DISKUS

Version 3.99.7

Das Massenspeicher-Tool

Für Atari 16/32 Bit-Computer und Kompatible

Benutzerhandbuch

Copyright © Uwe Seimet

Dr. Uwe Seimet

Reuterstraße 2

D-76275 Ettlingen

Copyright © 1989-2026 Uwe Seimet

Es wurden alle erdenklichen Maßnahmen getroffen, um die Richtigkeit der vorliegenden Produkt-Dokumentation zu gewährleisten. Es kann jedoch keine Garantie für die Vollständigkeit und Fehlerfreiheit der Software und der dazugehörigen Programmbeschreibung sowie für eventuelle aus der Nutzung resultierende Folgeschäden übernommen werden.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Liebe Anwenderin, lieber Anwender,

dies ist die Dokumentation zu DISKUS, dem universellen Werkzeug zur Bearbeitung und Rettung von Daten auf Massenspeichern. Seit der Einstellung des kommerziellen Supports Ende 2008 ist DISKUS freie Software.

Die Distribution enthält die folgenden Dateien:

DISKUS.APP	Dies ist die eigentliche DISKUS-Programmdatei
DISKUS.PDF	Dieses Benutzerhandbuch
NOBOOT.BOT	Diese Datei lässt sich mit DISKUS als Bootsektor auf einer Diskette installieren und unterdrückt das Booten von Platte. Mehr dazu in der Datei README.
ICONS.RSC	Ein Satz mit Icons für DISKUS
README	Hinweise, die Sie vor der Benutzung von DISKUS lesen sollten

Informationen zu Neuerungen und Updates gibt es im Internet:

https://www.hddriver.net

Bleibt mir noch, Ihnen viel Erfolg zu wünschen, falls bei Problemen mit Massenspeichern der Einsatz von DISKUS erforderlich ist.

Uwe Seimet, im Mai 2026

Inhalt

1. Allgemeine Hinweise....1
2. Der Desktop....2
2.1. Das Editieren von Daten....2
2.2. Die Schriftgröße....2
2.3. Die Datenfenster....3
2.4. Lesen und Schreiben von Daten....3
2.4.1. Bootsektor....3
2.4.2. FAT1....3
2.4.3. FAT2....4
2.4.4. Directory....5
2.4.5. log. Cluster....6
2.4.6. phys. Cluster....7
2.4.7. Sektor....7
2.4.8. Harddisk....8
2.5. Statusanzeigen....8
2.5.1. Cluster belegt....8
2.5.2. BPB ungültig....9
2.5.3. Schreibschutz....9
2.6. Laufwerke....9
2.7. Position....10
2.8. Die Cursorposition....11
2.9. Die Laufwerkskennung....11
3. Die Dateiauswahlbox....12
4. Das "Datei"-Menü....14
4.1. Information....14
4.2. Öffnen....15
4.3. Kopieren....15
4.4. Löschen....15
4.5. Eliminieren....15

4.6. Verketten....15
4.7. Vertauschen....16
4.8. Programmflags....16
4.9. Eintrag löschen....17
4.10. Leerer Eintrag....17
4.11. Neuer Ordner....17
4.12. Ende....17
5. Das "Disk" Menü....18
5.1. Information....18
5.2. Struktur....18
5.3. Kopieren....19
5.4. Löschen....20
5.5. Bootsektor....20
5.6. Format definieren....21
5.7. Formatieren....21
5.8. Sektorgröße ändern....22
6. Das "Harddisk"-Menü....23
6.1. Busse neu scannen....23
6.2. Geräteauswahl....23
6.3. Geräteinformation....23
6.4. Kommando senden....24
6.5. Rootsektor....25
6.6. Partitionen....25
6.7. Sektoren kopieren....27
6.8. Partition initialisieren....27
7. Das "Sektor"-Menü....29
7.1. Physikalisch lesen/schreiben....29
7.2. Sektorfolge laden/speichern....29
8. Das "Block"-Menü....30
8.1. Alles markieren....30

8.2. Ausschneiden....30
8.3. Kopieren....30
8.4. Einfügen....30
8.5. Überschreiben....30
9. Das "Spezial"-Menü....31
9.1. Daten laden/speichern....31
9.2. Daten drucken....31
9.3. Daten suchen....31
9.4. Weitersuchen....32
9.5. Daten testen....32
9.6. Daten retten....34
9.6.1. Datei/Ordner restaurieren....34
9.6.2. Wurzelverzeichnis rekonstruieren....35
9.6.3. FAT-Prototyp erzeugen....36
9.6.4. Sektor zurückholen....36
9.7. Daten schützen....36
9.8. Medium kopieren....37
9.9. Daten optimieren....37
9.9.1. Verzeichnisse aufräumen....38
9.9.2. Dateien sortieren....38
9.9.3. Defragmentierung....39
9.10. FATs schreiben....39
9.11. Umrechnung....39
9.12. Optionen....40
10. Hinweise zur Tastaturbelegung....41
11. Besonderheiten bei MiNT und MagiC....42
12. Der SCSI-Treiber....43
13. Daten auf Disketten, Festplatten und Speicherkarten....44
13.1. Welcher Diskettentyp ist geeignet?....44
13.2. Das physikalische Diskettenformat....44

13.3. Das logische Diskettenformat....44
13.4. Der Bootsektor....45
13.5. Die Diskbelegungstabelle (FAT)....46
13.6. Das Wurzelverzeichnis....47
13.7. Der Aufbau von Ordnern....48
13.8. Der Bios Parameter Block (BPB)....49
13.9. Alternative Dateisysteme....50
14. Das Beheben von Datenverlusten....51
14.1. Fehler in Verzeichnissen....51
14.2. Fehler in der FAT....52
14.3. Fehler im Bootsektor....53
15. Das Wiederherstellen von Dateien....54
15.1. Automatisches Restaurieren gelöschter Dateien....54
15.2. Das Restaurieren "per Hand"....55
15.3. Lokalisieren von Fehlern....56
15.4. Das Verhindern von Datenverlusten....57
16. Das Arbeiten mit Disketten....58
16.1. Erhöhung der Diskettenkapazität....58
16.2. Änderung des physikalischen Formats....58
16.3. HD-Laufwerke am ST....58
16.4. HD-Laufwerke am TT und Falcon030....58
16.5. ED-Laufwerke....59
16.6. Der Interleave-Faktor....59
16.7. Atari-Disketten und PC-Kompatibilität....59
17. Das Arbeiten mit Festplatten und Speicherkarten....61
17.1. Der Rootsektor....61
17.2. Aufteilung in Partitionen....62
17.2.1. GEM-Partitionen....62
17.2.2. BGM-Partitionen....62
17.2.3. XGM-Einträge....62

17.3. DOS/Windows-konforme Partitionierung....63
17.4. TOS/Windows-kompatible Partitionierung....63
17.5. GPT-Partitioniierung....63
17.6. Hinweise zur Partitionierung....63
17.7. Physikalische Zugriffe....63
17.8. Die Plattenstatistik....64
17.9. SCSI-, IDE/ATAPI- und USB-Peripherie am Atari....65
17.9.1. Geräte am ACSI- und SCSI-Bus....65
17.9.2. IDE-Festplatten....65
17.9.3. ATAPI- und SATA-Peripherie....65
17.10. USB-Festplatten....66
17.11. Hinweise zum Kauf von Peripheriegeräten....66

1. Allgemeine Hinweise

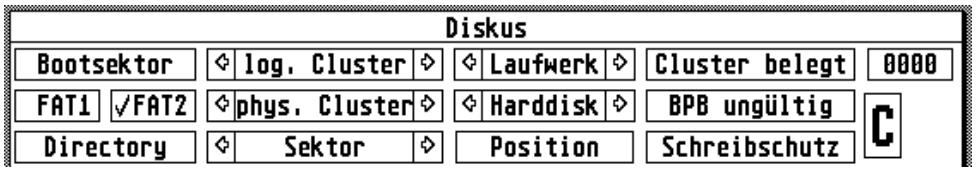
Willkommen zu DISKUS, dem universellen Werkzeug zur Bearbeitung von Daten auf Massenspeichern. Dieses Handbuch führt Sie in die Bedienung von DISKUS ein und bietet darüber hinaus viele Informationen allgemeiner Art. Mit Hilfe dieser Informationen sollen Sie in die Lage versetzt werden, auch dann von DISKUS zu profitieren, wenn Sie (noch) kein Experte auf dem Gebiet der Datenspeicherung bei Ataris sind. Sofern Sie mit Funktionen von DISKUS experimentieren wollen, sollten Sie dies mit einer Diskette oder Platte/Speicherkarte tun, auf der sich keine wichtigen Daten befinden.

Die Bedienung von DISKUS gestaltet sich grundsätzlich so, wie Sie es von Software für den Atari gewohnt sind. Kommandos, die über die Menüleiste erreichbar sind, werden meist zusammen mit dem Menütitel aufgeführt, z. B. **Harddisk->Geräteinformation**.

In einigen Dialogen erwartet DISKUS die Eingabe einer Zahl. Hier sind in der Regel sowohl dezimale als auch hexadezimale Eingaben erlaubt. Hexadezimale Angaben werden durch ein führendes "$"-Zeichen gekennzeichnet.

2. Der Desktop

Der Desktop von DISKUS lässt sich in zwei Bereiche unterteilen. Im oberen Bildschirmbereich befinden sich in der Regel Fenster, in denen diverse Daten dargestellt werden. Im unteren Bereich wird ein Fenster mit Knöpfen angezeigt, mit denen verschiedene Operationen eingeleitet werden können.

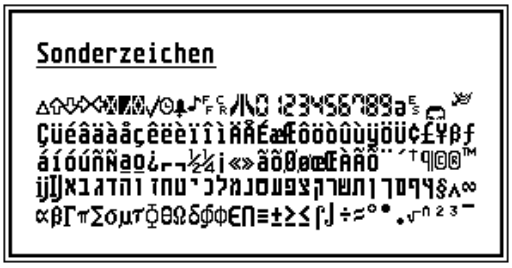


2.1. Das Editieren von Daten

DISKUS unterstützt zwei Datenfenster, in denen Sektor- und FAT-Daten des aktuellen Laufwerks dargestellt werden. Die Daten des Sektorfensters werden sowohl hexadezimal als auch im ASCII-Format dargestellt. Am linken Rand jedes Fensters befindet sich die Nummerierung der einzelnen Bytegruppen in hexadezimaler Form.

Um ein Byte zu editieren, wird der Cursor mit der Maus oder den Cursortasten auf die gewünschte Position gesetzt. Nun können Sie über hexadezimale Tastatureingaben den Inhalt des Bytes beliebig verändern. Um ASCII-Daten zu editieren, setzen Sie den Cursor auf ein Zeichen im ASCII-Bereich des Datenfensters.

Wird der ASCII-Bereich des Datenfensters editiert, lassen sich auch Sonderzeichen eingeben. Hierzu muss die [Insert]-Taste betätigt werden. Die Sonderzeichen lassen sich nun per Mausklick auswählen.



Innerhalb des Datenfensters befinden sich zwei Cursor, von denen einer fett, der andere in Form zweier senkrechter Striche dargestellt wird. Der fette Cursor befindet sich in dem Bereich der Anzeige, der editiert werden kann. Der Cursor kann nicht nur mit der Maus, sondern auch über die Pfeiltasten der Tastatur bewegt werden. Eine Sonderfunktion hat in diesem Zusammenhang die [Home]-Taste. Sie bewegt den Cursor auf das erste angezeigte Byte eines Fensters. Wird zusätzlich zur [Home]-Taste eine [Shift]-Taste betätigt, wird der Cursor auf das allererste Datenbyte gesetzt. Über die [Tab]-Taste wird zwischen dem hexadezimalen Teil der Anzeige und der ASCII-Anzeige gewechselt.

2.2. Die Schriftgröße

Normalerweise werden alle Daten in normaler Schriftgröße (8*16 Systemfont) dargestellt. Bei einer Auflösung von mindestens 640 x 400 Bildpunkten lässt sich eine kleinere Schrift (8*8 Systemfont) verwenden. Die Zeichengröße umschalten kann man bei gedrückter [Shift]-Taste mit einem Klick auf das das Bildschirmformatfeld (Vollfeld) der Fenster.

2.3. Die Datenfenster

DISKUS benutzt für die Darstellung von Daten zwei Fenster mit ähnlichem Aufbau. Das Sektorfenster stellt die Daten des Sektors oder Clusters dar, der momentan bearbeitet wird. Dieses Fenster öffnet sich bei Bedarf automatisch, sobald ein Sektor gelesen wird. Sektororientierte Operationen lassen sich in der Regel nur bei geöffnetem Sektorfenster aufrufen.

Das FAT-Fenster wird durch einen Mausklick bei gleichzeitig gedrückter [Shift]-Taste auf den **FAT 1**- oder **FAT 2**-Knopf geöffnet. In diesem Fenster wird ein Überblick über die Diskbelegungstabelle gegeben, die Angaben darüber enthält, welche Cluster des Mediums belegt sind.

2.4. Lesen und Schreiben von Daten

Mit den Knöpfen im unteren Teil des DISKUS-Desktops werden in erster Linie Lese- und Schreiboperationen veranlasst. In der Regel startet ein Mausklick mit der linken Maustaste eine Leseoperation, bei gleichzeitig gedrückter [Shift]-Taste werden Schreiboperationen eingeleitet. Die Bedeutung der einzelnen Knöpfe wird nun erläutert.

2.4.1. Bootsektor

Der Bootsektor ist der erste Sektor eines Mediums. Er enthält Informationen über die Parameter einer Diskette oder einer Partition. Wird der **Bootsektor**-Knopf mit der linken Maustaste angeklickt, wird der Bootsektor geladen und angezeigt. Klicken Sie den **Bootsektor**-Knopf bei gedrückter [Shift]-Taste an, können Sie den Inhalt des Datenfensters als neuen Bootsektor auf ein Medium schreiben. Hierbei bleibt die Ausführbarkeit des Bootsektors unverändert.

Ist der Bootsektor nicht lesbar, ist kein logischer Zugriff auf die einzelnen Sektoren des Mediums möglich. Auch bei einem ungültigen Bootsektor haben Sie mit DISKUS die Möglichkeit, physikalisch zuzugreifen.

DISKUS kann die Werte der im Bootsektor eingetragenen Parameter im Klartext anzeigen. Wird ein Mausklick auf einen Bereich des Bootsektors durchgeführt, der einen solchen Parameter repräsentiert, wird bei gedrückt gehaltener Maustaste in der Infozeile des Datenfensters angezeigt, um welchen Parameter es sich handelt und welchen Wert dieser besitzt.

Ähnliche Angaben auf einen Blick erhält man nach dem Anklicken der Laufwerksbezeichnung in der rechten unteren Bildschirmecke bei gedrückter [Shift]-Taste.

2.4.2. FAT1

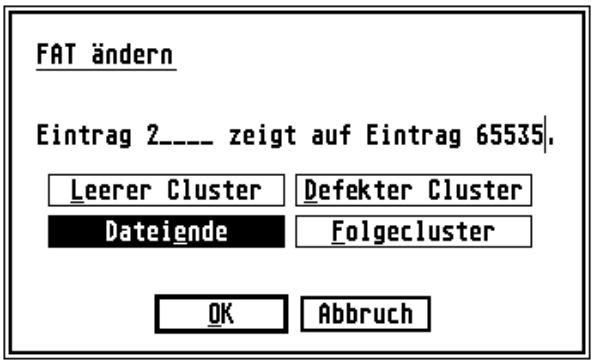
FAT ist die Abkürzung für *File Allocation Table*. Diese Belegungstabelle enthält Angaben darüber, welche Dateien welche Datencluster belegen. Aus Sicherheitsgründen befinden sich meist zwei FATs auf einer Diskette bzw. Partition. DISKUS unterstützt Medien mit einer 16-Bit-FAT.

