu CP/M 2.2 oder ZRDOS 1.x kompatibel ist.

Das dritte Systemsegment, der CCP, stellt die primäre Schnittstelle zum Benutzer des Computers dar. Diese Komponente ist wahrscheinlich die am häufigsten ersetzte. Vor allem die ZCPR-Familie erfreut sich größter Beliebtheit. Da Veränderungen am CCP für den Benutzer am deutlichsten zu spüren sind, halten ihn viele für den wichtigsten Teil des Betriebssystems. In diesem Handbuch wollen wir jedoch zeigen, daß die Eigenschaften des BDOS darüber entscheiden, wie flexibel und leistungsfähig der Computer letztendlich wird.

Vom BDOS werden alle logischen Ein- und Ausgaben des Systems kontrolliert. Es verwaltet alle Ressourcen in Form von logischen Geräten, wie z. B. Konsole, Drucker und Diskettenlaufwerk. Durch das BDOS erhalten die reinen Hardwaretreiber des BIOS eine festgelegte Struktur, wodurch ein einheitliches Erscheinungsbild unterschiedlichster CP/M Computer gegenüber den Anwendungsprogrammen entsteht. Diese strikte Trennung hardwareunabhängiger Routinen von hardwareabhängigen war einer der bedeutendsten Fortschritte, die CP/M für die Entwicklung von Betriebssystemen für Mikrocomputer brachte.

ZSDOS ist ein vollständiger Ersatz des BDOS Segmentes und Gegenstand dieses Handbuches.

1.2 Speicherbelegung

Die Aufteilung des Arbeitsspeichers unter ZSDOS ist identisch zu CP/M 2.2- und ZRDOS 1.x-Systemen. Beginnend bei der absoluten Adresse 0 befinden sich die reservierten 256 Bytes (0 - 0FFH) der Systempage. Von Adresse 0100H bis zum unteren Ende des niedrigsten Systemsegments befindet sich der TPA-Bereich (Transient Program Area). Dort werden alle Anwendungsprogramme (z. B. Textverarbeitung, Datenbanken, Assembler usw.) ausgeführt. Die untere Adresse des niedrigsten Systemsegments ist durch ZSDOS nicht festgelegt.

1.2.1 Speicherbereich der Segmente

Als Systemsegmente werden alle Programm(teil)e bezeichnet, die nach einem Warmstart im System enthalten sind bzw. bleiben. Klassische Vertreter für derartige Segmente sind zunächst einmal BIOS und BDOS, die für Anwendungsprogramme unentbehrlich sind. Der CCP kann von Programmen überschrieben werden und wird jeweils bei der Rückkehr zum Betriebssystem nachgeladen.

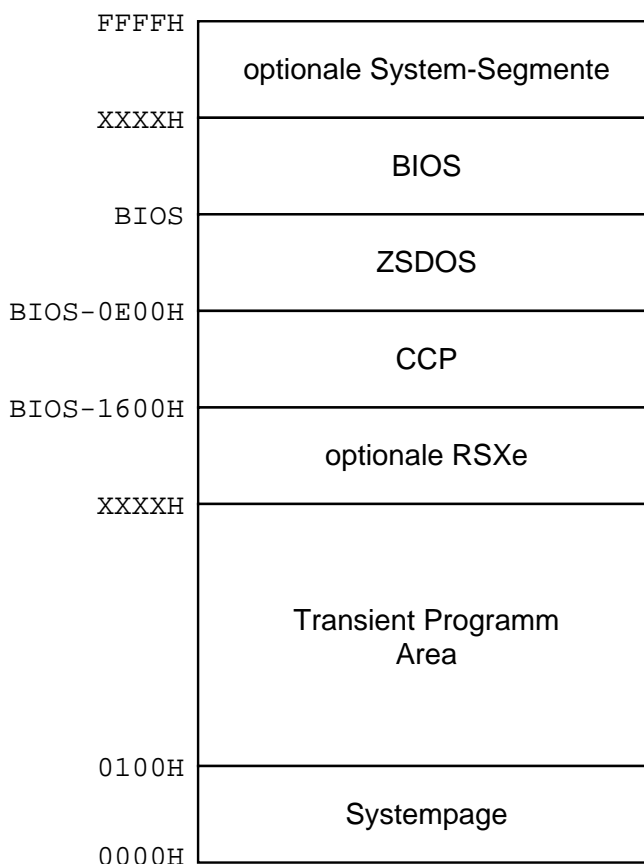
Einige CP/M 2.2 BIOSe laden nach einem Warmstart sowohl den CCP als auch das BDOS nach. Diese Eigenart ist historisch bedingt und geht auf eine Aussage von Digital Research zurück, wonach Anwendungsprogramme im Bedarfsfall auch das BDOS überschreiben können. Die Praxiserfahrung zeigt aber, daß kein (uns bekanntes) Programm das BDOS überschreibt und die meisten modernen BIOSe das BDOS nicht nachladen. Für einige Funktionen von ZSDOS ("Nur-Lesen-Vektor erhalten" und "schnelles Wiedereinloggen")

muß das BDOS resident vorhanden sein, anderenfalls haben sie keine Wirkung. (Konfigurationsdaten werden überschrieben)

