

hilfe vorgesehen – und man fragt sich natürlich auch, warum es davon eigentlich keine Quelle gibt. Und spätestens hier kommt wirklich Heiterkeit auf: Der Treiber ist in der Tat IOCTL-fähig, nur leider verbirgt er sämtliche Geheimnisse, wie denn nun so ein IOCTL-fähiger Treiber aussehen soll, dadurch, daß er alle interessanten Tätigkeiten über den Interrupt 2Fh abwickelt – und zwar mit der undokumentierten Funktion 08. Wir haben die Finger von der IOCTL-Unterstützung gelassen, weil nicht klar ist, ob Microsoft den Weg über 2Fh aus Bequemlichkeit gewählt hat, oder weil Fehler umschifft werden mußten.

Umwege

Und damit stellt sich schon die nächste Frage: Warum wird die Formatierung anderer als der Standardformate überhaupt mittels Treiber gelöst? ('Standardformate' soll heißen: die dem jeweiligen Rechner von seinem Hersteller zugebilligten Formate.) Der Treiber übernimmt nämlich nur die Parameterauswertung seiner Aufrufzeile und ermittelt daraus das jeweilige Format, schiebt aber die eigentliche Arbeit wieder ans DOS ab. Außerdem frißt er Speicherplatz, erst recht, wenn man ihn für verschiedene physikalische Laufwerke oder Formate mehrmals laden muß.

Vielleicht hatte man ja bei Microsoft mal vor, einen intelligenten Formatierer zu schreiben, der ohne DRIVER.SYS auskommt. Denn in einigen MSDOS-Handbüchern findet man beim Formatter-Aufruf Hinweise auf Parameter zur Angabe der Sektorzahl pro Spur und der Spurzahl. Aber stets erhielten wir bei deren Anwendung eine Meldung wie 'Parameter nicht unterstützt' oder Sinngemäßes.

Aber mal ehrlich: Wem kann denn an einem anwenderfreundlichen Betriebssystem gelegen sein, das womöglich statt 'Ungültiger oder fehlender Parameter' selbständig erklärt, wie man es bedient? Na, doch allenfalls dem Anwender – und bei dem reicht es schließlich, daß er die Software kauft. . .

Für die Praxis im folgenden heißt das also, Formatieren geht nur über DRIVER.SYS. Da

hiermit aber keineswegs beliebige Formate zugelassen sind, lassen sich nur Disketten mit Formaten erzeugen, die mehr oder weniger von IBM unterstützt werden. Eigentlich verbleibt derzeit auf PCs/ATs nur

stellte Disketten zu verwenden. Manchmal kann man sich aber auch behelfen: Das zweiseitige Atari-Format zum Beispiel läßt sich durch Formatieren im System/2-Format und anschließendes Übertragen der ersten 18

rungen an anderer Stelle auf der Disk befinden.

Die möglichen Parameter, mit denen DRIVER.SYS innerhalb CONFIG.SYS aufgerufen werden kann, zeigt die Tabelle. Ganz wesentlich ist, daß hinter dem Treibernamen und vor dem ersten Schrägstrich unbedingt ein Leerzeichen eingefügt werden muß, damit die Parameter akzeptiert werden.

Zwischen den Treibern von MS- und PCDOS gibt es noch ein paar kleinere Unterschiede. Vor allem gibt es unter MSDOS meistens noch das Config-Kommando DRIVPARM, das die Möglichkeit bietet, das Programm FORMAT mit anderen als den üblichen Parametern auch auf Drives mit den Laufwerksbuchstaben A: und B: aufzurufen. Dies Kommando 'steckt' übrigens auch in neueren PCDOS-Versionen im IBMBIO.COM, aber man hat es einfach totgelegt. Das Laden DRIVER.SYS hingegen erzeugt immer ein neues logisches Laufwerk (neuer Buchstabe), und nur auf diesem ist dann das gewünschte Format per FORMAT ohne Parameter-Angabe zu erzeugen.