Nach dem Anklicken von **FAT1** wird der erste Sektor der ersten FAT geladen und angezeigt. Wird **FAT1** bei gedrückter [Shift]-Taste angeklickt, erhält man eine detaillierte FAT-Anzeige, in der die Belegung jedes Clusters durch eines der unten aufgeführten Symbole markiert wird:

Anzeige	*Bedeutung*
" "	Cluster ist reserviert
"-"	Cluster ist unbelegt
"*"	Cluster gehört zu einer Datei
"#"	Cluster ist letzter Cluster einer Datei
"?"	Cluster ist defekt

Um die Daten eines Clusters aus der FAT-Übersicht heraus anzeigen zu lassen, führt man einen Mausklick auf das entsprechende Symbol der Übersicht aus. Ein Doppelklick zeigt Informationen an, aus denen hervorgeht, zu welcher Datei der angeklickte FAT-Eintrag gehört.

Nach dem Anklicken eines FAT-Eintrags bei gedrückter [Shift]-Taste lässt sich dieser Eintrag editieren.



Dieser Dialog wird bei der Beschreibung des Knopfes **Cluster belegt** näher erklärt.

2.4.3. FAT2

Die zweite FAT ist die einzige der beiden FATs, die vom Betriebssystem des Atari genutzt wird. Zwar wird bei Schreiboperationen stets auch FAT 1 aktualisiert, bei Leseoperationen zählt jedoch nur FAT 2. Selektiert man diesen Knopf mit der linken Maustaste, erscheint der erste Sektor von FAT 2 im Datenfenster.

Stehen die Daten eines FAT-Sektor im Datenfenster, erhält man während des Anklickens eines FAT-Eintrags bei weiterhin gedrückter Maustaste innerhalb der Infozeile des Datenfensters Informationen darüber, um welchen FAT-Eintrag es sich handelt und auf welchen Folge-Eintrag er zeigt.

Wird auf einen Eintrag des im Datenfenster dargestellten FAT-Sektors ein Doppelklick ausgeführt, wird der Daten-Cluster angezeigt, der diesem Eintrag zugeordnet ist. Bei einem Doppelklick mit der rechten Maustaste wird der Cursor auf den nächsten logischen FAT-Eintrag gesetzt. Zeigt Eintrag 30 auf Eintrag 349, wird zunächst der FAT-Sektor geladen, der Eintrag 349 enthält. Anschließend wird der Cursor auf diesen Eintrag gesetzt. Durch eine Folge von Mausklicks kann man sich so durch die FAT bewegen.

Beim Anklicken des Knopfes **FAT2** bei gedrückter [Shift]-Taste erhalten Sie eine Übersicht über die gesamte FAT, wobei mehrere Cluster bei Bedarf zu einer Einheit zusammengefasst werden. In der FAT-Gesamtübersicht wird keine Unterscheidung zwischen defekten und belegten Sektoren getroffen.

Nach dem Start von DISKUS wird der Knopf **FAT2** abgehakt angezeigt. Dies besagt, dass der Inhalt der

zweiten FAT für alle FAT- und dateiorientierten Operationen zu Grunde gelegt wird. Nach dem gleichen Prinzip geht auch das Betriebssystem der Atari-Computer vor. Ist ein Sektor der zweiten FAT defekt, kann auf ein Medium nur noch über die erste FAT zugegriffen werden. Ein Doppelklick mit der rechten Maustaste auf den **FAT1**-Knopf teilt DISKUS mit, intern mit der ersten FAT zu arbeiten. Wird der Knopf **FAT2** mit der rechten Maustaste doppelt angeklickt, wird wieder auf FAT 2 umgeschaltet.

2.4.4. Directory

Das Directory enthält die Daten aller Dateien, die sich im Wurzelverzeichnis eines Mediums befinden. Der erste Directory-Sektor wird über den **Directory**-Knopf in die Anzeige gebracht.

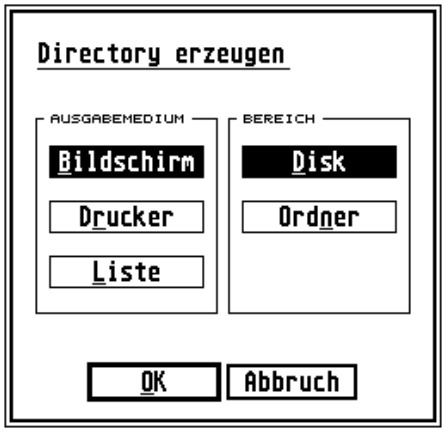
Wird ein Sektor oder Cluster dargestellt, der Teil eines Verzeichnisses ist, werden mit Hilfe der linken Maustaste die Daten der angezeigten Dateien im Klartext erhalten. Klicken Sie hierzu die Bytes eines Datei-Eintrags an und halten Sie die Maustaste gedrückt. Daraufhin wird der entsprechende Bildschirmbereich invertiert dargestellt und in der Infozeile des Fensters erscheint die gewünschte Information. Bei der Anzeige der Dateiattribute erhält man neben dem binären Wert des Attributbytes die folgenden Information über die von TOS verwendeten Attribute:

Anzeige	***Bedeutung***
N	Der Eintrag enthält den Namen des Mediums
O	Der Eintrag repräsentiert einen Ordner
R	Die Datei kann nur gelesen werden
S	Es handelt sich um eine Systemdatei
V	Die Datei ist eine versteckte Datei
A	Das Archiv-Bit ist gesetzt
L	Der Eintrag ist ein symbolischer Link (MiNT und MagiC)

Handelt es sich bei den mit der Maustaste selektierten Daten nicht um Teile eines Datei-Eintrags, kommt den angezeigten Daten keine Bedeutung zu.

Befindet sich ein Directory-Sektor (oder ein Cluster eines Ordners) in der Anzeige, kann der erste Cluster einer Datei durch einen Doppelklick auf den zu öffnenden Dateieintrag in die Anzeige geholt werden.

Wird der **Directory**-Knopf bei gedrückter [Shift]-Taste angeklickt, erscheint ein Dialog, der die Ausgabe der Daten des Wurzelverzeichnisses oder eines Ordners erlaubt.



Die Ausgabe kann auf dem Bildschirm oder einem Drucker erfolgen. Bei entsprechender Einstellung unter **Spezial->Optionen** ist auch die Ausgabe in eine Datei möglich. Wurde ein Bildschirm komplett aufgebaut, kann die Ausgabe mit einem Tastendruck fortgesetzt werden, wobei [Esc] die Operation abbricht. Namen von Dateien, die sich in Ordnern befinden, werden eingerückt dargestellt. Der Startcluster einer Datei lässt sich in die Anzeige bringen, indem man den Dateinamen doppelt anklickt.

Um die von DISKUS erzeugten Verzeichnis-Informationen in Datenbanken einlesen zu können, existiert der Knopf **Liste**, der in der Regel nur mit der Umleitung der Ausgabe in eine Datei verwendet wird. **Liste** gibt alle Datei-Informationen in folgendem Format aus:

Pfad,Name,Datum,Zeit,Länge

Ordner besitzen einen Backslash ("\") als letztes Zeichen des Namens und lassen sich so von normalen Dateien unterscheiden.

2.4.5. log. Cluster

Die eigentlichen Daten befinden sich auf den Daten-Clustern. Der erste Cluster besitzt aus historischen Gründen die Nummer 2 und nicht 0. Wird der Knopf **log. Cluster** mit der linken Maustaste angeklickt, wird der erste Daten-Cluster des aktuellen Laufwerks angezeigt.

Beim Anwählen bei gedrückter [Shift]-Taste wird geprüft, ob der aktuelle Cluster Bestandteil einer Datei ist. In diesem Fall wird angezeigt, zu welcher Datei der Cluster gehört, um welchen Cluster dieser Datei es sich handelt, und auf welchem Byte der Datei sich der Cursor befindet. So kann man sich nach Suchoperationen leicht orientieren, in welcher Datei die Suche erfolgreich war.

Ist der im Datenfenster angezeigte Cluster Bestandteil einer Datei, ist es mit den Pfeilen links und rechts des

log. Cluster-Knopfes möglich, sich in dieser Datei vor und zurück zu bewegen. Beim Anklicken des Pfeils nach rechts wird der nächste Cluster der Datei gelesen und angezeigt, der Pfeil nach links sorgt für das Lesen des vorhergehenden Clusters. Werden die Pfeile bei gedrückter [Shift]-Taste angeklickt, sucht DISKUS den letzten bzw. ersten Cluster der Datei auf, zu der der angezeigte Cluster gehört.

2.4.6. phys. Cluster

Dieser Button erleichtert das Arbeiten mit Clustern, wenn diese nicht logisch (also dateiorientiert), sondern physikalisch (in der Reihenfolge, wie sie sich auf dem Medium befinden) angesprochen werden sollen.

Um einen beliebigen Daten-Cluster in die Anzeige zu bringen, wird der **phys. Cluster**-Knopf mit der linken Maustaste angeklickt. In einer Dialogbox können Sie die Nummer des Clusters eintragen, der von DISKUS geladen und angezeigt werden soll.

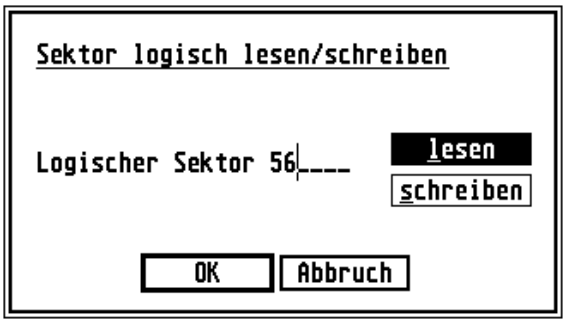
Um einen kompletten Cluster zu schreiben, wird **Cluster** bei gedrückter [Shift]-Taste angewählt.

In der Regel besteht ein Cluster aus zwei Sektoren. Wird die zweite Cluster-Hälfte editiert, also der zweite Sektor eines Clusters, und soll anschließend zurückgeschrieben werden, muss dies mit der soeben beschriebenen Funktion zum Schreiben eines Clusters geschehen. Es genügt nicht, nur den aktuellen Sektor zurückzuschreiben, da dieser meist nur einen Teil des gesamten Clusters repräsentiert.

Die Pfeile rechts und links des **phys. Cluster**-Knopfes holen den folgenden bzw. vorhergehenden physikalischen Cluster in die Anzeige. Nach dem Anklicken der Pfeile bei gedrückter [Shift]-Taste wird der nächste bzw. vorhergehende unbelegte Cluster angezeigt. Dies ist hilfreich, wenn eine gelöschte Datei "per Hand" restauriert werden soll, da man in diesem Fall nur nach unbelegten Clustern Ausschau halten muss.

2.4.7. Sektor

Über diesen Kopf kann ein Medium sektorweise untersucht werden.



Nach Auswahl des **Sektor**-Knopfes bei gedrückter [Shift]-Taste können Sie einen Sektor schreiben.

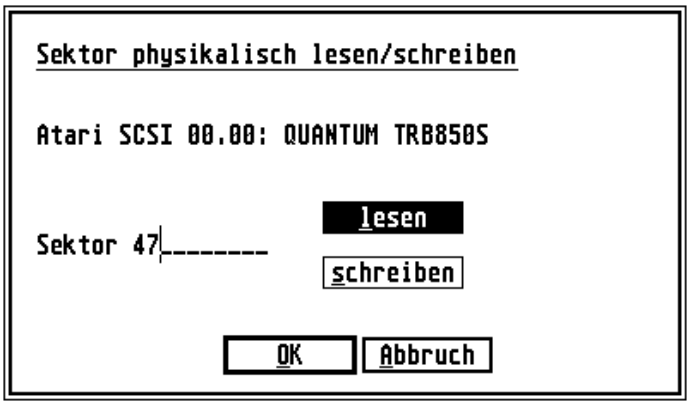
Die Pfeile links und rechts des **Sektor**-Knopfes lassen sich auch per Tastatur über die Cursortasten (Cursor links bzw. Cursor rechts) in Verbindung mit einer [Shift]-Taste bedienen.

Die Einteilung einer Diskette oder Partition in logische Sektoren und Cluster wird vom Betriebssystem verwaltet. Ein Sektor wird durch seine logische Sektornummer angesprochen, ohne dass man wissen muss, wo er sich physikalisch befindet. Logische Zugriffe sind nur möglich, wenn ein Medium ein Format (Dateisystem) besitzt, das vom Betriebssystem erkannt und unterstützt wird. Dazu muss der Bootsektor fehlerfrei sein, oder bei Disketten mit **Disk->Format definieren** das korrekte Format vorgegeben werden. Andernfalls kann nur physikalisch zugegriffen werden. Ist der BPB eines Mediums ungültig führt ein Anklicken des **Sektor**-Knopfes zum Dialog für das physikalische Lesen oder Schreiben und nicht zum Dialog für logische Operationen.

Werden FAT- oder Directory-Sektoren geändert und anschließend auf das aktuelle Medium geschrieben, muss man nach dem Schreibvorgang die [Esc]-Taste betätigen, damit DISKUS die Daten des Mediums neu einliest und alte Informationen verwirft.

2.4.8. Harddisk

Dieser Knopf ist nur dann sichtbar, wenn ein SCSI-Treiber installiert ist, wie er beispielsweise im Festplattentreiber HDDRIVER integriert ist. In diesem Fall besteht über die **Harddisk**-Buttons die Möglichkeit, physikalisch auf eine Fest- oder Wechselplatte, eine Speicherkarte oder auch ein CD-ROM/DVD/BD-Laufwerk zuzugreifen. Dabei verwendet DISKUS das bei der Geräteauswahl ausgewählte Gerät. Diese Funktion unterstützt Sektornummern bis zu 64 Bit, kann also auf nahezu beliebig große Medien zugreifen, sofern der vorhandene SCSI-Treiber dies unterstützt.



Bei physikalischen Schreibzugriffen auf die Festplatte ist Vorsicht angebracht, da ein Fehler wichtige Daten zerstören kann. Sollten Sie versehentlich einen falschen Sektor auf die Platte geschrieben haben, können Sie den ursprünglichen Sektorinhalt mit **Sektor zurückholen** unter **Spezial->Daten retten** wieder in das Datenfenster bringen und so einen Schaden eventuell beheben.

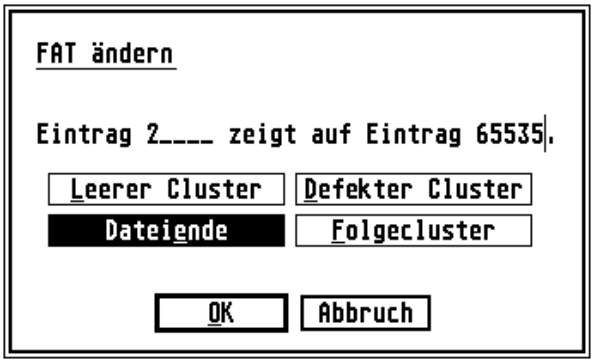
2.5. Statusanzeigen

2.5.1. Cluster belegt

Dieser Button gibt Auskunft darüber, ob der aktuelle Cluster belegt, unbelegt oder defekt ist. Ist der Cluster

in der FAT als defekt gekennzeichnet, ist der Text dieses Knopfes **Cluster defekt**. Handelt es sich um einen belegten Cluster, wird **Cluster belegt** invers angezeigt, bei einem unbelegten Cluster erscheint **Cluster belegt** in normaler Schrift.

Wird **Cluster belegt** angeklickt erscheint ein Dialog zum Ändern der FAT.



