

PDRIVE .SYS

In der Datei PDRIVE.SYS sind die Pdriveparameter zwischengespeichert. Ein Pdrivedatensatz besteht immer aus 128 Bytes. Er besitzt also genau die Länge eines "CP/M Sektors". In jedem dieser Datensätze sind alle Informationen enthalten, um das jeweilige Laufwerk hardwaremäßig zu beschreiben. Zusätzlich enthält ein solcher Datensatz alles über die logische Organisation einer bestimmten Diskette, die dann auf dem oben hardwaremäßig beschriebenen Laufwerk betrieben werden soll.

Nun zum Aufbau eines solchen Datensatzes. Die Bytes zwei, drei und vier eines jeden Datensatzes sind unbenutzt. Das erste und das fünfte Byte sind BIOS Bytes. Hier wird das jeweilige Laufwerk hardwaremäßig beschrieben.

Nun folgt ein 15 Bytes langer String, der den Namen des Datensatzes enthält. Unter diesem Namen finden die verschiedenen Programme einen bestimmten Datensatz aus der Masse der anderen heraus. Hier liegt auch der große Vorteil der Pdrivedatensätze: Hat man sich einmal der Mühe unterzogen einen Datensatz für eine bestimmte Diskette zu entwerfen und auszutesten, kann man diesen Datensatz immer wieder unter seinem Namen wiederfinden. Dieser Name muß mindestens ein Zeichen lang sein. Er darf alle ASCII-Zeichen und auch Leerzeichen enthalten. Ein Name, der kürzer ist als 15 Bytes, muß nach rechts mit Leerzeichen aufgefüllt werden. Dies erledigt das Programm PDRIVE.MAS automatisch.

Das nächste Byte (Byte 21 dez) muß immer das Zeichen \$ (24 hex 36 dez) sein. An diesem Byte erkennen die verschiedenen Programme, ob sie einen korrekten Datensatz bearbeiten, oder nicht.

Nun folgen 15 Bytes, die die logische Organisation einer Diskette wiedergeben. Diese 15 Bytes werden nur vom BDOS benutzt, nicht vom BIOS. Die Beschreibung dieser Bytes folgt unter PDRIVE.MAS in diesem Handbuch.

Die nächsten 92 Bytes sind für die Sektortabelle bestimmt. Es brauchen nicht alle 92 Bytes belegt zu sein, was eigentlich auch nie vorkommt. Trotzdem sind alle 92 Bytes für die Sektortabelle reserviert.

Die Datei PDRIVE.SYS ist demnach nur eine lose, ungeordnete Aneinanderreihung solcher 128 Bytes langer Datensätze. Die Anzahl der verschiedenen Datensätze bestimmt die Länge der Datei PDRIVE.SYS.

Die Bedeutung der einzelnen Bytes eines Pdrivedatensatzes in Tabellenform:

00 hex - 00 hex	BIOS Byte (erster Teil)
01 hex - 03 hex	unbenutzt
04 hex - 04 hex	BIOS Byte (zweiter Teil)
05 hex - 13 hex	Name des Datensatzes
14 hex - 14 hex	\$ (24 hex 36 dez)
15 hex - 23 hex	logische Organisation der Diskette
24 hex - 7F hex	Sekortabelle der Diskette

In der von uns gelieferten Datei PDRIVE.SYS sind folgende Datensätze enthalten:

0: S40	1: S40 DS
2: S40 DD	3: S40 DSDD
4: S40 DT	5: S40 DSDT
6: S40 DDDT	7: S40 DSDDDT
8: S80	9: S80 DS
10: S80 DD	11: S80 DSDD
12: 40	13: 40 DS
14: 40 DD	15: 40 DSDD
16: 40 DT	17: 40 DSDT
18: 40 DDDT	19: 40 DSDDDT
20: 80	21: 80 DS
22: 80 DD	23: 80 DSDD
24: FORMULA	25: FORMULA DT
26: OSBORNE	27: OSBORNE DT
28: OSBORNE DD	29: OSBORNE DDDT
30: GENIE III 40	31: GENIE III 40 DS
32: GENIE III	33: GENIE III DS
34: IBM 3740	35: OMIKRON
36: ALPHATRONIC	37: 80 DSDDSP

Die Tabelle der Datensätze enthält nur deren Namen, also jeweils die Bytes 05 hex bis 13 hex eines jeden Datensatzes. Diese Namen können, wie oben schon gesagt, beliebig gewählt werden. Trotzdem folgen sie einem Regelsystem, welches es Ihnen erlaubt, den jeweils richtigen Datensatz auszuwählen, ohne immer lange in der oben abgebildeten oder sonstigen Tabellen nachschauen zu müssen.