Außer den eben erwähnten Segmenten können noch weitere Systemsegmente in einem ZSDOS System vorhanden sein. Im Gegensatz zu den bereits genannten sind diese zusätzlichen Segmente nicht zum Betrieb eines ZSDOS Systems erforderlich. Es können jedoch zusätzliche Funktionen durch derartige Segmente zur Verfügung gestellt werden.

Beispiele für diese Segmente, die man auch als Resident System Extension (RSX) bezeichnet, sind: BackGrounder ii und DosDisk, ZCPR3 Systemkomponenten (wie ENV, RCP, IOP, NDR und FCP), Utilities wie DateStamper und nicht zuletzt die ZSDOS-Erweiterungen zur Unterstützung von Datumsstempeln für Dateien.

RSXe befinden sich normalerweise direkt unterhalb des CCP im Speicher. Alle anderen Systemerweiterungen sind im Normalfall oberhalb des BIOS angesiedelt. Zusammenfassend ist die Speicherbelegung in folgender Übersicht dargestellt:



Wie dieser Darstellung zu entnehmen ist, sind nur die Adressen der Systempage und der Beginn des TPA-Bereiches in einem ZSDOS System genau festgelegt. Alle anderen Systemadressen sind abhängig vom BIOS. Die Größen der

Systempage, des ZSDOS und des CCP (außer in erweiterten Z-Systemen) sind genau definiert. Andere Systemsegmente können den eigenen Bedürfnissen angepaßt sein.

1.2.2 Systempage

Die Systempage (Adreßbereich von 0 - 0FFH) wird von ZSDOS für wichtige Systeminformationen genutzt. Für ZSDOS gelten dieselben Festlegungen für die Systempage wie unter CP/M 2.2. Die Systempage bildet die Schnittstelle zu ZSDOS, weshalb das Verstehen der Funktion jedes einzelnen Bereiches äußerst wichtig ist.

00H - 02H Sprung zur BIOS-Warmstartroutine (BIOS+03H)

Diese Adresse darf von keinem Programm verändert werden. Sie bietet die einzige Möglichkeit, die Basisadresse des ZSDOS Systemsegmentes mit Sicherheit festzustellen. Ein Beispiel für die "richtige" Benutzung des Warmboot Vektors folgt später in diesem Handbuch. Wenn Programme die BIOS Sprungvektoren verändern wollen, dann dürfen nur die Werte der BIOS Sprungtabelle angepaßt werden, nicht das Sprungziel der Adresse 0.

Einzig Alpha System's NZCOM verändert den BIOS Warmboot Vektor in akzeptabler Weise. Es wird ein BIOS einschließlich Sprungtabelle "imitiert", das sich weiter unten im Speicher befindet. Die Systemsegmente des ZCPR werden zwischen dem echten System-BIOS und dem "Imitat" von NZCOM geladen. Der BIOS Warmboot Vektor zeigt auf das "imitierte" BIOS. Dadurch laufen alle Programme, bis auf systemspezifische Utilities, weiterhin fehlerfrei. Wenn man ein BIOS schreibt oder überarbeitet, sollte man bei den Systemutilities die Funktionsweise von NZCOM beachten.

03H IOBYTE

Das IOBYTE enthält eine BIOS-abhängige Struktur. Es kann vom Programmierer des BIOS verwendet werden, um eine Umlenkung byteorientierter Ein-/Ausgaben zu ermöglichen. Im Byte selbst sind 4 Felder enthalten, die für die logischen Geräte Konsole (CON:), Leser (RDR:), Stanzer (PUN:) und Listgerät (LST:) stehen. Jedes logische Gerät kann einem von bis zu vier verschiedenen physikalischen Geräten zugeordnet werden.

Weil die Einbindung des IOBYTE optional und systemabhängig ist, schlagen Sie bitte in den Handbüchern Ihres Computers nach, um die genauen Festlegungen für Ihr System zu ermitteln.