Anhand des Zustands der Buttons kann man sich orientieren, welcher Art der aktuelle FAT-Eintrag ist. Handelt es sich um den letzten Cluster einer Datei, wird zusätzlich zur nummerischen Angabe über den Inhalt des FAT-Eintrags der Knopf **Dateiende** invertiert. Wird **Folgecluster** aktiviert, wird nach dem Verlassen des Dialogs mit **OK** die Zahl, die als neuer Eintrag angegeben wurde, übernommen. Achten Sie beim Verlassen deshalb darauf, diesen Button bei Bedarf zu selektieren. **Leerer Cluster** markiert den angegebenen FAT-Eintrag als unbenutzt. **Defekter Cluster** trägt den Wert für einen defekten Cluster ein. **Dateiende** kennzeichnet den Eintrag mit dem Wert für das Dateiende.

Eine praktische Anwendung dieser Funktionen besteht darin, zwei Cluster eines Mediums miteinander zu verketten. Ist der Cluster, an den ein weiterer Cluster angefügt werden soll, bereits Bestandteil einer Datei, sollte der Einfachheit halber auf **Datei->Verketten** zurückgegriffen werden.

2.5.2. BPB ungültig

Der Bios Parameter Block, den DISKUS aus den Daten im Bootsektor zusammenstellt, ermöglicht logische Zugriffe auf ein Medium.

Wollen Sie mit DISKUS Disketten bearbeiten, deren Bootsektor, aus dem sich das Diskettenformat ergibt, fehlerhaft ist, empfiehlt es sich, den Knopf **BPB ungültig** zu aktivieren. Ist dies geschehen, ignoriert DISKUS die Daten, die sich im Bootsektor befinden. In der Regel erkennt DISKUS defekte Disketten automatisch und aktiviert dann **BPB ungültig** selbstständig. Ist der BPB ungültig, ist kein logischer Zugriff auf ein Medium möglich. Alle Lese- und Schreiboperationen müssen physikalisch erfolgen. Wollen Sie trotz eines fehlerhaften BPBs logisch auf eine Diskette zugreifen, hilft **Disk->Format definieren** weiter.

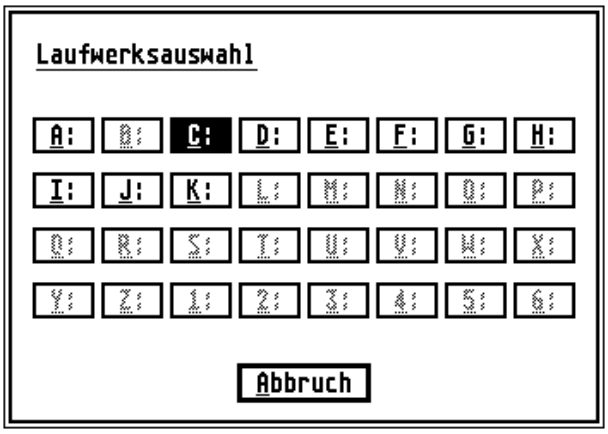
2.5.3. Schreibschutz

Durch Anklicken des **Schreibschutz**-Knopfes wird das aktuelle Laufwerk softwaremäßig schreibgeschützt. Dieser Schutz ist nur innerhalb von DISKUS wirksam.

2.6. Laufwerke

Hier lässt sich einstellen, welches Laufwerk bei den von DISKUS durchgeführten Operationen

angesprochen wird. Die Kennung U: ist für MiNT und MagiC reserviert und kann daher grundsätzlich (also auch unter TOS) nicht ausgewählt werden.



Von DISKUS werden bis zu 32 BIOS-Geräte unterstützt, sofern diese ein TOS- oder DOS-konformes FAT-Dateisystem besitzen. Laufwerke, die ausschließlich GEMDOS bekannt sind (insbesondere CD-ROM Laufwerke), werden nicht aufgeführt. Das FAT32-Dateisystem wird von DISKUS nicht unterstützt.

Werden die Pfeile links oder rechts des **Laufwerk**-Knopfes angeklickt, wechselt DISKUS zum nächsten Laufwerk vor oder zurück. Ein Wechsel des Laufwerks ist auch über die Tastatur möglich. Betätigen Sie hierzu die Tastenkombination [Shift][Alternate] zusammen mit dem Laufwerksbuchstaben.

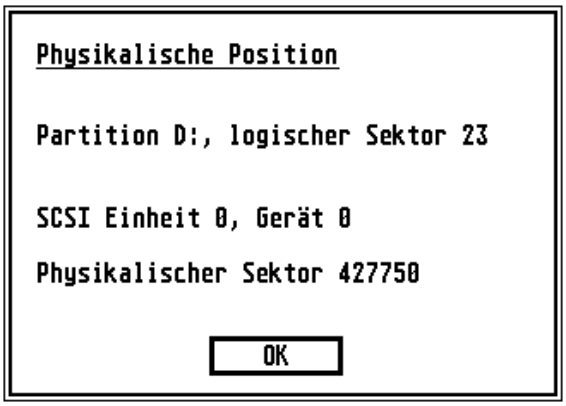
2.7. Position

Dieser Knopf führt bei Floppies zu einem Dialog, der die Umrechnung der logischen Sektornummer in physikalische Angaben über Seite, Spur und physikalischen Sektor, auf der sich der logische Sektor befindet, vornimmt. Auch die Umrechnung in die umgekehrte Richtung ist möglich.

Sollen logische in physikalische Angaben umgerechnet werden, muss nach dem Eintragen der logischen Sektornummer in das dafür vorgesehene Eingabefeld dieses Feld angeklickt werden. Zur Umwandlung physikalischer in logische Angaben muss eines der Eingabefelder mit den physikalischen Daten (Seite, Track, physikalischer Sektor) angeklickt werden, nachdem die erforderlichen Angaben in die dafür vorgesehenen Eingabefelder eingetragen wurden.

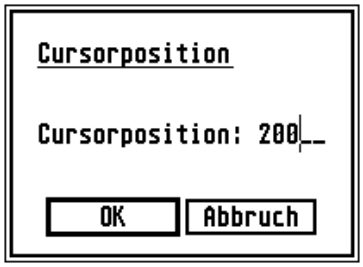
Handelt es sich beim aktuellen Laufwerk um eine Festplatten-Partition, ist der Button **Position** nur dann vorhanden, wenn ein Treiber wie HDDRIVER installiert ist, der den XHDI-Standard unterstützt. Dann wird beim Anklicken des Buttons angezeigt, welchem physikalischen Sektor der aktuelle logische Sektor

entspricht.



2.8. Die Cursorposition

Der Cursor kann innerhalb der Datenfenster mit der Maus oder über die Cursortasten an eine neue Position gesetzt werden. Soll ein bestimmtes Byte des Fensters angesprungen werden, wird das Feld mit der Anzeige der Cursorposition angeklickt. In einer Dialogbox lässt sich die neue Cursorposition angeben.

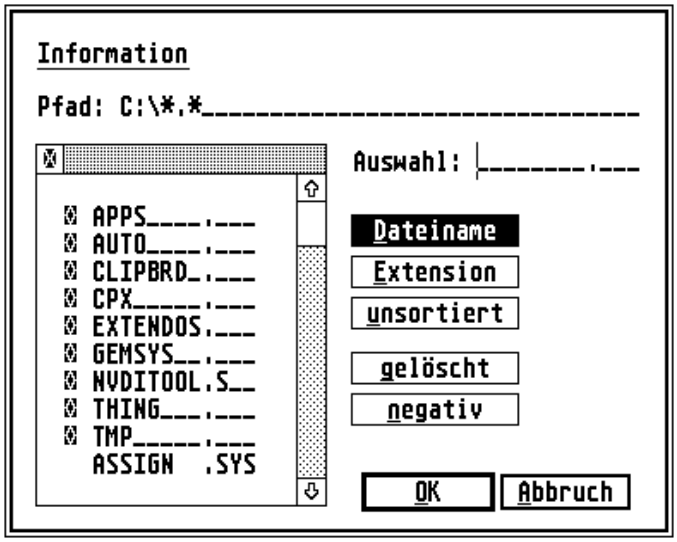


2.9. Die Laufwerkskennung

Ein Mausklick auf die Laufwerkskennung am unteren rechten Fensterrand führt zur Anzeige der Disk-Information, wie beim **Disk**-Menü beschrieben. Ein Mausklick bei gedrückter [Shift]-Taste führt zur Anzeige der Disk-Struktur, ebenfalls beschrieben beim **Disk**-Menü.

3. Die Dateiauswahlbox

Dateien auf dem aktuellen Laufwerk werden bei DISKUS nicht über die Auswahlbox des Betriebssystems ausgewählt, sondern über eine spezielle Dialogbox, die erweiterte Möglichkeiten bietet.



Ein wichtiger Unterschied zu den Ihnen bekannten Auswahlboxen besteht im zusätzlichen Knopf **gelöscht**. Ist dieser selektiert, werden nur gelöschte Dateien aufgeführt. Solche Dateien haben als erstes Zeichen ein "ó" im Dateinamen. Für DISKUS ist es wichtig, gelöschte Dateien auswählen zu können, weil insbesondere die Funktion **Datei restaurieren** hiervon Gebrauch macht.

Die Auswahlbox kann Dateien nach Dateinamen oder nach Extensions (Dateitypen) ordnen. Auch die Anzeige von Dateien in der Reihenfolge, wie sie gespeichert sind, ist möglich. Aktivieren Sie hierzu den Button **unsortiert**. Besonders praktisch ist dies, wenn man feststellen will, in welcher Reihenfolge die Programme im AUTO-Ordner angeordnet sind. Außerdem sollte man diesen Knopf aktivieren, wenn man mit der Funktion **Datei->Vertauschen** Dateien in eine neue Reihenfolge bringen will.

Sollen nur Dateien mit einer bestimmten Extension angezeigt werden, kann diese Extension am Ende des Pfadnamens eingetragen werden. Wird nun der Balken im oberen Bereich der Auswahlbox angeklickt, wird die eingetragene Extension übernommen. Dieser Balken muss auch angeklickt werden, wenn die Diskette im aktuellen Laufwerk gewechselt wurde.

Die endgültige Auswahl einer Datei oder eines Ordners geschieht durch einen Doppelklick. Dateien können auch durch einen einfachen Mausklick und anschließendes Betätigen des **OK**-Knopfes oder durch Eingabe des Dateinamens selektiert werden. Für Ordner ist jedoch ein Doppelklick notwendig, da bei einem einfachen Mausklick ein Ordner lediglich betreten, der Dialog jedoch nicht verlassen wird.

Einige Funktionen von DISKUS können auf mehrere Dateien hintereinander angewendet werden. Um mehrere Dateien auszuwählen haben Sie zwei Möglichkeiten. Zunächst können Sie mit Wildcards im Dateinamen arbeiten. Ein "?" steht für einen beliebigen Buchstaben einer Datei, ein "*" ersetzt ganze Teile eines Dateinamens. Geben Sie beispielsweise den Dateinamen "TEST?.*" an, werden von der folgenden Operation unter anderem Dateien wie "TEST0.BAS" oder auch "TEST9.S" erfasst.

Um mehrere Dateien mit völlig unterschiedlichen Namen gleichzeitig selektieren zu können, bietet DISKUS

noch eine weitere Möglichkeit. Wird beim Selektieren einer Datei die [Shift]-Taste gedrückt gehalten, können bei manchen Operationen Dateien in eine Dateiliste übernommen werden. Wird die Auswahlbox mit **OK** verlassen, wird die nachfolgende Operation bei allen selektierten Dateien und zusätzlich bei der im Eingabefeld eingetragenen Datei durchgeführt.

Im Zusammenhang mit der Möglichkeit, eine Dateiliste anzufertigen, ist der Knopf **negativ** vorhanden. Wird dieser Knopf angeklickt, wird eine Negativ-Auswahl der ausgewählten Dateien vorgenommen. Alle selektierten Dateien werden dabei deselektiert, alle deselektierten Dateien werden selektiert. Dieses Vorgehen ist praktisch, wenn eine bestimmte Operation auf nur wenige Dateien nicht angewendet werden soll. In diesem Fall selektiert man zunächst die Dateien, die nicht bearbeitet werden sollen, dann kehrt man diese Auswahl mit Hilfe des **negativ**-Knopfes um.

4. Das "Datei"-Menü

In diesem Menü befinden sich die dateiorientierten Funktionen von DISKUS. Dateizugriffe sind nur dann möglich, wenn ein Medium in einem Format vorliegt, das vom Betriebssystem erkannt wird. Die meisten Operationen können auch auf Ordner angewendet werden.

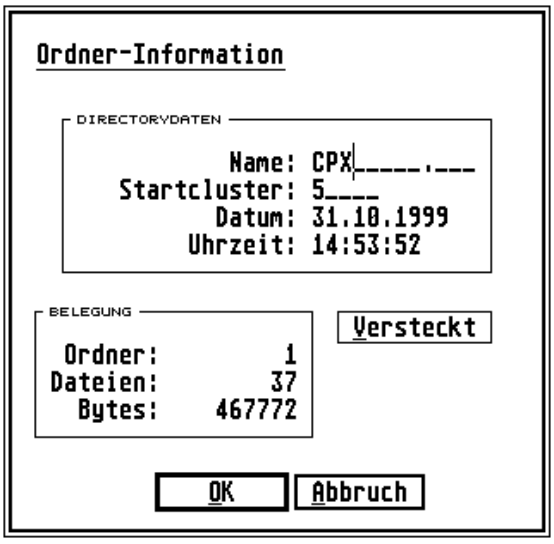
4.1. Information

Hier lassen sich diverse Daten über eine Datei abrufen.

Alle Angaben bis auf die Clusterzahl können geändert werden. Von einer Änderung der Daten für die Dateilänge oder den Startcluster ist allerdings abzuraten.

Mit **OK** sichert DISKUS die Änderungen und geht zur nächsten Datei über, falls mehrere Dateien in der Dateiauswahlbox ausgewählt wurden. **Weiter** führt zu Informationen über die nächste Datei, ohne dass Änderungen übernommen werden.

Wird per Doppelklick ein Ordner selektiert, erscheint ein anderer Dialog.



Die zur Verfügung stehenden Manipulationsmöglichkeiten sind nahezu die gleichen wie bei Dateien.

4.2. Öffnen

Diese Funktion erleichtert die Dateibearbeitung durch das Aufsuchen einer bestimmten Dateiposition.

Um eine Datei zu bearbeiten ist es nicht zwingend, die Datei vorher auf diesem Weg geöffnet zu haben.

Ist der im Datenfenster angezeigte Sektor ein Verzeichnis-Sektor, kann der erste Cluster einer in diesem Sektor aufgeführten Datei durch einen Doppelklick auf den Datei-Eintrag in die Anzeige gebracht werden.

4.3. Kopieren

Mit dieser Funktion werden Dateien kopiert. Wählen Sie hierzu die Namen der zu kopierenden Dateien in der DISKUS-eigenen Dateiauswahlbox aus. Nachdem die Dateien ausgewählt wurden, erscheint die normale Dateiauswahlbox, in der der Zielpfad für den Kopiervorgang angegeben werden muss.

Diese Kopierfunktion arbeitet clusterorientiert und vergleicht während des Kopiervorgangs die Zahl der Cluster einer Datei mit der Dateilänge. Stimmen Dateilänge und Anzahl der Cluster einer Datei nicht überein, weil die Datei fehlerhaft ist, zeigt DISKUS eine Warnung an.

Diese Kopier-Funktion kann nur auf Dateien, nicht aber auf Ordner angewendet werden.

4.4. Löschen

Das Löschen von Dateien oder Ordnern geschieht über die Dateiauswahlbox. Falls ein Ordner gelöscht werden soll, der noch Dateien enthält, erfolgt eine Nachfrage, ob die Dateien des Ordners gelöscht oder in die nächsthöhere Verzeichnis-Ebene übernommen werden sollen. So besteht die Möglichkeit, Dateien ohne Kopiervorgang aus einem Ordner zu verschieben.