Die ersten 24 Datensätze (0 bis 23) sind extra für das Genie I/II CP/M entwickelt worden. Sie liefern maximalen Speicherplatz in Verbindung mit optimalen Zugriffszeiten. Das Regelsystem der Datensatznamen bezieht sich vorwiegend auf diese 24 Datensätze.

Das "S" am Anfang eines solchen Namens bedeutet soviel wie System. Das heißt, diese Datensätze sind für Disketten bestimmt, die den drei Spuren langen Booter des Betriebssystems auf den Spuren 0 bis 2 tragen. Das Directory solcher Disketten liegt also auf Spur 3. Die Datensätze ohne "S" am Anfang des Namens sind für reine Datendisketten bestimmt. Auf solchen Disketten liegt das Directory auf Spur 0. Sie können nicht in Laufwerk A betrieben werden.

Nach dem "S" kommt entweder eine "40" oder eine "80" im Datensatznamen. Sie stehen für mit 40 Track (Spuren) oder 80 Track (Spuren) beschriebene Disketten. Also alle Datensätze mit einer "40" im Namen sind für 40 Track Disketten bestimmt, die mit einer "80" für 80 Track Disketten.

Als letztes gibt es die Namenszusätze "DS", "DD" und "DT". Sie signalisieren, daß der Datensatz mit

- "DS" für Double Side formatierte Disketten bestimmt ist,
- "DD" für Double Density formatierte Disketten bestimmt ist und
- "DT" für Double Track betriebene Disketten, das heißt 40 Track Disketten in 80 Track Laufwerken, bestimmt ist.

Treten zwei oder drei dieser Namenszusätze gleichzeitig auf, so gilt die Reihenfolge "DS" vor "DD" vor "DT". Nach dieser grauen Theorie nun drei Beispiele:

- a) Sie haben eine Diskette von der Sie wissen, daß sie ein CP/M System enthält, also muß der richtige Datensatz mit "S" anfangen. Zusätzlich ist es eine 40 Track Diskette, demnach muß der Namen ein "40" enthalten. Und als letztes wurde die Diskette mit Double Density beschrieben, was ein "DD" notwendig werden läßt. Nun brauchen Sie nur noch die einzelnen Namensteile zusammenzustellen und es ergibt sich S40 DD. Sie brauchen also nur noch das Laufwerk, auf dem Sie die Diskette betreiben wollen, mit dem Datensatz, dessen Namen S40 DD heißt, zu programmieren. Dazu siehe unter PD.COM in diesem Handbuch.
- b) Zweites Beispiel: Sie wollen eine 40 Track Diskette mit CP/M System, die doppeltseitig mit Double Density beschrieben wurde, betreiben. Als zusätzliche Schwierigkeit nehmen wir an, Sie besäßen "nur" 80 Track Laufwerke. Als erstes gehört wieder ein "S" wie System zum gesuchten Namen. Als zweites benötigen wir eine "40" wie 40 Track. Ein "DS" wie doppeltseitig und ein "DD" wie Double Density gehören auch dazu. Und zuletzt noch das "DT" wie doppelt Track, weil Sie eine 40 Track Diskette auf ein 80 Track Laufwerk betreiben wollen. Nun hängen Sie wieder alles zusammen und es ergibt sich: S40 DSDDDT. Unter diesem Namen sind dann die Pdriveparameter abgespeichert, die Sie suchen.
- c) Als letztes Beispiel wähle ich eine 80 Track Datendiskette, die einseitig mit Single Density beschrieben wurde. Zu bemerken ist hier, daß Sie zum Lesen einer solchen Diskette auf jeden Fall ein 80 Track Laufwerk benötigen. Nun zum Datensatznamen; ein "S" darf nicht enthalten sein, da es eine reine Datendiskette ist. Ebenfalls treffen hier die Namenszusätze "DS", "DD" und "DT" nicht zu. Der gesuchte Datensatzname heißt in diesem Fall schlicht 80.

Die oben aufgeführten Beispiele beziehen sich nur auf Disketten, die mit den speziell entwickelten Genie I/II Formaten beschrieben sind, also mit dem Programm FORMAT.COM der Systemdiskette. Für die Datensatznamen zu systemfremden Formaten, also Disketten anderer Computer gilt, daß nur solche Namenszusätze frei wählbar sind, die auch auf den entsprechenden Computern frei wählbar sind. Beispiele: Im Datensatz "FORMULA" ist bereits enthalten, daß die entsprechenden Disketten doppelseitig unter Double Density beschrieben sind. Als frei wählbarer Namenszusatz bleibt nur noch "DT" übrig. Bei den Formaten "OSBORNE" ist zusätzlich der Namenszusatz "DD" zulässig, da der Osborne Computer Disketten unter Single Density wie auch unter Double Density beschreibt. Beim Format "GENIE III" dagegen ist nur der Namenszusatz "DD" unerlaubt, da das Genie III unter CP/M immer nur Double Density fährt, also alle GENIE III Datensätze grundsätzlich Double Density sind.