Formatieren in der Praxis

Um also beispielsweise eine PS/2-gerechte 3,5-Zoll-Disk auf einem PC oder AT mit nachträglich eingebautem 3,5-Zoll-Drive zu formatieren, gibt man hinter DEVICE = DRIVER.SYS zunächst mit dem Parameter '/D:xx' das physikalische Laufwerk an, üblicherweise 0 oder 1 für xx. Sodann folgt der Parameter '/F:2', und das war's.

Was vielleicht noch nicht ganz klar herausgekommen ist: Nur das Formatieren muß beim 720-KB-Format unter DOS ab Version 3.2 über den Treiber bewerkstelligt werden. Anders als bei MF-Laufwerken, wo unser Treiber das Double-Stepping abstellen muß, kann auf Einfachlaufwerken der Boot-Sektor ausgewertet werden, und das DOS stellt sich korrekt darauf ein. In einem AT etwa, in den als zweites Laufwerk ein 3,5-Zoll-Drive eingebaut wurde, lassen sich PS/2-Disketten auch unter Laufwerk B: beschreiben und lesen.

Leider ist es etwas kniffliger und umständlicher, eine normale 80-Spur-Diskette im MF-Laufwerk des AT zu formatieren.

Aufruf in CONFIG.SYS:

device=driver.sys /D:ddd/F:f/C/H:hh/N/S:ss/T:ttt

/D:ddd

Angabe der physikalischen Laufwerksnummer,

Wertebereich 00 < ddd < 127 bei Floppy
128 < ddd < 255 bei Harddisk

/F:f

Laufwerkstyp

f = 0: 160/180 und 320/360 KB Floppy

1: 1,2 MByte MF-Floppy

2: 720 KByte Floppy

3: 8 Zoll, Single Density

4: 8 Zoll, Double Density

5: Festplatte

6: Magnetband

7: andere Geräte, ab PCDOS 3.3

3,5 Zoll High Density mit 1,44 MByte

Alle folgenden Angaben sind optional, sofern mit '/F:' schon ein konkretes Format vorgegeben wurde.

/C

Gibt an, daß ein Change-Disk-Signal vom Laufwerk generiert wird und ausgewertet werden soll. Nützt im PC nichts, da dieser das Signal hardwaremäßig nicht auswerten kann.

/H:hh

Anzahl der Köpfe (entspricht bei Floppies den Seiten), Wertebereich 1 < hh < 99

/N

Zeigt einen Massenspeicher mit nicht wechselbarem Datenträger an (z. B. Harddisk)

/S:ss

Anzahl der Sektoren pro Plattenoberfläche, Wertebereich 1 < ss < 99

/T:ttt

Anzahl der Zylinder (Spuren pro Plattenoberfläche), Wertebereich 1 < ttt < 999

Anmerkung aus dem technischen PCDOS-Handbuch (IBM):

DRIVER.SYS ist nicht für IBM-Festplattenlaufwerke bestimmt.

Weiterhin sind gemäß IBM nur die Laufwerkstypen (/F:) 0, 1, 2 und 7 definiert.

Für DRIVPARM, das nur bei MSDOS unterstützt wird, gelten diese Parameter in gleicher Weise, jedoch erfolgt der Aufruf in CONFIG.SYS ohne vorangestelltes 'device=', also in der Form drivparm=/parameter/parameter...

DRIVPARM ermöglicht es, das Standardformat des Formaters für die Laufwerke 1 und 2 unter den Laufwerksbuchstaben A: und B: durch das per Parameter eingestellte zu überschreiben.

Wenn es darum geht, ein Diskettenformat zu erzeugen, für das die ursprüngliche Rechner-Laufwerk-Konfiguration nicht ausgelegt ist, muß man DRIVER.SYS oder DRIVPARM verwenden.

das 720-KB-Format mit genau den von IBM vorgegebenen Parametern. Denn die veränderbaren Parameter lassen keine explizite Einstellung von FAT-Größe, FAT-Anzahl oder ähnliches zu. In den meisten Fällen ist man daher darauf angewiesen, bereits im Fremdformat er-

Sektoren von einer (möglichst leeren) Original-Atari-Disk mittels DEBUG erzeugen. Denn physikalisch sind die Formate gleich, nur die logischen Informationen im Boot-Sektor lauten anders, und demgemäß müssen sich die FATs und das Directory mit korrekten Initialisie-