4.5. Eliminieren

Beim Löschen von Dateien werden nicht die eigentlichen Daten gelöscht, sondern nur Informationen in Directory und FAT. Soll eine Datei physikalisch gelöscht, also überschrieben werden, kann das mit dieser Funktion geschehen. Eine so gelöschte Datei kann nicht mehr restauriert werden.

4.6. Verketten

Besonders beim Restaurieren einer Datei "per Hand" ist das **Verketten** hilfreich. Der aktuelle Cluster wird

mit der ausgewählten Datei verkettet. So ist es leicht möglich, beim Reparieren einer Datei die gefundenen Cluster zu einer neuen Datei zusammenzusetzen. Anhand der Zahl der beim Reparieren zusammengesetzten Cluster kann auch die ungefähre Dateilänge bestimmt und mit **Datei->Information** gesichert werden.

Handelt es sich beim angehängten Cluster um einen Cluster, der in der FAT als unbelegt vermerkt ist, wird dieser Cluster automatisch als letzter Cluster der verlängerten Datei markiert.

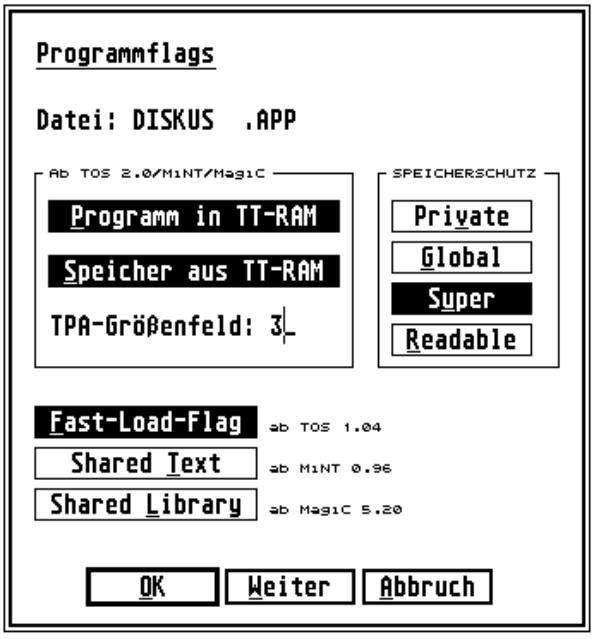
4.7. Vertauschen

Diese Funktion ermöglicht das Vertauschen der Verzeichnis-Verweise zweier Dateien. So lassen sich bestimmte Dateien am Anfang eines Verzeichnisses unterbringen, um einen möglichst schnellen Zugriff auf diese Dateien zu haben. Nützlich ist diese Funktion auch, um die Reihenfolge der Programme im AUTO-Ordner festzulegen. Die beiden zu vertauschenden Datei-Einträge werden bei gedrückter [Shift]-Taste mit der Maus ausgewählt. Es empfiehlt sich, vorher den Knopf **unsortiert** zu aktivieren, um zu sehen, wie die physikalische Reihenfolge der Dateien aussieht.

Auch **Dateien sortieren** unter **Spezial->Daten optimieren** erlaubt es, Datei-Einträge zu sortieren.

4.8. Programmflags

Jede Programmdatei enthält eine Reihe von Einstellungen, die mit DISKUS geändert werden können.



Ab TOS 1.04 existiert das sogenannte **Fast-Load-Flag**. Ist dieses gesetzt wird nach dem Laden eines Programms nur der tatsächlich benötigte Speicher gelöscht, so dass das Programm schneller gestartet wird. Manche alten Programme gehen davon aus, dass nach ihrem Start der komplette Speicher gelöscht ist. Man sollte daher ausprobieren, ob bei einem bestimmten Programm das Fast-Load-Flag gesetzt werden darf.

MiNT unterstützt sogenannte **Shared Text**-Bereiche. Das TEXT-Segment eines Programms, bei dem dieses Flag gesetzt ist, wird unter Umständen mehrfach verwendet. Wird dieses Flag bei Programmen gesetzt, die nicht darauf vorbereitet sind, kann ein Absturz die Folge sein. Nähere Informationen finden sich in der Dokumentation zu MiNT.

Seit Version 3.0 kennt TOS zwei weitere Bits im Programmheader, die festlegen, ob ein Programm Speicher aus dem TT- RAM bzw. dem virtuellen RAM nutzen darf. Wird **Programm in TT-RAM** aktiviert, wird das Programm beim Start in das TT-RAM geladen, falls dort ausreichend Speicherplatz zur Verfügung steht. Mit **Speicher aus TT-RAM** wird TOS angewiesen, einem Programm bei Speicheranforderungen möglichst Speicher aus dem TT-RAM zuzuweisen.

Das **TPA-Größenfeld** ist ebenfalls nur für Ataris mit alternativem RAM von Bedeutung. Soll ein Programm im TT-RAM laufen und ist mehr ST-RAM als TT-RAM vorhanden, wertet TOS dieses Feld aus. Es gibt an, wie viel RAM (in Blöcken von 128 KByte) zusätzlich zum beim Start reservierten Speicher zur Laufzeit benötigt wird. Stellt das TT-RAM genug Speicher zur Verfügung, wird das Programm dorthin geladen, andernfalls wird auf das ST-RAM ausgewichen. Eine 0 im Größenfeld bedeutet, dass zur Laufzeit 128+0*128 KByte Speicher benötigt werden, der Maximalwert 15 steht für 128+15*128 KByte = 2 MByte.

Die Einstellungen zum Speicherschutz sind nur für MiNT von Bedeutung. Ist dort der Speicherschutz aktiviert, können Programme lediglich auf ihre eigenen Speicherbereiche zugreifen (**Private**). Insbesondere Daten in residenten Programmen müssen manchmal jedoch auch für andere Programme zugänglich sein. Die Einstellung **Global** lässt auf ein im Speicher befindliches Programm uneingeschränkte Lese- und Schreibzugriffe zu. Wird **Super** angewählt, sind solche Zugriffe nur aus dem Supervisor-Modus heraus möglich. **Readable** sorgt schließlich dafür, dass lediglich Lesezugriffe erlaubt sind.

4.9. Eintrag löschen

Sind auf einem Medium Datenverluste aufgetreten, die durch eine defekte FAT oder ein defektes Verzeichnis hervorgerufen wurden, befinden sich in manchen Fällen Datei-Einträge in einem Verzeichnis, die keiner Datei mehr zugehörig sind. Unter Umständen lassen sich solche Dateien nicht über den Desktop entfernen. **Eintrag löschen** entfernt lediglich den Verzeichniseintrag, lässt die FAT jedoch unverändert. Nach dieser Operation sind die Cluster der Datei weiterhin als belegt gekennzeichnet. Um solche Cluster aufzufinden und zu entfernen sollte die FAT getestet werden (**Spezial->Daten testen**).

4.10. Leerer Eintrag

Mit dieser Funktion lässt sich ein Verzeichniseintrag für eine Datei oder einen Ordner erzeugen. Dies sollte z. B. vor dem Restaurieren zerstörter Dateien geschehen, falls der ursprüngliche Eintrag nicht mehr im Directory aufgeführt ist. Leere Einträge sollten im Hauptverzeichnis platziert werden, da beim Erzeugen solcher Einträge innerhalb eines Ordners nicht auszuschließen ist, dass TOS hierfür einen neuen Cluster belegt. Dies kann bei einer defekten FAT zu Datenverlusten führen.

4.11. Neuer Ordner

Diese Funktion richtet ein neues Unterverzeichnis ein.

4.12. Ende

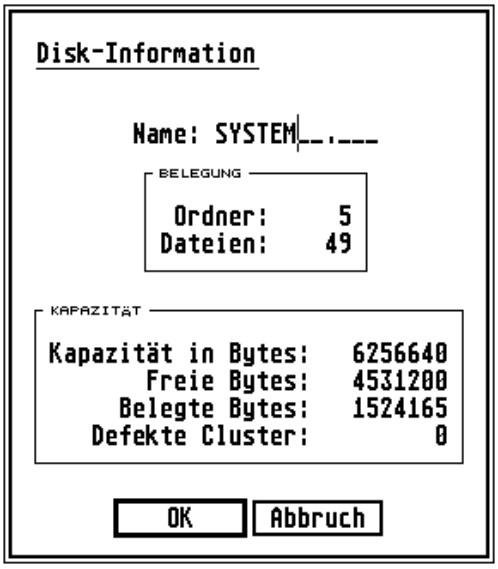
An dieser Stelle können Sie DISKUS verlassen. Änderungen, die das FAT-Format betreffen (12- oder 16-Bit-FAT) bleiben dabei erhalten.

5. Das "Disk" Menü

Dieses Menü enthält disketten- und plattenbezogene Operationen. Ein Teil dieser Funktionen kann nur auf Disketten angewendet werden. Solche Menüpunkte sind daher nicht aktiviert, wenn es sich beim aktuellen Laufwerk nicht um ein Floppylaufwerk handelt.

5.1. Information

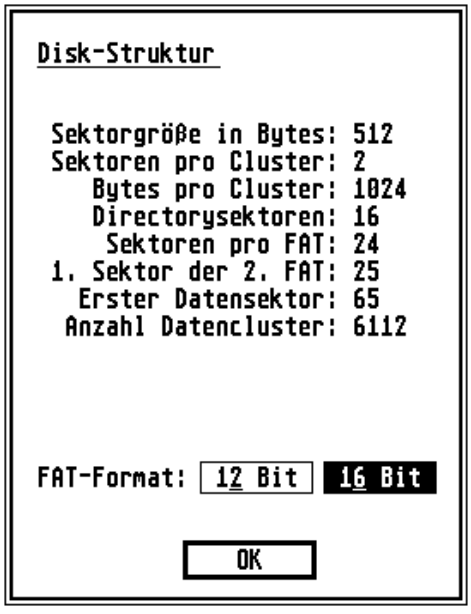
Hier wird ein Überblick über die Belegung des aktuellen Mediums angezeigt.



Die Angabe der defekten Cluster bezieht sich auf Cluster, die in der FAT als defekt markiert sind und deshalb von TOS nicht zur Datenspeicherung herangezogen werden.

5.2. Struktur

Der Bootsektor eines Mediums enthält wichtige Daten über den Aufbau eines Datenträgers. Hierzu gehören u. a. die Zahl der Sektoren und die Größe der FAT. Bei Ramdisks oder Festplatten werden einige Parameter (z. B. Sektoren pro Track) nicht angezeigt, da diesen Angaben nur bei Disketten eine Bedeutung zukommt.

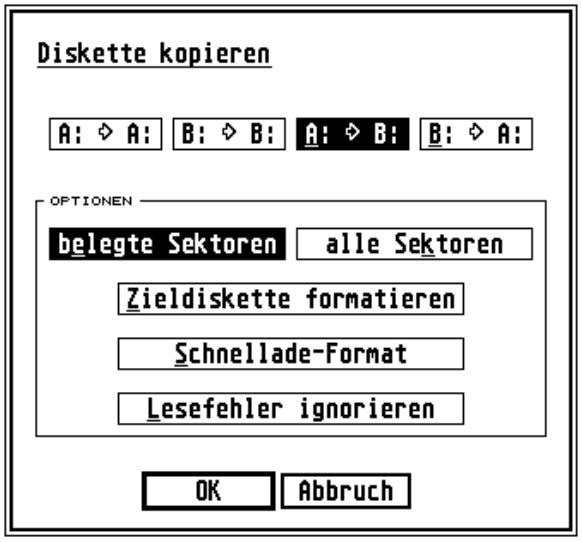


Der Zustand der Knöpfe **12-Bit-FAT** und **16-Bit-FAT** spiegelt das FAT-Format wieder. Die hier angegebene Einstellung ist dem BPB des Mediums entnommen und sollte nicht geändert werden.

Beim Einsatz von DOS-Emulatoren können auf Festplatten auch FATs auftreten, in denen jeder Eintrag eine Länge von nur 12 Bit hat. Um auf solche Partitionen zugreifen zu können, muss der Knopf **12-Bit-FAT** aktiviert werden. Das darf jedoch nur dann geschehen, wenn auch wirklich eine 12-Bit-FAT vorliegt. Dies ist nur dann der Fall, wenn eine Partition kleiner als 4 MByte ist und unter DOS eingerichtet wurde.

5.3. Kopieren

Diese Funktion erlaubt das Kopieren von Disketten.



Während des Kopiervorgangs kann die Zieldiskette formatiert werden. Falls diese bereits fehlerfrei formatiert worden ist, ist eine Neuformatierung nicht notwendig. Ist die Zieldiskette unformatiert, wird dies von DISKUS erkannt und vor dem eigentlichen Kopiervorgang wird zunächst formatiert.

Es ist möglich, nur die Sektoren kopieren zu lassen, die laut FAT belegt sind. Dies hat den Vorteil, dass der Kopiervorgang bei nicht vollständig gefüllter Diskette schneller durchgeführt wird. Dennoch wird ein Kopiervorgang mit DISKUS in der Regel langsamer ablaufen als mit speziellen Kopier-Programmen, da die Fehlerprüfungen von DISKUS Zeit brauchen. Ist der Knopf **alle Sektoren** aktiviert, werden auch die unbelegten Sektoren einer Diskette kopiert.

Sollen Lesefehler von vornherein ignoriert werden, können Sie dies mit **Lesefehler ignorieren** einstellen. Dies ist beispielsweise dann sinnvoll, wenn bekannt ist, dass mehrere Sektoren der Quelldiskette defekt sind und alle noch irgendwie lesbaren Daten auf eine andere Diskette gerettet werden sollen. Im Normalfall würde für jeden defekten Sektor eine Fehlermeldung erscheinen. Um dies zu verhindern und ein zügiges Kopieren einer defekten Diskette zu ermöglichen, kann das Ignorieren von Lesefehlern eingestellt werden. Es empfiehlt sich, bei defekten Disketten die Anzahl der Zugriffsversuche (**Spezial->Optionen**) zu erhöhen.

Nachdem eine Diskette komplett kopiert wurde, bietet DISKUS bei ausreichendem Hauptspeicher die Möglichkeit, eine weitere Diskette zu beschreiben, ohne dass die Quelldiskette erneut benötigt wird.

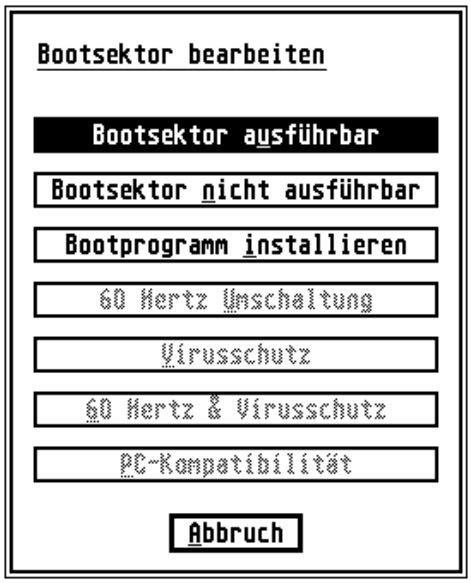
5.4. Löschen

Um die Dateien eines Mediums zu löschen ist es nicht notwendig, den Datenträger neu zu formatieren. Es genügt, das Directory und die beiden FATs zu löschen. Diese Aufgabe erfüllt das logische Löschen eines Mediums. Die Daten-Cluster bleiben dabei unberührt. Sollen zusätzlich alle Daten-Cluster mit Nullen überschrieben werden, kann physikalisches Löschen ausgewählt werden. In diesem Fall ist ein nachträgliches Rekonstruieren der gelöschten Daten in keinem Fall mehr möglich.

5.5. Bootsektor

Der Bootsektor enthält als erster Sektor eines Mediums Angaben über das logische und physikalische Format. Zusätzlich kann dort ein Programm abgelegt sein, das beim Einschalten oder nach einem Reset ausgeführt wird, sofern sich die Diskette zu diesem Zeitpunkt in Floppy A: befindet.