Natürlich können nicht immer alle Formate benutzt werden. Besitzen Sie zum Beispiel ausschließlich einseitig schreibende Laufwerke, so können Sie keine Formate benutzen, die den Namenszusatz "DS" enthalten. Zusätzlich fallen dann die FORMULA Formate aus, da dieser Computer die Disketten immer doppelseitig beschreibt. Entsprechendes gilt für "DD" und "DT".

Wie Sie sicher schon bemerkt haben, können Sie mit diesem BIOS unter GENIE I/II auch Disketten fremder Computersysteme lesen und schreiben. Sie brauchen nur einen richtig erstellten Pdrivedatensatz. Mitgeliefert bekommen Sie die ausgetesteten Datensätze zum Betreiben von OSBORNE, FORMULA und GENIE III Disketten. Sie können quasi alle CP/M Diskettenformate mit diesem BIOS Ihrem Computer anpassen. Haben Sie beim Programmieren eines bestimmten Datensatzes Probleme, setzen Sie sich mit Ihrem Fachhändler in Verbindung. Zum Anpassen eines bestimmten Diskettenformates benötigen wir folgende Daten und Sachen:

- 1) Eine Diskette, des anzupassenden Computersystems; diese Diskette muß ein CP/M System beinhalten und auf dem anzupassenden Computer bootfähig sein. Auch sollten die Programme STAT.COM und PIP.COM auf der Diskette abgespeichert sein.
- 2) Zusätzlich benötigen wir den Ausdruck (abschreiben tut's auch) der Anweisung "STAT DSK:", der auf dem anzupassenden Computer angefertigt werden muß.
- 3) Es reicht nicht aus, wenn Sie uns sagen, daß Sie gerne einen Pdrivedatensatz zum Betreiben einer Diskette des CP/M Computers XYZ ungelöst hätten. Wir können nicht die Systemdisketten aller CP/M Computer auf Lager haben.

Mit diesen Daten sind wir normalerweise in der Lage einen Pdrivedatensatz für die entsprechende Diskette zu erstellen.

PDRIVE .MAS

Mit dem Programm PDRIVE.MAS sind Sie in der Lage, ein File mit Pdrivedatensätzen, also zum Beispiel die Datei PDRIVE.SYS, zu editieren oder neu zu erstellen. Mit dem Funktionstastensatz, der auf der Systemdiskette geliefert wird, können Sie das Programm mit fkt "Q" anstarten. Der Funktionsablauf ist analog zu fkt "F" beim Anstarten von FUNKTION.MAS. Das Programm PDRIVE.MAS meldet sich dann als erstes mit:

Wollen Sie einen alten File bearbeiten ?

Hier können Sie analog zu FUNKTION.MAS wieder entweder J (wie ja) oder N (wie nein) antworten. Bei nein wird im Computer eine Leerdatei erzeugt. Bei ja fragt Sie der Computer mit

Alter Filename ?

nach dem Filenamen einer Pdrivedatei. Hier können Sie demnach mit PDRIVE.SYS antworten. Der Inhalt des kompletten Pdrive-datensatzes wird in den Computer eingeladen. Diese Datei darf maximal 100 Datensätze und damit 100 Records lang sein. Mehr kann das Programm PDRIVE.MAS (gleichzeitig) nicht verarbeiten.

Auf jeden Fall meldet sich das Programm PDRIVE.MAS als nächstes im Grundmodus mit:

Befehlseingabe ?

Hier stehen Ihnen sechs Befehle zur Verfügung. Ausgeführt werden diese Befehle durch Eingeben eines Buchstabens (siehe unten) und Drücken der <NEW LINE> Taste. Das Programm führt dann im Dialogstil durch die jeweilige Funktion.

- 1) P Mit diesem Befehl können Sie sich die Datensatznamen der einzelnen Pdrivedatensätze auf dem Bildschirm listen lassen. Da die Gesamtdatei mehr Datensätze enthalten kann, als gleichzeitig auf dem Bildschirm passen, werden sie immer nur in 12er Schritten gelistet. Nach jedem 12er Block müssen Sie einmal <NEW LINE> drücken, um sich durch die Datensätze durchzutasten. Die einzelnen Datensatznamen sind dabei von 0 an aufwärts durchnummeriert.
- 2) L Mit diesem Befehl können Sie Datensätze löschen. Dazu geben Sie nur die Nummer des betreffenden Datensatzes ein. Dieser wird dann gelöscht. Wenn Sie sich danach die Namen der Datensätze noch einmal mit "P" listen lassen, werden Sie feststellen, daß der Name des Datensatzes zwar fehlt, dessen frühere Nummer jedoch noch auftaucht. Sollten Sie versehentlich einen Datensatz löschen, können Sie ihn trotzdem noch mit "E" editieren und seinen Namen neu eingeben. Damit ist der Vorgang des Löschens rückgängig gemacht.