04H aktuelles CCP Standardlaufwerk und -nutzerbereich

Im Byte dieser Speicherstelle legt der CCP die Werte für das aktuelle Standardlaufwerk und den aktuellen Nutzerbereich ab. Der Wert für das Laufwerk wird in den Bits 0 bis 3 gespeichert, beginnend bei 0 für das Laufwerk A. In den Bits 4 bis 7 wird der Nutzerbereich mit Modulo 16 gespeichert. Dabei ist zu beachten, daß nur auf die Nutzerbereiche 0 bis 15 direkt zugegriffen wer-

den kann. Da im Directory für den Nutzerbereich fünf Bits zur Verfügung stehen, können Dateien bzw. Programme auch in den Nutzerbereichen 16 bis 31 gespeichert werden. Ab Version 3.3 des ZCPR ist das Einloggen in die Nutzerbereiche 16 bis 31 auf Kommandoebene optional möglich. Durch den zusätzlichen CCP Code bleibt (bis auf wenige Ausnahmefälle) die Nummer des hohen Nutzerbereiches erhalten.

05H - 07H Sprung zum BDOS

Ein Ruf auf die Adresse 5 wird verwendet, um eine Funktion von ZSDOS auszuführen. Der Wert auf Adresse 6 kann jedoch nicht als direkter Zeiger auf ZSDOS verwendet werden!

Wird von einem Programm die Größe des verfügbaren TPA-Bereiches benötigt, dann kann die auf den Adressen 6 und 7 gespeicherte Sprungadresse zur Berechnung verwendet werden. Das höherwertige Byte des Wertes auf Adresse 6 zeigt stets auf die letzte Page des TPA-Bereiches. Je nachdem, ob eine RSX geladen ist oder nicht, befindet sich auf der nächsten Page das ZSDOS Segment.

08H - 2FH frei für Systemerweiterungen

30H - 37H reserviert

38H - 3FH Restart-Vektor 38H

Im Normalfall wird diese Adresse von Debuggern für die Speicherung der Breakpoint-Routine benutzt. Während der Testphase wird vom Debugger somit nur ein Byte (Opcode: RST 38H, Wert: 0FFH) für den Eintrag des Breakpoints benötigt.

40H - 4FH frei für Systemerweiterungen

Auf einigen Computern werden Teile dieses Speicherbereiches für Systemfunktionen benutzt. So wird z. B. beim Ampro Little Board in den Bytes 40H bis 4DH der ZCPR3 Pfad abgelegt.

50H - 5BH frei für Programme

Verschiedene Modem/BBS Programme aus dem Public Domain Bereich speichern in diesem Bereich Parameter wie z. B. die Baudrate ab.

5CH - 6BH standardmäßiger Dateisteuerblock (FCB) 1

6CH - 7BH standardmäßiger Dateisteuerblock (FCB) 2

In den Dateisteuerblöcken werden durch den CCP die ersten beiden Parameter der Kommandozeile abgelegt. Bei Benutzung des ersten Dateisteuerblocks zum Öffnen einer Datei wird der Inhalt des zweiten Dateisteuerblocks überschrieben. Beim Öffnen einer Datei mit dem zweiten Dateisteuerblock wird ein Teil des DMA-Puffers überschrieben. Die 16 Bytes des zweiten Dateisteuerblocks sollten vor der Benutzung von 6CH auf einen anderen Speicherbe-

reich kopiert und erst dort verwendet werden. Somit kann der Dateisteuerblock auf Adresse 5CH unverändert benutzt werden. Nach dem Öffnen der Datei umfaßt der vollständige Dateisteuerblock 36 Bytes (von 5CH bis 7FH).

80H - 0FFH standardmäßiger DMA Puffer / Kommandozeile

Dieser Speicherbereich wird für zwei Aufgaben verwendet. Wenn ein Programm durch den CCP gestartet wird, legt dieser hier alle Argumente der Kommandozeile zur weiteren Verwendung durch das Programm ab. Das erste Byte des Puffers (80H) enthält die Anzahl der gültigen Zeichen. Anschließend folgt der Rest der Kommandozeile selbst, beginnend mit dem Leerzeichen (auf 81H), welches die Argumente vom Kommando trennt.

Außerdem wird dieser Bereich für den standardmäßigen DMA Puffer benutzt. Solange die Adresse des DMA Puffers nicht über die BDOS Funktion 26 geändert wurde, erfolgen alle Datenübertragungen zwischen Speicher und Diskette über diesen Speicherbereich. Die Kommandoparameter müssen vom Programm also ausgewertet werden, bevor der Puffer durch Diskettenoperationen überschrieben wird.