Nach dem Betreten der Dialogbox lässt sich anhand des Zustands der beiden oberen Knöpfe erkennen, ob der Bootsektor des aktuellen Mediums ausführbar ist.



Um die Ausführbarkeit des Bootsektors zu ändern verlassen Sie die Dialogbox einfach über den entsprechenden Knopf.

Um einen ausführbaren Bootsektor zu erzeugen, der ein vorgegebenes Programm enthält, wählen Sie **Bootprogramm installieren**. Ein solches Programm muss positionsunabhängig programmiert sein und darf kein BSS-Segment besitzen. Bei der Installation entfernt DISKUS automatisch den Programmheader und der Bootsektor wird als ausführbar markiert. Die Programmlänge kann höchstens 452 Bytes betragen, da sie durch die Zahl der im Bootsektor reservierten Bytes bestimmt ist.

60 Hertz-Umschaltung installiert ein Programm im Bootsektor, das während des Bootens eine Umschaltung der Bildfrequenz von 50 auf 60 Hertz vornimmt. Dies ist für die Benutzer eines Farbmonitors oder Fernsehers interessant, da eine höhere Frequenz die Bildqualität verbessert. Allerdings kommt nicht jeder Monitor mit 60 Hertz klar, darum empfiehlt es sich, dieses Bootprogramm bei Bedarf probeweise zu installieren. Ist kein Betrieb mit 60 Hertz Bildfrequenz möglich, kann das Bootprogramm durch **Bootsektor nicht ausführbar** wieder deaktiviert werden.

Mit **Virusschutz** wird ein Programm installiert, das verhindert, dass Viren unbemerkt auf den Bootsektor geschrieben werden. Das Programm zeigt die Meldung "Der Bootsektor dieser Diskette enthält keinen Virus!" an, wenn die Diskette bei einem Reset oder beim Einschalten des Rechners in Floppy A: liegt. So lange diese Meldung erscheint, kann sich kein Virus im Bootsektor eingenistet haben. **60 Hertz & Virusschutz** installiert schließlich beide oben genannten Bootprogramme. Da ein Virusschutz für den Bootsektor nur bei Disketten, nicht aber bei Festplatten sinnvoll ist, können die entsprechen Buttons nur ausgewählt werden, wenn es sich beim aktuellen Laufwerk um eine Floppy handelt.

Disketten, die mit TOS 1.04 oder einer neueren Version formatiert wurden, sind bereits von vornherein PC-kompatibel. Wurde mit einer älteren TOS-Version formatiert, müssen jedoch Daten auf der Diskette korrigiert werden. Wollen Sie eine Diskette für den PC anpassen, erledigt das der Knopf **PC-Kompatibilität**. Alle Daten auf dem Datenträger bleiben dabei erhalten. Solche Disketten können jedoch bei den TOS-Versionen 1.00 und 1.02 keinen ausführbaren Bootsektor besitzen. Disketten, die mit DISKUS formatiert wurden, brauchen nicht nachträglich an das PC-Format angepasst werden.

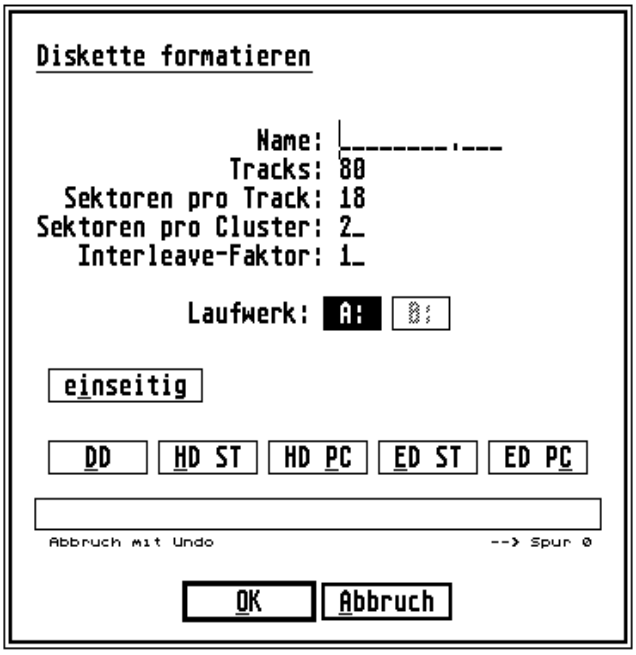
5.6. Format definieren

Es kann vorkommen, dass eine Diskette ausgerechnet auf dem Bootsektor schadhaft wird. Dann ist es für TOS unmöglich, dieses Medium zu benutzen. Um dennoch an die restlichen Daten zu gelangen, vorausgesetzt, sie sind einigermaßen fehlerfrei, kann man DISKUS mitteilen, welches Format eine Diskette besitzt. Die angegebenen Daten werden dazu verwendet, einen neuen BPB zu erzeugen und so Zugriffe wieder zu ermöglichen.

In der Dialogbox ist zunächst das bisherige Format des aktuellen Mediums eingetragen. Soll das neu definierte Format als Grundlage für einen neuen Bootsektor verwendet, also auf die Diskette geschrieben werden, muss die Dialogbox mit **Schreiben** verlassen werden.

5.7. Formatieren

Beim Formatieren von Disketten mit DISKUS sind Formate möglich, die von den Standardformaten mit 80 Spuren zu 9, 18 bzw. 36 Sektoren abweichen.



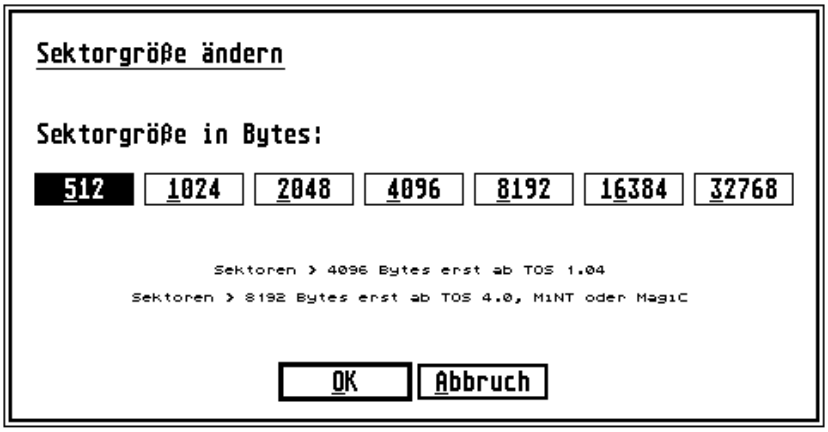
DISKUS bietet mehrere vordefinierte Formate an. Andere lassen sich durch Editieren der Eingabedaten erzeugen. Ein Doppelklick auf einen der fünf Format-Buttons sorgt dafür, dass die aktuellen Formatangaben die bisher über diesen Button zugänglichen Daten ersetzen. Um die Kapazität einer Diskette zu erhöhen lassen sich DD-Disketten mit mehr als 80 Tracks zu jeweils 10 Sektoren formatieren. Mehr als 87 Spuren sind jedoch keinesfalls möglich. Die meisten Diskettenlaufwerke erlauben ohnehin höchstens 82 Tracks. Falls Sie diese Grenze überschreiten, sind Datenverluste als Folge nicht auszuschließen!

Bei DD-Disketten besteht ein Cluster in der Regel aus zwei Sektoren. Mehr als zwei Sektoren pro Cluster werden von TOS nicht unterstützt. Werden HD- oder ED-Disketten formatiert, sollte die Zahl der Sektoren pro Cluster auf 1 gesetzt werden, wenn man diese Disketten auf DOS-kompatiblen Systemen einsetzen will.

Die von DISKUS formatierten Disketten können auch von PCs verwendet werden. PC-Kompatibilität kann man nachträglich mit **PC-Kompatibilität** unter **Disk->Bootsektor** erzielen.

5.8. Sektorgröße ändern

Mit dieser Experten-Funktion lässt sich die logische Sektorgröße eines Mediums ändern.



Dies kann in speziellen Fällen sinnvoll sein, wobei die Limitationen zu beachten sind, die das Betriebssystem im Hinblick auf die unterstützte Sektorgröße aufweist.

6. Das "Harddisk"-Menü

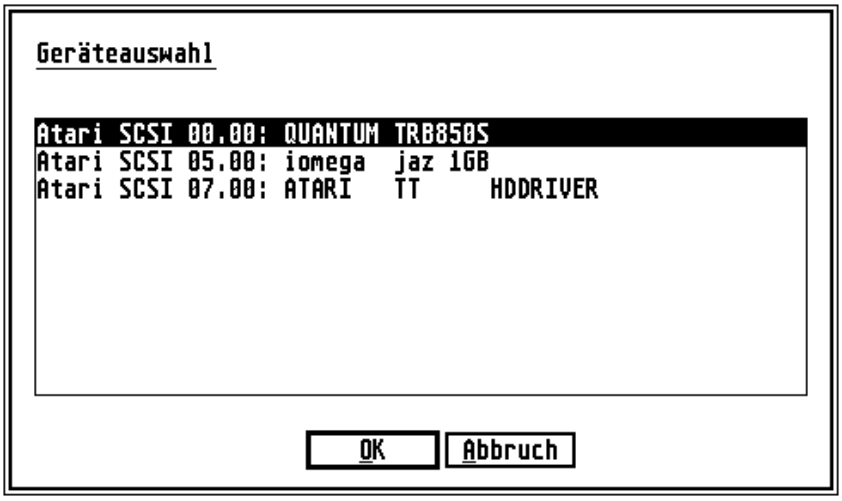
Sofern ein SCSI-Treiber installiert ist, lassen sich spezielle Funktionen von DISKUS auf beliebige Peripheriegeräte anwenden. Der in HDDRIVER integrierte SCSI-Treiber erlaubt es auf dem Atari und dem Milan, IDE-, ATAPI- und SATA-Geräte wie SCSI-Geräte zu behandeln. Daher ist die Installation von HDDRIVER (und sei es nur der Demoversion) erforderlich, um über die Funktionen des **Harddisk**-Menüs Zugriff auf IDE- und SATA-Peripherie zu haben. Mit Software (z. B. HDDRIVER-Modulen) von Drittanbietern ist so sogar ein Zugriff auf USB-Massenspeicher möglich.

6.1. Busse neu scannen

Alle Busse werden neu nach Geräten durchsucht. Diese Funktion ist hilfreich, wenn ein Gerät nachträglich eingeschaltet wurde und mit den Funktionen des **“Harddisk“**-Menüs angesprochen werden soll.

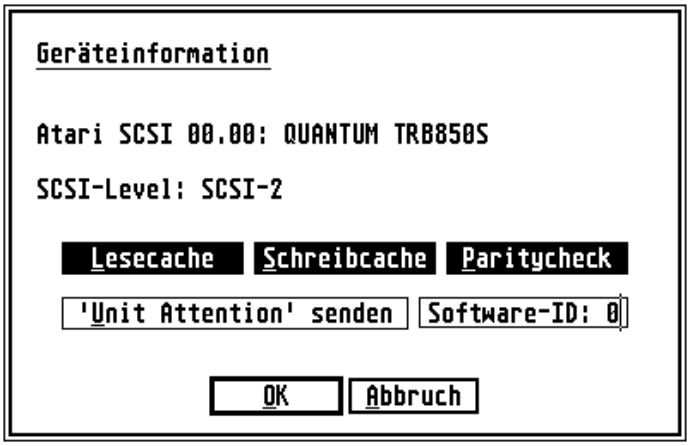
6.2. Geräteauswahl

An dieser Stelle erfolgt die Auswahl des Geräts, auf das sich die nachfolgenden Funktionen beziehen. Es werden alle Busse bzw. Geräte aufgelistet, für die ein SCSI-Treiber existiert.



6.3. Geräteinformation

Dieser Punkt liefert Informationen über einige Parameter eines Peripheriegeräts.

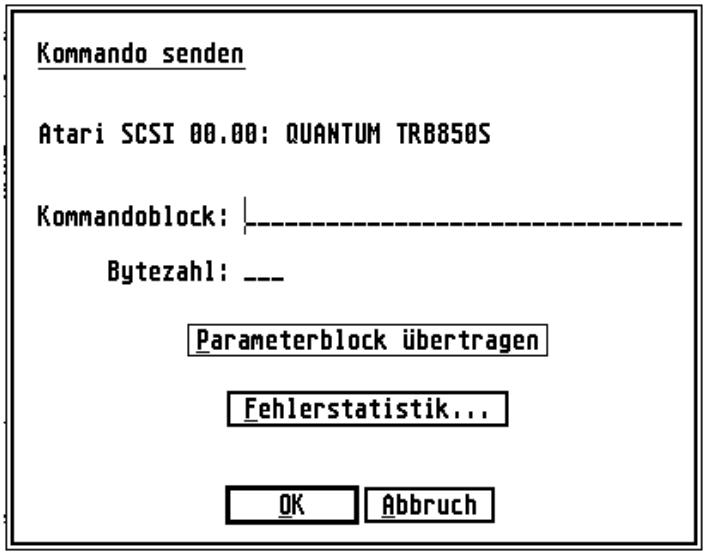


Bei Platten, die mit einem eigenen Cache ausgestattet sind, lässt sich dieser ein- oder ausschalten. Bei

einigen Platten von Syquest und Quantum ist konfigurierbar, ob die Platte nach einem Reset ein "Unit Attention"-Signal an den Rechner schicken soll. Dies ist abzuschalten, wenn eine Platte an einem Atari mit einer alten TOS-Version betrieben wird, da andernfalls nicht von dieser Platte gebootet werden kann. Bei manchen Quantum-Festplatten lässt sich per Software einstellen, ob die Platte mit Paritycheck arbeiten soll. Platten am ACSI-Bus können mit vielen SCSI-Hostadaptern nur bei abgeschalteter Paritätsprüfung betrieben werden. Schließlich lässt sich noch bei einigen Platten von Quantum die SCSI-ID per Software ändern.

6.4. Kommando senden

Mit dieser Experten-Funktion lassen sich spezielle Fähigkeiten von Peripheriegeräten nutzen.



Es lassen sich beliebige SCSI-Kommandos verschicken. Dies ist möglich für alle Geräte, für die ein SCSI-Treiber existiert, also auch für IDE- oder USB-Peripherie. In das Eingabefeld **Befehlsblock** werden die bis zu 16 Bytes eines Befehls eingetragen. Für die ACSI-Schnittstelle des ST sind ohne ICD-kompatiblen Hostadapter wie den LINK96/97 oder Lösungen wie SCSI2Pi, PiSCSI oder BlueSCSI nur Befehle der SCSI-Klasse 0 möglich.

Zusätzlich zu den Befehls-Bytes können Parameter in Form eines vorher markierten Blocks gesendet werden, wenn **Parameterblock** aktiviert wird. Wie viele Bytes im Rahmen eines Kommandos übertragen werden sollen, wird unter **Bytezahl** angegeben. Diese Information muss der tatsächlich übertragenen Zahl der Bytes entsprechen, da es andernfalls je nach installiertem SCSI-Treiber zu Fehlern kommen kann. Werden Daten vom Gerät geliefert, werden diese im Datenfenster dargestellt und als Block markiert.

Die Controller einiger alter SCSI-Platten führen eine Statistik über alle Such- und Leseoperationen sowie über aufgetretene Fehler. Mit **Fehlerstatistik** werden diese Daten anzeigt. Falls eine Platte keine Fehlerstatistik führt, meldet DISKUS dies in Form eines SCSI-Fehlers #32 (Ungültiges Kommando).

Tritt ein korrigierbarer Lesefehler stets an derselben Stelle auf, ist damit zu rechnen, dass sich dieser zu einem nicht korrigierbaren Fehler auswachsen wird.