- 3) K Mit diesem Befehl können Sie Datensätze kopieren. Dazu geben Sie lediglich die Nummer des betreffenden Datensatzes ein. Das Programm erzeugt dann einen zusätzlichen Datensatzplatz am Ende der Datei. In diesen wird dann der Datensatz kopiert. Diese Operation führt der Computer vorläufig nur im Arbeitsspeicher aus. Dieser Befehl ist dann sehr nützlich, wenn Sie den ursprünglichen Datensatz, leicht modifiziert, zusätzlich benötigen. Er erspart Ihnen damit das neue Eingeben aller Parameter eines Datensatzes. Vergessen Sie nicht, auch den Namen des Datensatzes zu ändern! Denn auf Grund des Namens findet das Programm PD.COM einen Datensatz innerhalb der Gesamtdatei.
- 4) A Mit diesem Befehl können Sie eine zweite Pdrive-datendatei in den Arbeitsspeicher einladen. Damit koppeln Sie also zwei verschiedene Dateien zu einer. Die Datensätze werden am Ende angehängt und entsprechend numeriert. Achten Sie aber darauf, daß die Anzahl der einzelnen Datensätze die Zahl 100 nicht überschreitet.
- 5) S Mit dem Befehl S verlassen Sie den Grundmodus endgültig. Das Programm fragt Sie jetzt mit:

Einen anderen Filenamen (J/N)?

ob Sie einen neuen Filenamen benutzen wollen, um die Pdrivedatensätze, die sich momentan im Speicher befinden, auf der Diskette abzugelegen. Wenn Sie beim Anstarten des Programms keinen alten Datensatz eingeladen haben, können Sie diese Frage nur mit einem "J" beantworten. Das System fragt Sie dann mit:

Neuer Filenamen?

Nach einem Filenamen, unter dem die neuen Datensätze auf der Diskette verankert werden sollen. Existiert ein entsprechendes File noch nicht, so wird es vom Programm eröffnet. Nach dem Abspeichern der Datensätze springt das Programm wieder zum jeweils aktiven CCP zurück.

- 6) E Mit diesem Befehl können Sie einen beliebigen Pdrivedatensatz, unter Angabe seiner Nummer, editieren. Haben Sie beim Anstarten des Programms keine alte Pdrivedatei eingeladen, bleibt der Bildschirm beim P Befehl natürlich leer, Sie können trotzdem jeden noch nicht existierenden "Datensatz" editieren und damit einen neuen programmieren. Das gilt auch, wenn die Datensätze 0 bis 30 schon belegt sind. Geben Sie beim Editieren einfach die Nummer 31 ein und programmieren Sie einen neuen Datensatz. In beiden Fällen führt Sie das Programm durch insgesamt vier Bildschirmseiten, auf denen Sie Parameter editieren oder programmieren können.

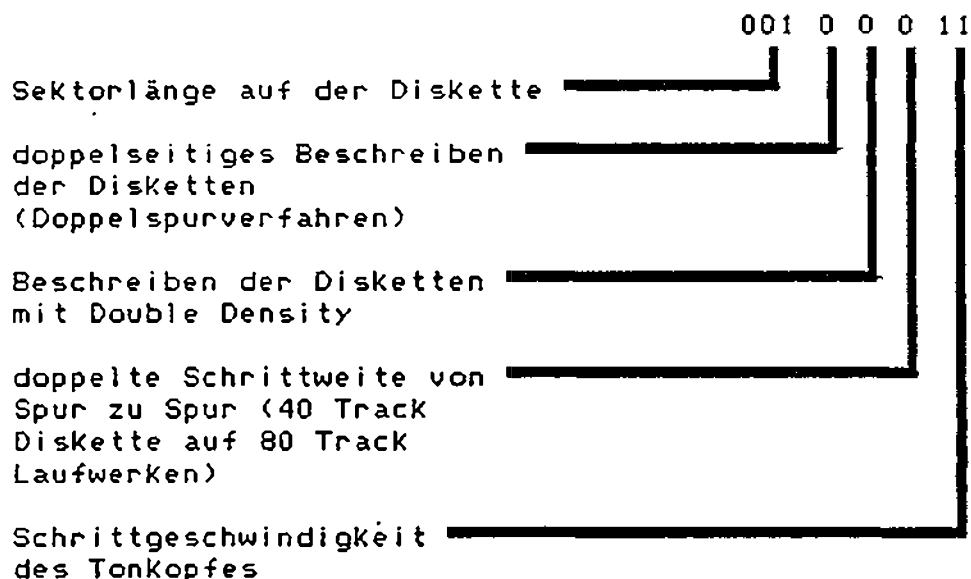
Editieren eines DatensatzesErste Bildschirmseite

über jeder der vier Bildschirmseiten steht der Name des betreffenden Datensatzes. Man weiß also immer, wobei man gerade ist. Auf der ersten Bildschirmseite werden die BIOS-Bytes programmiert. Als erstes wird gefragt, ob der Datensatz für ein 8 Zoll Laufwerk bestimmt ist:

8 Zoll Laufwerk:Dauerläufer :

Die Frage nach dem "Dauerläufer" bezieht sich auf den Antrieb des entsprechenden Laufwerkes. Läuft der Antriebsmotor immer, so sollten Sie hier "J" eingeben, denn es ergibt sich ein erheblicher Zeitvorteil. Das ist bei (fast) allen 8 Zoll Laufwerken der Fall. Wenn Sie die Einstellung dieser Parameter nicht ändern wollen, geben Sie einfach <NEW LINE> ein.

Der zweite Teil der BIOS Bytes besteht aus insgesamt 8 Einsen und Nullen. Die Bedeutung der einzelnen Bits wird am Beispiel des "Osborne" Datensatzes gezeigt:



Die hardwaremäßige Sektorlänge auf der Diskette stimmt bei Microcomputer CP/M Systemen in den seltensten Fällen mit der Länge eines "CP/M Sektors" überein. Ein "CP/M Sektor" ist bekanntlich immer 128 Bytes lang. Sehr oft werden dann 2, 4 oder 8 solcher 128 Bytes langen Sektoren zusammengefaßt und gleichzeitig in einem Sektor der Diskette untergebracht. Dies ist keine Funktion des eigentlichen CP/M's. Dafür haben die "Sektoren" immer eine Länge von 128 Bytes. Das ist im CP/M intern so festgelegt. Das Blocking und Deblocking erzeugt das jeweilige BIOS. Es sammelt dann beim Schreiben von Sektoren immer 2, 4 oder 8 "CP/M Sektoren" in einem Buffer des BIOS und führt erst dann eine Schreiboperation aus. Umgekehrt liest dann das BIOS einen langen Hardsektor in seinen Buffer ein und gibt ihn dann in 128 Bytes umfassenden Häppchen ans BDOS weiter.

Im BIOS 2.1 können Sie die hardwaremäßige Länge der Sektoren mit den linken drei Bits des BIOS-Byte, wie folgt, selbst bestimmen.

000	128 Bytes pro Hardsektor
001	256 Bytes pro Hardsektor
011	512 Bytes pro Hardsektor
111	1024 Bytes pro Hardsektor

Das vierte Bit von links des BIOS Byte gibt an, ob die Rückseite der Diskette mit dem Doppelspurverfahren angesprochen werden soll, oder nicht.

Das Doppelspurverfahren:

Bei diesem Verfahren organisiert das BIOS die Diskette so, daß sie für das CP/M aussieht, als wäre sie einseitig beschrieben und besäße doppelt so viele Spuren. Dabei liegen dann alle Spuren mit geraden Nummern (0,2,4,6...) auf der Vorderseite der Diskette und alle Spuren mit ungeraden Nummern (1,3,5,7...) auf der Rückseite der Diskette. Dieses Verfahren wird zum Beispiel beim FORMULA-Microcomputer angewendet. Bei diesem Format sind auch drei Spuren zu Systemspuren ernannt und damit dem CP/M entzogen, wie bei fast allen Microcomputern mit CP/M Betriebssystem. Beachten Sie dabei bitte, daß die Vorder- und Rückseite der ersten Spur und die Vorderseite der zweiten Spur je einzeln gezählt werden müssen. Dadurch beginnt das Directory bereits auf der Rückseite der zweiten und nicht auf der Vorderseite der vierten Spur.

Das Doppelsektorverfahren:

Wenn Sie Disketten mit diesem Verfahren ansteuern, dürfen Sie das vierte Bit von links des BIOS Byte nicht 1 setzen, obwohl Sie die Diskette doppelseitig beschreiben wollen. Die entsprechenden Parameter werden in der Sektortabelle gesetzt. Näheres dazu folgt. Für das CP/M sieht dann eine Diskette aus, als hätte sie pro Spur die doppelte Anzahl von Sektoren als eine gleiche einseitig beschriebene Diskette. Dieses Verfahren wird zum Beispiel beim GENIE III CP/M angewendet, um die Rückseite einer Diskette anzusprechen. Auch hier sind drei Spuren zu Systemspuren ernannt. Dabei belegen sie beim Doppelsektorverfahren wirklich die ersten drei Spuren und nicht eineinhalb wie beim FORMULA. Das Directory liegt dann wie erwartet auf der vierten Spur.

Das fünfte Bit von links des BIOS-Byte gibt an, ob die Diskette mit doppelter Schreibdichte beschrieben werden soll.