6.5. Rootsektor

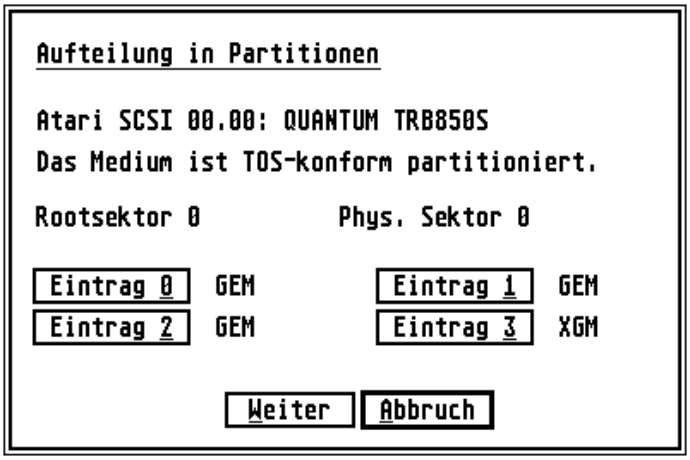
Diese Funktion erlaubt Operationen am Rootsektor der Festplatte. Dieser Sektor enthält wichtige Daten über die Aufteilung einer Platte in Partitionen. Ferner kann er ein Programm enthalten, das nach einem Reset automatisch ausgeführt wird. Es empfiehlt sich, den Rootsektor aus Gründen der Datensicherheit auf einem anderen Medium mit **Spezial->Daten speichern** zu sichern. So hat man im Fehlerfall die Möglichkeit, die Daten des Rootsektors leicht zu rekonstruieren.

Ähnlich wie beim Bootsektor lässt sich auch beim Rootsektor die Ausführbarkeit ändern. Der momentane Status entspricht dem Zustand der Buttons **Rootsektor ausführbar** bzw. **Rootsektor nicht ausführbar**.

Mit **Bootprogramm installieren** kann ein geeignetes Programm im Rootsektor der Festplatte installiert werden, das beim Booten automatisch ausgeführt wird.

6.6. Partitionen

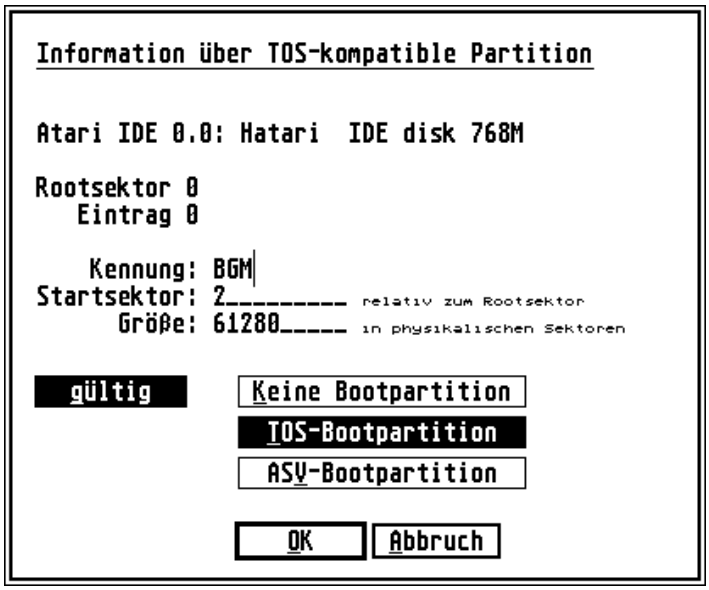
Diese Funktion liefert genaue Angaben über eine bestimmte Partition. Es erscheint zunächst eine Dialogbox, die die Nummer des aktuellen Rootsektors und die in diesem Rootsektor vorhandenen Partitionen anzeigt.



Bei Atari-kompatiblen Formaten verweisen GEM- und BGM-Einträge auf "normale" Partitionen, ein XGM-Eintrag enthält einen Verweis auf einen weiteren Rootsektor. Durch Anklicken von **Weiter** wird die Aufteilung des nächsten Rootsektors angezeigt, falls ein weiterer Rootsektor vorhanden ist.

Platten, die PC-konform partitioniert sind, weisen eine ähnliche Struktur auf. Hier besitzt eine erweiterte Partition (vergleichbar mit XGM) die Kennung $05 oder $0F. Partitionen, die kleiner als 4 MByte sind und eine 12-Bit-FAT aufweisen besitzen die Kennung $01, Partitionen kleiner 32 MByte mit 16-Bit-FAT die Kennung $04. Solche Partitionen können mit einem Festplattentreiber wie HDDRIVER auch unter TOS direkt genutzt werden und eignen sich daher zum Datenaustausch zwischen Ataris und PCs. PC-konforme Partitionen mit einer Kapazität von mehr als 32 MByte besitzen die Kennungen $06/$0E oder $0B/ $0C (FAT32) und sind nicht von vornherein TOS-kompatibel. Sie können jedoch zusammen mit HDDRIVER und MiNT oder MagiC genutzt werden.

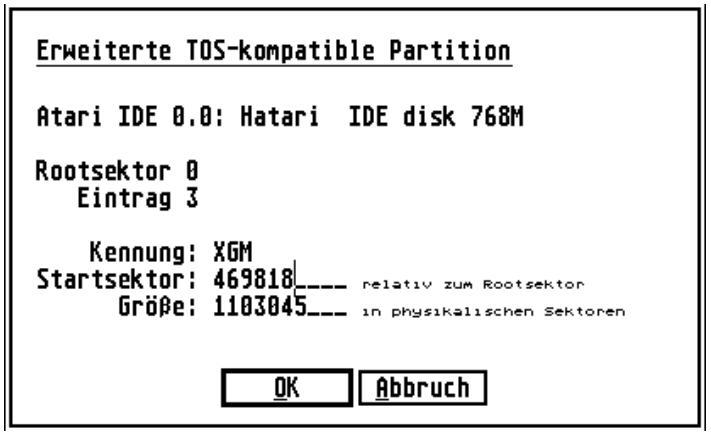
Nähere Informationen über eine bestimmte Partition erhält man nach Anwählen des entsprechenden Knopfes. Wird beispielsweise eine GEM- oder BGM-Partition ausgewählt, erscheint dieser Dialog:



Bei unsinnigen Daten für den Startsektor oder die Größe zeigt DISKUS im entsprechenden Eingabefeld eine Null an. Dies kann dann der Fall sein, wenn die Daten einer ungenutzten Partition ungültig sind. Die Angabe des Startsektors einer Partition ist stets relativ zum vorhergehenden Rootsektor zu verstehen.

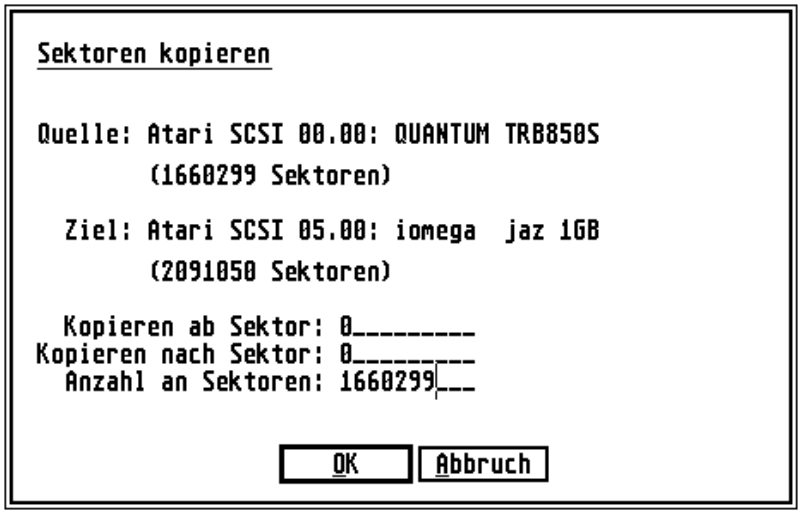
Soll eine Partition nicht unter TOS genutzt werden, kann die Kennung der Partition geändert werden. Ist der Knopf **TOS-Boot** aktiv, kann TOS von dieser Partition gebootet werden. Der Knopf **gültig** bestimmt, ob die Partition überhaupt Verwendung findet.

Informationen über einen XGM-Eintrag werden wie folgt dargestellt:



6.7. Sektoren kopieren

Mit dieser Funktion wird ein physikalischer Bereich einer Platte auf ein anderes Medium kopiert.



Beim Quelllaufwerk handelt es sich um das Gerät, das bei der Geräteauswahl angegeben wurde. Das Zielgerät wird ebenfalls über die Geräteauswahl selektiert. Für den Kopiervorgang lassen sich der Startsektor auf dem Quellmedium, der Zielsektor auf dem Zielmedium und die Zahl der zu kopierenden Sektoren angeben. Soll eine bestimmte Partition kopiert werden, lassen sich Startsektor und Größe der Partition vorher mit **Harddisk->Partitionen** ermitteln.

6.8. Partition initialisieren

Diese Funktion führt eine Art Formatierung nur für eine bestimmte Partition durch. Beim Initialisieren einer Partition wird der Bootsektor neu angelegt, wobei die im Rootsektor vermerkten Partitionsdaten zugrunde gelegt werden. Die Daten auf der Partition gehen dabei in der Regel verloren, es sei denn, es handelt sich um eine TOS-kompatible Partition mit denselben Bootsektor-Parametern, wie sie nach der Initialisierung vorliegen. Bis auf den Bootsektor verändert **Partition initialisieren** keine weiteren Daten. Soll die neu

initialisierte Partition unter TOS genutzt werden und sind die ursprünglichen Daten nicht weiter von Bedeutung oder handelte es sich ursprünglich nicht um eine TOS-kompatible Partition, muss die Partition noch mit **Disk->Löschen** logisch gelöscht werden.

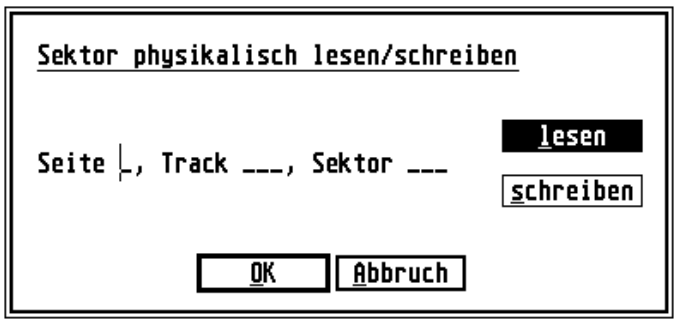
Die HDDRIVER-Software bietet erweiterte Möglichkeiten zum Initialisieren von Partitionen.

7. Das "Sektor"-Menü

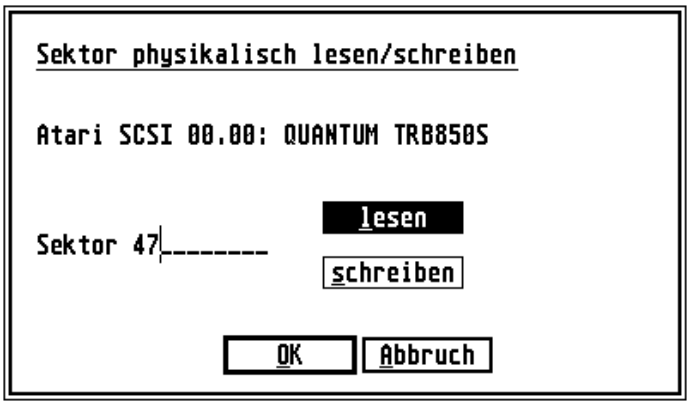
Dieses Menü enthält spezielle sektor-bezogene Operationen. Die meisten sektor-bezogenen Operationen sind über den Desktop zugänglich.

7.1. Physikalisch lesen/schreiben

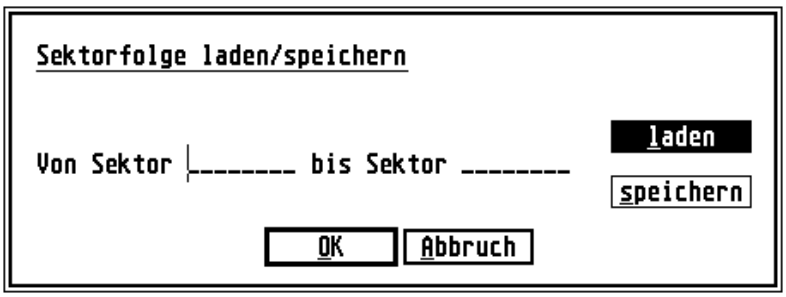
Physikalische Zugriffe auf eine Diskette erfolgen unter Angabe der Seite, Spur und Nummer eines Sektors.



Ist ein SCSI-Treiber installiert lassen sich beide Funktionen auch auf Festplatten anwenden.



7.2. Sektorfolge laden/speichern



Mit diesen Funktionen können aufeinander folgende Sektoren in eine Datei mit der Extension DAT geschrieben bzw. mit dem Inhalt einer Datei gefüllt werden. Wird die zweite Sektornummer nicht angegeben, wird so lange gelesen bzw. geschrieben, bis das Dateiende oder der letzte Sektor des Mediums erreicht ist.

8. Das "Block"-Menü

Dieses Menü enthält blockorientierte Funktionen, die sich auf einen internen Puffer oder auf das GEM-Klemmbrett beziehen. Das Markieren eines Blocks geschieht per Maus. Wird ein Byte im Datenfenster bei gedrückter [Shift]-Taste angeklickt, wird der Bereich zwischen dieser Position und dem Cursor markiert.

8.1. Alles markieren

Alle Daten im Datenfenster werden als Block markiert.

8.2. Ausschneiden

Der markierte Block wird in den internen Zwischenspeicher bzw. ins Klemmbrett verschoben. Die restlichen Bytes des Datenfensters werden dabei nach vorne aufgerückt und am Ende der Daten werden Nullbytes eingefügt.

8.3. Kopieren

Der markierte Block wird in den Zwischenspeicher oder das Klemmbrett kopiert.

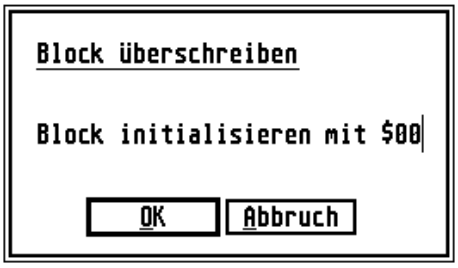
8.4. Einfügen

Mit dieser Funktion wird ein Block im Datenfenster an der Cursorposition eingefügt. Dabei werden die bisher an dieser Stelle vorhandenen Bytes überschrieben.

Wird ein Block über das Klemmbrett eingelesen, berücksichtigt DISKUS Dateien mit den Extensions DAT und TXT in der angegebenen Reihenfolge.

8.5. Überschreiben

Ein markierter Bereich lässt sich mit einem bestimmten Byte initialisieren. Geben Sie in der Dialogbox den Wert des Bytes an, das hierzu benutzt werden soll.



9. Das "Spezial"-Menü

In diesem Menü sind einige nützliche Funktionen zusammengefasst. Zu solchen zählen beispielsweise das Optimieren von Datenträgern sowie das Schützen oder Testen von Daten.

9.1. Daten laden/speichern

Soll ein Block oder die im Datenfenster angezeigte Information als Datei gesichert werden, geschieht das mit dieser Funktion. Die Datei sollte die Extension DAT erhalten. Wurde kein Block markiert, speichert DISKUS alle im aktuellen Fenster enthaltenen Daten. Andernfalls wird der Block gespeichert. Wird beim Speichern ein bereits vorhandener Dateiname angegeben, können die Daten an die bestehende Datei angehängt werden.

Mit **Daten laden** werden die so gesicherten Daten zurück in das Datenfenster geholt. Das erste Zeichen der Datei wird an der aktuellen Cursorposition eingefügt.