Das sechste Bit von links des BIOS-Byte muß gesetzt werden, wenn eine 40 Track Diskette auf einem 80 Track Laufwerk betrieben werden soll. Das BIOS läßt dann den Kopf des Laufwerks von Spur A bis Spur B einfach doppelt so viele Schritte ausführen wie sonst. Hier sei bemerkt, daß das Schreiben von 40 Track Disketten auf 80 Track Laufwerken in der Regel nicht einwandfrei funktioniert. Die betreffenden Stellen können zwar auf dem 80 Track Laufwerk wieder gelesen werden, meistens jedoch nicht mehr auf 40 Track Laufwerken. Sie sollten diese Möglichkeit am besten nur benutzen, um Programme und Daten in den Computer einzulesen.

Die letzten beiden Bits des BIOS Byte, Bit sieben und acht von links, geben die Schrittgeschwindigkeiten an, mit der das Floppyinterface das Laufwerk steuern sollen. Sie bedeuten dann folgendes:

00	5 ms von Spur zu Spur
01	10 ms von Spur zu Spur
10	20 ms von Spur zu Spur
11	40 ms von Spur zu Spur

Wenn Sie das BIOS-Byte nicht ändern wollen, drücken Sie einfach <NEW LINE>. Damit kommen Sie automatisch zur nächsten Bildschirmseite. Wenn Sie jedoch eines der Bits ändern wollen, müssen Sie immer alle acht eingeben, auch solche, die Sie nicht ändern wollen.

Zweite Bildschirmseite

Auf der zweiten Bildschirmseite erfolgt die Programmierung der logischen Organisation der Diskette. Diese Parameter werden nur vom BDOS benutzt. Es sind 10 verschiedene Parameter, die insgesamt 15 Bytes eines Pdrivedatensatzes belegen. Da diese Parameter miteinander zusammenhängen, reicht die Eingabe von fünf aus. Die restlichen werden vom Programm PDRIVE.MAS berechnet und ergänzt. Nun die fünf notwendigen Parameter:

```
Sektoren pro Spur      :
Datenblocklänge       :
Kapazität der Diskette :
Zahl der Directory Blöcke:
Zahl der Systemspuren  :
```

Beim Übergang auf diese Bildschirmseite werden die aktuellen Inhalte dieser Parameter angezeigt. Programmieren Sie gerade einen neuen Datensatz, so werden zufällige Inhalte angezeigt, die aber keinen Sinn ergeben.

Jetzt können Sie hinter jeden alten Wert einen neuen eingeben. Wollen Sie einen der Parameter nicht ändern, so geben Sie nur <NEW LINE> ein. Wenn Sie einen der Parameter mit 0 programmieren wollen, so müssen Sie -1 eingeben. Null wird als "keine Änderung" erkannt. Die -1 rechnet das Programm zu 0 um.

Sektoren pro Spur Hier müssen Sie die Zahl der "CP/M Sektoren" eingeben, die auf eine Spur passen. Wenn Sie eine Diskette doppelseitig mit dem Doppelsektorverfahren betreiben wollen, müssen Sie die Sektoren der Vorder- und Rückseite addieren und hier angeben. Diese Zahl ist identisch mit dem Parameter "Sektors/Track" des Programmausdrucks "STAT DSK:".

Datenblocklänge Hier folgt die Anzahl der Bytes, die mindestens durch einen Directoryeintrag belegt werden müssen. Die Bedeutung kommt der der Granule beim Newdos gleich. Eingeben dürfen Sie hier nur 1024, 2048, 4096, 8192 und 16384.

Kapazität der Diskette Hier muß die Anzahl von "CP/M Sektoren" angegeben werden, die auf die Diskette passen. Hat die Diskette Systemspuren, die nicht vom CP/M benutzt werden dürfen, müssen Sie deren Kapazität abziehen. Dieser Wert ist identisch mit dem Wert des Parameters "128 Bytes Record Capacity" des Programmausdrucks "STAT DSK:".

Zahl der Directory Blöcke Hier bestimmen Sie, wieviele Blöcke der unter Datenblocklänge bestimmen Länge das Directory umfassen soll. Sie müssen darauf achten, daß die Zahl der gesamten Directorybytes 8192 (gleich 8K Bytes) nicht überschreitet.

Zahl der Systemspuren Hier geben Sie die Anzahl der geschützten Spuren am Anfang der Diskette an. Beachten Sie, daß Sie beim Doppelspurverfahren mit n Systemspuren in Wirklichkeit nur n/2 Spuren schützen. Dieser Wert ist identisch mit dem Wert des Parameters "Reserved Tracks" des Programmausdrucks "STAT DSK:".

Dritte Bildschirmseite

Auf der dritten Bildschirmseite können Sie die Sektortabelle programmieren oder editieren. Als erstes werden in sieben Zeilen insgesamt 92 Elemente angezeigt. In der Regel werden nicht alle wirklich benötigt. Die Zahl der relevanten Elemente schwankt bei den verschiedenen Pdrivedatensätzen zwischen 20 und 80.