9.2. Daten drucken

Der Inhalt des aktuellen Fensters oder eines Blocks wird gedruckt. Falls die Ausgabe in eine Datei erfolgen soll, kann dies bei den Programmoptionen eingestellt werden.

9.3. Daten suchen

Dieser Menüpunkt erlaubt das Suchen nach einer Zeichenkette oder nach hexadezimalen Daten.

Ist das erste Zeichen im Eingabefeld ein "$" geht DISKUS davon aus, dass es sich bei den Daten im Eingabefeld nicht um eine Zeichenkette sondern um eine hexadezimale Angabe handelt. Soll beim Suchen nach einer Zeichenkette nicht zwischen Groß- und Kleinschreibung unterschieden werden, muss der Knopf **Groß-/Kleinschreibung** deaktiviert werden. Es lässt sich ein Zeichen angeben, das als Wildcard dient.

Soll eine bestimmte Datei durchsucht werden, muss der Knopf **in Datei** angeklickt werden. Die Datensektoren des aktuellen Laufwerks werden unabhängig von Dateigrenzen durchsucht, wenn **auf Disk** angewählt wird. Angefangen beim aktuellen Sektor werden alle noch folgenden Datensektoren berücksichtigt. Der Knopf **auf Harddisk** kann nur ausgewählt werden, wenn ein SCSI-Treiber verfügbar ist und es sich beim ausgewählten Gerät um eine Festplatte oder ein vergleichbares Gerät handelt. Gesucht wird ab dem aktuellen physikalischen Sektor.

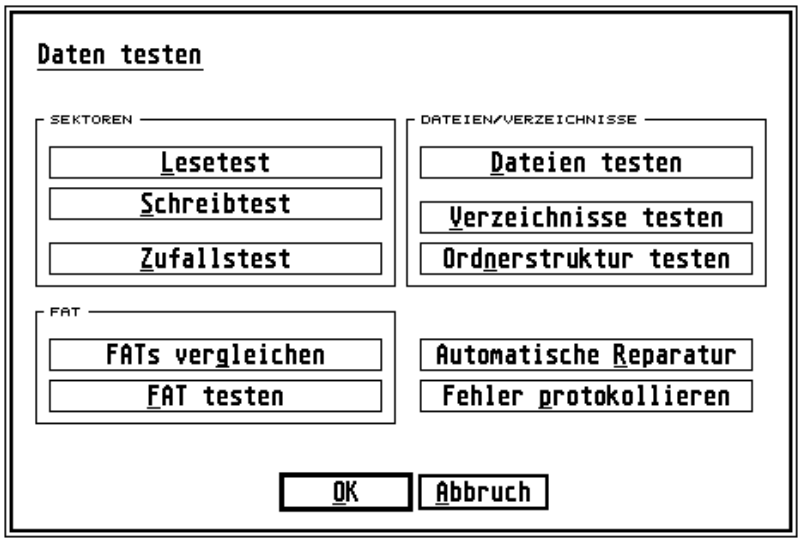
Wird die Zeichenkette gefunden, wird der Sektor, bei dem die Suche Erfolg hatte, im Datenfenster dargestellt. Der Cursor befindet sich dabei auf dem ersten Byte der Zeichenkette.

9.4. Weitersuchen

Eine Suchoperation wird mit dieser Funktion fortgesetzt.

9.5. Daten testen

Um Datenverlusten vorzubeugen bietet DISKUS an, einen Datenträger gründlich zu testen.



Um einen kompletten Datenträger auf Fehler zu testen, ist die Funktion **Lesetest** vorhanden. Es werden alle Sektoren gelesen, angefangen beim aktuellen Sektor. In der FAT als defekt markierte Sektoren werden übersprungen. Jeder Sektor wird zweimal gelesen und die Sektorinhalte werden verglichen.

Mit der soeben beschriebenen Testfunktion werden alle Sektoren eines Mediums mit steigender Sektornummer getestet. Beim **Zufallstest** werden die Sektoren eines Mediums in zufälliger Reihenfolge gelesen. Somit werden eventuell Fehler erkannt, die beim Ansprechen aufeinander folgender Sektoren nicht gefunden werden. **Zufallstest** hat nur dann eine Wirkung, wenn gleichzeitig einer der Knöpfe **Lesetest** oder **Schreibtest** aktiviert wird. Der Zufallstest muss mit der [Esc]-Taste abgebrochen werden.

Selbst wenn das fehlerfreie Lesen einzelner Sektoren möglich ist, können beim Schreiben dennoch Fehler auftreten. Um zu testen, ob neben fehlerfreien Lesezugriffen auch korrekte Schreibzugriffe möglich sind, steht der **Schreibtest** zur Verfügung. Hierbei liest DISKUS zunächst einen Sektor ein und versucht, den Inhalt des gelesenen Sektors zurückzuschreiben. Ist dies fehlerfrei möglich, wird der Sektor erneut gelesen und die Daten werden verglichen. Beim **Schreibtest** wird der Inhalt jedes Sektors vor dem Überschreiben in einem Puffer untergebracht und kann mit **Sektor zurückholen** unter **Spezial->Daten retten** nach einem Schreibfehler in die Anzeige gebracht werden.

Bestehen Unterschiede zwischen beiden FATs, wie sie insbesondere durch einen Fehler beim Schreiben hervorgerufen werden können, wird dies durch **FATs vergleichen** festgestellt. DISKUS informiert über den ersten FAT-Eintrag, bei dem ein Unterschied auftritt und gibt die Möglichkeit, die FATs einander anzugleichen. Dabei ist zu beachten, dass FAT 2 die Grundlage für Dateizugriffe des Betriebssystems

darstellt. Es empfiehlt sich deshalb, vor einem Angleichen der FATs zunächst wichtige Daten anderweitig zu sichern, es sei denn, man ist sich sicher, welche FAT die fehlerhafte ist.

Beim Testen der FAT werden folgende Fehlerzustände erkannt:

- Ein Cluster zeigt auf einen nicht existierenden Cluster.
- Ein Cluster-Eintrag ist fehlerhaft.
- Ein Cluster ist belegt, obwohl er zu keiner Datei gehört.
- Zwei Einträge verweisen auf den gleichen Cluster.
- Ein Cluster-Eintrag zeigt auf sich selbst.

Belegte Cluster, die zu keiner Datei gehören, stellen in der Regel durch einen Fehler hervorgerufene Dateileichen dar, die Speicherplatz verbrauchen, von TOS jedoch nicht verwendet werden. Solche Cluster können zu einer Datei mit dem Namen "!!!!LOST.DAT" zusammengefasst werden. Anschließend kann man sich diese Datei mit Hilfe eines Editors ansehen, um festzustellen, ob es sich um Daten handelt, die noch von Bedeutung sind. Ist dies nicht der Fall, können die nicht vom TOS verwendeten Cluster entfernt werden, indem die Datei, die ja nun diese Cluster repräsentiert, gelöscht wird.

Zeigen zwei Einträge auf denselben Cluster, führt dies im weiteren Testverlauf auf jeden Fall dazu, dass unbenutzte Cluster gemeldet werden. Hierbei handelt es sich lediglich um einen Folgefehler.

Neben dem Testen der FAT lassen sich alle von Dateien belegten Sektoren überprüfen (**Dateien testen**).

Beim Testen von Dateien werden neben Lesefehlern folgende Fehler erkannt:

- Die Dateilänge stimmt nicht mit der Anzahl der Cluster überein.
- Ein Cluster-Eintrag zeigt auf einen unbelegten Cluster.
- Ein Eintrag zeigt auf einen defekten Cluster.

Bei einem Lesefehler wird der entsprechende Cluster, falls er nicht belegt ist, in der FAT automatisch als defekt markiert. Handelt es sich bei einem defekten Sektor um einen belegten Sektor, lässt sich der zugehörige fehlerhafte Cluster in der FAT als defekt vermerken. Ferner kann DISKUS in diesem Fall die restlichen Cluster der Datei neu verketten, damit die Datei weiterhin von TOS gelesen werden kann. Die Daten des fehlerhaften Clusters gehen dabei natürlich verloren und die Dateilänge stimmt nicht mehr mit der Clusterzahl überein. Die Dateilänge kann mit **Datei- >Information** korrigiert werden.

Es kann im Fehlerfall vorkommen, dass derselbe Dateiname in einem Verzeichnis zweimal auftaucht. **Verzeichnisse testen** erkennt solche Inkonsistenzen und weist darauf hin. Darüber hinaus erkennt diese Funktion Dateien, die denselben Startcluster aufweisen. Zur Fehlerbehebung sollte man mit **Datei->Eintrag löschen** einen der betroffenen Datei-Einträge entfernen.

Mit **Ordnerstruktur testen** lassen sich die Verwaltungseinträge "." und ".." von Ordnern prüfen. Findet DISKUS bei diesem Test einen Fehler, lassen sich die Verwaltungseinträge des fehlerhaften Ordners neu erzeugen, um so den Fehler zu beseitigen.

DISKUS bietet beim Testen von Daten die Möglichkeit, die FAT automatisch anzupassen (**Automatische Reparatur**). In diesem Fall erscheint beim Auftreten eines Lesefehlers bei belegten Daten-Clustern keine Alarmbox, sondern DISKUS markiert den defekten Cluster selbstständig als fehlerhaft und bringt die Dateiverkettung wieder in Ordnung. Besonders dann, wenn viele Sektoren defekt sind, erspart man es sich so, ständig Fehlermeldungen zu bestätigen. Am Ende des Tests erhält man eine Meldung über die Zahl der defekten Sektoren und hat nun die Möglichkeit, abzubrechen, oder die betroffenen Cluster endgültig als defekt markieren zu lassen, wobei die FAT zurückgeschrieben wird.

In der Regel empfiehlt es sich jedoch, bei defekten Sektoren die Reparatur per Hand vorzunehmen, um eine bessere Kontrolle über die notwendigen FAT-Änderungen zu haben.

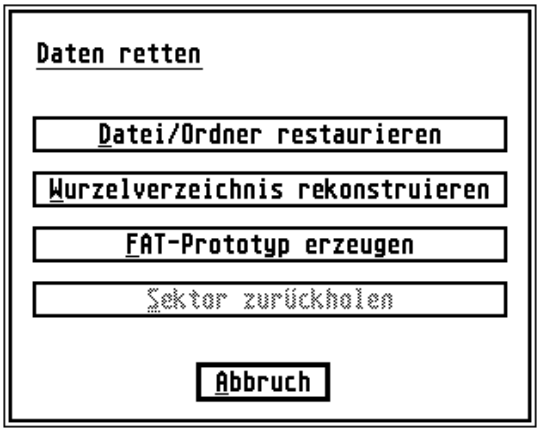
Wird die automatische Reparatur in Verbindung mit dem Testen von Dateien eingesetzt, werden folgende Fehler behoben:

- Hat eine Datei mehr Cluster als der Dateilänge entspricht, wird aus der Dateilänge die Nummer des letzten Clusters ermittelt. Dieser wird dann als solcher markiert.
- Hat eine Datei weniger Cluster als der Dateilänge entspricht, wird aus der Clusterzahl die Dateilänge ermittelt und der Dateieintrag korrigiert.
- Zeigt ein Datencluster auf den Startcluster einer Datei, wird dieser Datencluster als letzter Cluster der zugehörigen Datei markiert.

Soll DISKUS Fehler beim Testen von Daten nicht auf dem Bildschirm anzeigen, sondern in eine Protokolldatei schreiben, muss der Button **Fehler protokollieren** aktiviert werden. Die Protokolldatei sollte nicht auf dem Laufwerk angelegt werden, dessen Daten getestet werden. Andernfalls können bereits beschädigte Daten durch Schreibzugriffe auf die Protokolldatei weiter in Mitleidenschaft gezogen werden.

9.6. Daten retten

Je nach Art eines Datenverlustes kann DISKUS verlorene Daten automatisch rekonstruieren.



9.6.1. Datei/Ordner restaurieren

Wurde eine Datei oder ein Ordner irrtümlich gelöscht, kann die gelöschte Datei mit Hilfe dieser Funktion eventuell wiederhergestellt werden. Allerdings dürfen nach dem Löschen der Datei keine Schreibzugriffe

stattgefunden haben, da diese bereits einen Teil der Daten zerstört haben könnten.

Nach dem Aufruf dieser Funktion befinden Sie sich in der Dateiauswahlbox. Der Knopf **gelöscht** ermöglicht es, nur gelöschte Dateien anzusehen. Haben Sie das Verzeichnis gefunden, in dem sich die gelöschte Datei befindet, sollte dieser Knopf aktiviert werden. Lassen Sie sich nicht dadurch irritieren, dass die Dateinamen gelöschter Dateien mit dem "ó"-Zeichen beginnen. Mit diesem Zeichen kennzeichnet TOS solche Dateien.

Nach dem Auswählen der Dateien versucht DISKUS, die ursprünglichen Cluster der gelöschten Datei zu finden und miteinander zu verbinden. Dazu werden, beginnend mit dem Startcluster, alle folgenden unbelegten Cluster verkettet, bis die ursprüngliche Dateilänge erreicht ist. Anschließend muss noch mit **Datei->Information** das erste Zeichen des Dateinamens korrigiert werden, damit TOS die restaurierte Datei wieder im Verzeichnis anzeigt. Nach dieser Operation sollten Sie die Datei öffnen und in ihr herumblättern. Bei einer Textdatei kann man so meist gut erkennen, ob die Reparatur erfolgreich war. In diesem Fall muss die FAT zunächst zurückgeschrieben werden (**Spezial->FATs schreiben**).

Ist das automatische Restaurieren einer gelöschten Datei nicht möglich, hilft nur ein Restaurieren "per Hand". Mehr dazu später in diesem Handbuch.

9.6.2. Wurzelverzeichnis rekonstruieren

Wurde das Wurzelverzeichnis ganz oder teilweise zerstört, kann DISKUS die Directory-Daten rekonstruieren. Voraussetzung hierfür ist, dass die FAT des betroffenen Mediums noch intakt ist. Anhand der FAT ist es möglich, Anzahl, Startcluster und ungefähre (maximale) Länge der verlorenen Dateien zu ermitteln und so ein neues Directory zu erzeugen. Die ursprünglichen Dateinamen des Wurzelverzeichnisses können allerdings nicht mehr ermittelt werden. Alle gefundenen Dateien werden deshalb beginnend mit "FILE0" durchnummeriert. Ordner erhalten Bezeichnungen wie "FOLDER0".

Die genaue Länge der Dateien im Hauptverzeichnis kann ebenfalls nicht mehr ermittelt werden, da sie ausschließlich im Directory vermerkt ist. DISKUS trägt deshalb Vielfache der Clustergröße als Dateilänge ein. So werden auf jeden Fall alle Bytes der Datei erfasst. Es kann jedoch sein, dass die Datei ursprünglich einige Bytes kürzer war. Bei Textdateien kann man dieses Problem einfach beheben, indem man die überflüssigen Bytes am Textende mit Hilfe eines Texteditors entfernt. Bei Programmdateien fallen zusätzliche Bytes nicht weiter ins Gewicht.

Beim Rekonstruieren ist es nicht möglich, eindeutig zu erkennen, welchem Dateityp die Dateien ursprünglich angehört haben. Da Programmdateien in der Regel in den ersten beiden Bytes den Wert $601A enthalten, verleiht DISKUS solchen Dateien automatisch die Extension "PRG". Es kann sich aber auch um Accessories oder Objektdateien handeln. Hier muss man durch das Starten solcher Dateien testen, um welchen Programmtyp es sich tatsächlich handelt. Textdateien können leicht erkannt werden, wenn eine rekonstruierte Datei mit **Datei->Öffnen** oder in einem Texteditor angeschaut wird.