Jetzt können Sie neue Werte für die einzelnen Elemente eingeben. Dabei korrespondieren die einzelnen Positionen mit den oberen. Probieren Sie das am besten einmal aus. Wollen Sie einen Wert nicht verändern, so drücken Sie einfach <NEW LINE>. Für 0 geben Sie -1 ein, da 0 als keine Änderung erkannt wird. Wollen Sie die Editierung der Sektortabelle abbrechen, so geben Sie -2 ein. Das Programm geht dann zur vierten Bildschirmseite über.

An dieser Stelle taucht vielleicht die Frage auf, wofür man eigentlich eine Sektortabelle benötigt? Mit Hilfe der Sektortabelle wird eine logische Sektornummer in eine physikalische Sektornummer umgesetzt. Dabei korrespondiert dann die logische Nummer 0 mit dem Inhalt der ersten Stelle der Sektortabelle, die logische Nummer 1 mit dem Inhalt der zweiten Stelle der Sektortabelle usw..

Würde keine Umsetzung der logischen in physikalische Sektoren stattfinden, würde das Einlesen einer Spur so ablaufen: Der Controller soll zuerst den Sektor 0 einlesen. Nachdem das geschehen ist, benötigt der Computer immer noch zusätzlich Zeit, um den Inhalt des Sektors an die richtige Adresse zu kopieren, oder sonstwie weiterzuverarbeiten. Als nächstes soll der Sektor 1 eingelesen werden. Da aber der Computer Zeit verbraucht hat, ist der Sektor 1 gerade eben unter dem Tonkopf vorbeigerauscht. Der Controller muß jetzt noch eine ganze Diskettenumdrehung warten, bis der Sektor 1 wieder am Tonkopf vorbei kommt. Nach diesem Verfahren benötigt der Computer mit Controller zum Lesen einer Spur mit 10 Sektoren insgesamt mindestens 11 Umdrehungen der Diskette und damit 2,2 Sekunden.

Setzt man jedoch die logischen Sektoren mit der Sektortabelle (0,4,8,2,6,1;5,9,3,7) in physikalische um, so läuft ein Einladen so ab: Wie oben wird als erstes der Sektor 0 eingeladen und bei der Weiterverarbeitung Zeit verbraucht. Dann soll der Sektor 1 eingeladen werden. Laut Sektortabelle wird dieser aber in die Nummer 4 umgewandelt. Der Controller soll also als nächstes den Sektor 4 einladen. Und siehe da, gerade in diesem Augenblick kommt der Sektor 4 am Tonkopf vorbei und kann sofort eingeladen werden. Dem CP/M wird der Inhalt dieses Sektors als Sektor 1 übergeben, obwohl er auf der Diskette die Nummer 4 trägt. Der Computer hat also in unserem Beispiel zum Verarbeiten eines Sektors soviel Zeit, wie drei Sektoren (1,2,3) benötigen um am Tonkopf vorbei zu kommen. Das sind ca. 60 Millisekunden. Um eine Spur mit 10 Sektoren zu lesen werden dann nur noch 3,8 Umdrehungen der Diskette benötigt, das sind 0,76 Sekunden. Die Zeitersparnis gegenüber dem ersten Beispiel beträgt rund 65 %. Damit ist das Vorhandensein einer Sektortabelle wohl ausreichend erklärt.

Die Sektortabelle des BIOS 2.1 erfüllt insgesamt drei Funktionen. Die erste, das sogenannte Interleaving mit dem entsprechenden Interleavingfaktor habe ich bereits erläutert. Die zweite ist das Umrechnen der ersten Nummer der logischen und der physikalischen Sektoren. Die logischen Sektoren zählen immer von 0 an aufwärts. Die physikalische Nummerierung dagegen kann mit 0 oder mit 1 beginnen. Fängt sie mit 0 an, muß innerhalb der Sektortabelle auch eine 0 stehen. Fängt sie jedoch mit 1 an, so wird es etwas komplizierter. In einem solchen Fall ist die Anzahl der "CP/M Sektoren", die in einen Hardsektor paßt die kleinste Zahl, die innerhalb der Sektortabelle auftauchen darf. Ein Beispiel:

Beim Osborne unter Double Density werden 1K-Bytes lange Sektoren auf die Diskette geschrieben. Die kleinste physikalische Nummer, die auf der Diskette steht ist nicht die 0 sondern die 1. Da aber in 1K-Bytes 8 "CP/M Sektoren" hineinpassen und damit gleichzeitig geschrieben oder gelesen werden, ist die kleinste Zahl, die innerhalb unserer Sektortabelle auftauchen, darf nicht die 0 sondern die 8. Da bei diesem Format insgesamt 40 "CP/M Sektoren" auf eine Spur passen, läuft die Nummerierung in der Sektortabelle von 8 bis 47. Schauen Sie sich das beim Format "OSBORNR DD" einmal an.

Die dritte Funktion, die die Sektortabelle im BIOS 2.1 erfüllen muß, ist die Realisierung des Doppelsektorverfahrens zum Ansteuern der Rückseite von Disketten. Dazu addieren Sie einfach die Zahl 100 zu den Sektoren, die auf der Rückseite liegen sollen. Diese 100 werden vom BIOS 2.1 beim normalen Betrieb wieder abgezogen. Nur werden solche Sektoren eben von der Rückseite der Disketten gelesen, oder auf ihr geschrieben. Dadurch können Sie die doppelte Anzahl von Elementen in die Sektortabelle eingeben. Genau eine Hälfte dieser Elemente muß dabei größer gleich 100 sein und die andere kleiner als 100. Einen Sektor, den Sie, zum Beispiel, mit 137 eingeben, wird auf der Diskette vom BIOS 2.1 unter der Nummer 37 gesucht. Ein Sektor mit der Nummer 37 würde ebenfalls unter der Nummer 37 auf der Diskette gesucht. Nur, der Sektor 37 liegt auf der Vorderseite der Diskette und der Sektor (137-100) auf der Rückseite.

Innerhalb einer Sektortabelle dürfen nie zwei gleiche Elemente stehen, da sonst das Blocking und Deblocking außer Funktion gesetzt wird. Datenverlust wäre die unmittelbare Folge! Dabei gelten die Sektorelemente 137 und 37 nicht als gleich, obwohl sie auf der Diskette die selbe Nummer tragen.

Wenn Sie die Diskette eines fremden Computers anpassen wollen, ist das Erstellen der richtigen Sektortabelle mit Sicherheit das Schwierigste. Dazu müssen Sie im BIOS des entsprechenden Computers, dessen Diskette Sie anpassen wollen, nachschauen. Dabei steht die Sektortabelle bei jedem BIOS an einer anderen Stelle im RAM Speicher. Ein Rezept kann ich Ihnen dafür deshalb leider nicht geben.

Vierte Bildschirmseite

Auf der vierten Bildschirmseite können Sie nur den Namen eines Pdrivedatensatzes ändern. Dazu geben Sie einfach einen neuen ein. Er darf nicht länger als maximal 15 Zeichen sein. Wenn Sie den Datensatznamen nicht ändern wollen, geben Sie nur <NEW LINE> ein.

PD .COM

Bis jetzt haben wir nur gesehen, was ein Pdrivedatensatz darstellt und wie man ihn programmiert. Hier soll nun gezeigt werden, wie man einen Pdrivedatensatz aus der Datei PDRIVE.SYS entnimmt und ihn einem bestimmten Laufwerk zuordnet. Dazu dient das Programm PD.COM. Es antwortet nach dessen Anstartung mit:

Einstellung der Laufwerksparemeter
Welche Diskette wollen Sie bearbeiten:

Sie können mit dem Programm PD.COM auch die Pdrivedatensätze von Disketten bearbeiten, die in den Laufwerken B bis D eingelegt sind. An dieser Stelle des Programms geben Sie also das Laufwerk an, in dem die zu bearbeitende Diskette liegt. Nachdem Sie A,B,C oder D eingegeben haben, fährt das Programm wie folgt fort:

Die Parameter der Diskette in Laufwerk A.

A:S40 DD

B:S40 DD

C:S40 DD

D:S40

Eingabe oder NEW LINE :

Wenn Sie die Parameter einer Diskette in Laufwerk B bis D abgefragt haben, so fehlt die Angabe zum Laufwerk A, da dessen Parameter nicht - wie die andern - in der Datei SYS.SYS steht, sondern im BOOTER auf den Systemspuren der Diskette. (Siehe dazu unter BOOTGEN.COM in diesem Handbuch.) Auf keinen Fall aber können Sie die A-Parameter einer Diskette ändern!

Durch Betätigen von <NEW LINE> ohne sonstige Angaben erfolgt ein Warmstart des CP/M Systems. Sie hätten sich dann mit diesem PD.COM Durchlauf nur die aktuellen Pdrivedatensätze listen lassen.

Angezeigt werden hier nur die Namen der aktuellen Datensätze und nicht deren Inhalte. Das ist in der Regel auch nicht nötig, da man weiß, welcher Name zu welchem Datensatz gehört. Man kann sich damit ein Bild machen, welche Diskette man in ein bestimmtes Laufwerk schieben darf und welche nicht.

Nun kommen wir zur Programmierung eines bestimmten Laufwerkes. Dazu geben Sie einfach an Stelle von <NEW LINE> einen Laufwerksnamen, gefolgt von einem Datensatznamen ein. Nun folgen einige Beispiele von gültigen Eingaben:

B:S40

C:OSBORNE

B:GENIE III DS