Nachdem die Funktion **Directory rekonstruieren** aufgerufen wurde, durchsucht DISKUS zunächst die FAT und meldet anschließend, wie viele Ordner und Dateien (für das Hauptverzeichnis, Dateien in Ordnern werden nicht aufgeführt) gefunden wurden. Nun besteht die Möglichkeit, ein neues Directory für diese

Dateien schreiben zu lassen. Werden mehr Dateien gefunden, als Platz für Directory-Einträge existiert, gibt es eine Warnung. Es werden dann nur so viele Dateien im Wurzelverzeichnis eingetragen, wie dieses zulässt. Ist die FAT des Mediums fehlerfrei kann dieser Fall nicht eintreten.

Um im obigen Fall auch die Dateien, deren Einträge nicht mehr im Hauptverzeichnis untergebracht werden konnten, rekonstruieren zu können, müssen die bisher geretteten Dateien zunächst auf einem anderen Datenträger untergebracht werden. Anschließend kann man alle bis zu diesem Zeitpunkt im Hauptverzeichnis aufgeführten Dateien löschen und erneut das Directory rekonstruieren lassen. Nun werden die Dateien im Wurzelverzeichnis vermerkt, für die vorher kein Platz mehr vorhanden war.

Die Funktion **Directory rekonstruieren** kann auch dann hilfreich sein, wenn Dateien innerhalb eines Ordners verschwunden sind. Diese Dateien finden sich nach dem Erzeugen eines neuen Directories im Hauptverzeichnis wieder.

9.6.3. FAT-Prototyp erzeugen

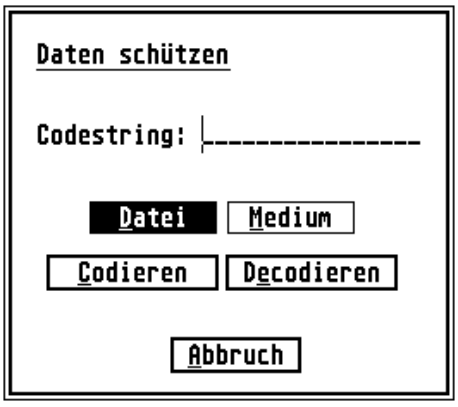
Ist das Hauptverzeichnis noch intakt, aber die FAT zerstört, kann diese Funktion zur Datenrettung eingesetzt werden. Es wird zunächst eine FAT erzeugt, bei der jeder Eintrag auf den nächsten zeigt. So erhält man eine FAT-Struktur, die alle Datencluster zu einer einzigen Datei zusammenfasst. Ruft man nun **Dateien testen** unter **Spezial->Daten testen** auf und stellt **Automatische Reparatur** ein, korrigiert DISKUS den FAT-Prototyp gemäß den im Hauptverzeichnis und den Ordnern eingetragenen Dateilängen. Anschließend ist ein gewisser Teil der Daten wieder hergestellt, je nachdem, wie stark das Medium zum Zeitpunkt des Datenverlustes fragmentiert war. Eventuell ist noch Handarbeit angesagt, die aufgrund der teilweise rekonstruierten FAT erheblich einfacher ist als bei gelöschter FAT.

9.6.4. Sektor zurückholen

Bevor DISKUS einzelne Sektoren auf ein Medium schreibt, wird der alte Sektorinhalt in einen Puffer eingelesen. Mit **Sektor zurückholen** wird der ursprüngliche Inhalt des Sektors in die Anzeige zurückgeholt.

9.7. Daten schützen

Enthält eine Datei oder Diskette Daten, die anderen verborgen bleiben sollen, bietet diese Funktion die Möglichkeit, diese Daten zu codieren und wieder zu decodieren.



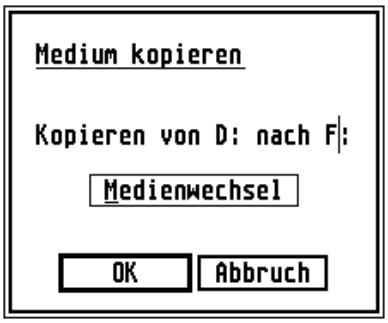
Als Grundlage für die Codierung (und auch für die Decodierung) dient eine Zeichenkette. Ein

Wiederherstellen der codierten Daten ist nur mit dem korrekten Codestring möglich. Beim Codieren von Dateien werden die Sektoren der neuen Datei an die gleichen Positionen geschrieben, an denen sich vorher die Daten der alten Datei befunden haben, so dass sich anschließend kein Teil der Datei im nicht codierten Zustand auf dem Medium befindet.

Wird ein Ordner codiert, werden nur die Daten des Ordners an sich codiert, nicht jedoch jede einzelne Datei. Codierte Ordner dürfen nicht über den Desktop geöffnet werden, da dies einen Absturz nach sich ziehen kann. Darüber hinaus dürfen in codierte Ordner keine Dateien kopiert werden, da dies einen Datenverlust zur Folge haben kann.

9.8. Medium kopieren

An dieser Stelle ist es möglich, beliebige Speichermedien zu kopieren. So kann beispielsweise eine Partition auf eine andere Partition (Backup!) kopiert werden, eine doppelseitige Diskette kann auf eine einseitige, eine Ramdisk auf eine Diskette kopiert werden usw.



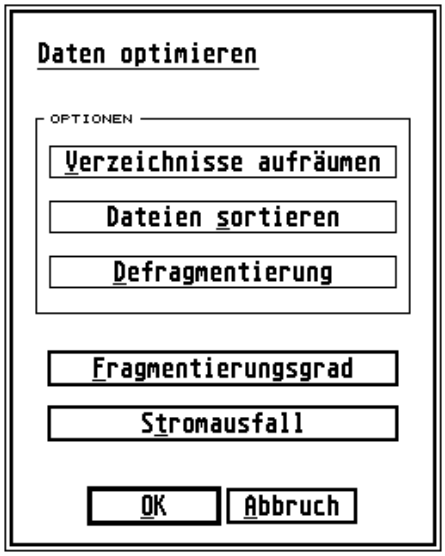
Jede Kombination ist möglich, jedoch sollte die Kapazität des Ziellaufwerks für die Kopier-Aktion ausreichen. Der Kopiervorgang geschieht nicht datei- sondern clusterweise und ist deshalb in kurzer Zeit erledigt. Die Cluster- und Sektorgrößen auf Ziel- und Quellmedium müssen identisch sein.

Kopiert werden das Directory, die FATs und so viele Datencluster, wie das Ziellaufwerk Platz bietet. Die Daten auf dem Ziellaufwerk werden überschrieben. Reicht die Kapazität des Ziellaufwerks für den Kopiervorgang nicht aus, besteht die Möglichkeit zum Abbruch oder aber es werden so viele Daten wie möglich kopiert.

Um das Kopieren zwischen zwei Wechselmedien zu unterstützen, steht der Knopf **Medienwechsel** zur Verfügung. Wird er aktiviert, wird vor jedem Lese- bzw. Schreibvorgang zum Medienwechsel aufgefordert.

9.9. Daten optimieren

Wurde mit einem Medium längere Zeit intensiv gearbeitet, ist ein Optimieren sinnvoll. Dies bringt Zeitvorteile beim Datenzugriff und erleichtert es, eine irrtümlich gelöschte Datei zu restaurieren.



DISKUS bietet zum Optimieren von Medien mehrere Funktionen, wobei **Verzeichnisse aufräumen** und **Dateien sortieren** nicht auf VFAT-Partitionen mit langen Dateinamen angewendet werden dürfen.

9.9.1. Verzeichnisse aufräumen

Nach dem Löschen von Dateien befinden sich zwischen den Einträgen der verbliebenen Dateien meist einige von inzwischen gelöschten Daten. TOS braucht dann für die Suche nach Dateien unnötig viel Zeit, da auch die gelöschten Einträge durchsucht werden. Beim Aufräumen der Verzeichnisse werden diese entfernt.

9.9.2. Dateien sortieren

Das Sortieren von Dateien nach Dateitypen ermöglicht es, auf bestimmte Dateien möglichst schnell zuzugreifen. Werden beispielsweise alle Programmdateien als erste Dateien in einem Verzeichnis untergebracht, werden diese beim Ladevorgang schneller gefunden und daher schneller geladen.

DISKUS kann Dateien innerhalb eines Verzeichnisses in eine neue Reihenfolge bringen. Hierzu werden die Extensions, die vom Sortieren erfasst werden sollen, in das entsprechende Eingabefeld eingetragen. Die so gebildete Reihenfolge der Dateitypen wird als Kriterium für das Sortieren verwendet.

Es kann sinnvoll sein, Ordner als erste Einträge in einem Verzeichnis unterzubringen. Dazu muss der Knopf **Ordner am Anfang** aktiviert werden. Beim Sortieren besteht die Möglichkeit, das Hauptverzeichnis oder einen bestimmten Ordner sortieren zu lassen. Sind beide Knöpfe aktiviert, werden alle Dateien in jedem Verzeichnis sortiert. Ist nur der **Ordner**-Knopf aktiv, lässt sich ein Ordner für das Sortieren auswählen.

Eine Besonderheit gilt für den AUTO-Ordner. Dieser enthält Programme in der Reihenfolge, wie sie nach dem Einschalten oder nach einem Reset gestartet werden sollen. Oft ist diese Abfolge wichtig und darf nicht verändert werden. Um ein Sortieren der Dateien des AUTO-Ordners zu verhindern, existiert der Knopf **AUTO-Ordner nicht sortieren**.

Um Datei-Einträge gezielt neu zu sortieren steht die Funktion **Datei->Vertauschen** zur Verfügung.

9.9.3. Defragmentierung

Durch wiederholtes Speichern und Löschen von Dateien werden die Cluster neu angelegter Dateien auf dem Medium verstreut. Dies hat zur Folge, dass Dateizugriffe unnötig viel Zeit in Anspruch nehmen. DISKUS kann die Cluster aller Dateien neu ordnen, so dass danach wieder optimale Zugriffszeiten gegeben sind. Darüber hinaus ist es im Falle einer irrtümlich gelöschten Datei umso wahrscheinlicher, die verlorenen Daten zurückgewinnen zu können, je weniger fragmentiert ein Medium ist.

Vor der Defragmentierung wird automatisch ein Test der FAT durchgeführt. Selbst bei einem Stromausfall während der Defragmentierung kann es nicht zu empfindlichen Datenverlusten kommen. Bevor nach einem Stromausfall weiter mit der betroffenen Diskette bzw. Partition gearbeitet werden kann, muss jedoch unbedingt der Knopf **Stromausfall** angewählt werden.

Um sich Gewissheit darüber zu verschaffen, ob ein Defragmentieren lohnenswert ist, kann **Fragmentierung** ausgewählt werden. Man erhält Angaben über die Zahl der unfragmentierten und fragmentierten Dateien. Außerdem wird angezeigt, wie stark sich die Fragmentierung insgesamt auswirkt.

9.10. FATs schreiben

Einige Operationen von DISKUS ändern die FAT des aktuellen Laufwerks. Solche Änderungen werden jedoch nicht sofort auf das Medium geschrieben, sondern erst dann, wenn Sie dies ausdrücklich wünschen. Beachten Sie, dass die FAT Informationen enthält, die entscheidend dafür sind, ob die Dateien eines Mediums korrekt gelesen werden können.

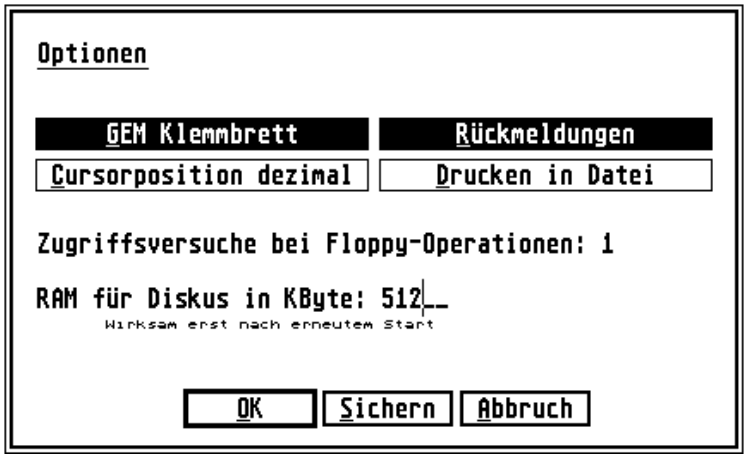
9.11. Umrechnung

Die Benutzung von Zahlen im hexadezimalen Format ist bei der Arbeit mit DISKUS hin und wieder

unumgänglich. Tragen Sie zur Umrechnung den vorliegenden Wert in das entsprechende Eingabefeld ein. Ein Doppelklick auf das Eingabefeld führt zur Umrechnung in das jeweils andere Zahlenformat.

9.12. Optionen

An dieser Stelle lässt sich DISKUS konfigurieren.



Das Ausschneiden oder Kopieren von Blöcken kann nicht nur in einen DISKUS-internen Zwischenspeicher, sondern auch auf das Klemmbrett erfolgen. Hierzu muss **GEM Klemmbrett** aktiviert werden.

DISKUS bringt nach dem Abschluss einiger Operationen Meldungen auch dann, wenn kein Fehler aufgetreten ist. Hier wird also der fehlerfreie Verlauf angezeigt. Solche Meldungen können jedoch störend sein und den Arbeitsfluss hemmen. Daher kann man diese Form der **Rückmeldungen** abschalten.

DISKUS zeigt die Cursorposition normalerweise im hexadezimalen Zahlenformat an. Optional ist mit **Cursorposition dezimal** auch eine dezimale Anzeige möglich.

Wird **Drucken in Datei** selektiert, werden alle Druckerausgaben in eine Datei umgelenkt. Der Name dieser Datei kann vor Beginn jedes Druck-Vorgangs in der Dateiauswahlbox angegeben werden.

Bei schlechtem Diskettenmaterial kann es nützlich sein, die Zahl der **Zugriffsversuche für Floppy-Operationen** heraufzusetzen.

Wird DISKUS in einem Multitasking-System eingesetzt, sollte der benutzte Speicher beschränkt werden, damit auch anderen Programmen Speicher zur Verfügung steht. **RAM für DISKUS in KByte** legt fest, wie viel RAM für Datenpuffer DISKUS maximal belegen soll. Wird eine Null eingetragen benutzt DISKUS den kompletten verfügbaren Speicher. Die aktuellen Multitasking-Systeme bieten vergleichbare Einstellungen von sich aus an, so dass die Speicherzuteilung dort auch außerhalb von DISKUS geregelt werden kann.

Beim Sichern der Optionen werden neben den meisten Einstellungen aus den Dialogboxen auch der Desktop-Aufbau sowie Fensterpositionen in einer Datei mit dem Namen "DISKUS.INF" abgelegt. Beim Programmstart wird diese Datei geladen. Dabei sucht DISKUS zunächst nach einer Umgebungsvariablen mit dem Namen "DISKUS". Der Inhalt dieser Variablen gibt den Pfad an, in dem DISKUS nach "DISKUS.INF“ sucht. Ist die Variable nicht gesetzt, wird "DISKUS.INF" auf dem aktuellen Laufwerk gesucht. Um das Laden der Einstellungen zu verhindern muss während des Programmstarts eine [Shift]-Taste gedrückt werden.

10. Hinweise zur Tastaturbelegung

Buttons wie **Abbruch** oder **Nein** können stets über die [Undo]-Taste bedient werden. [Undo] macht außerdem Änderungen rückgängig, die nach dem Lesen eines Sektors im Datenfenster vorgenommen wurden.

Mit [Control][W] kann zwischen dem Sektorfenster und dem FAT-Fenster gewechselt werden. Beide Fenster lassen sich mit [Control][U] schließen.

Um den Inhalt eines Sektors oder Clusters vorübergehend zu speichern, stellt DISKUS bis zu zehn Puffer bereit. Das Speichern der im Datenfenster sichtbaren Sektordaten erfolgt über die Tastenkombinationen [Control][1] bis [Control][0]. Die Pufferinhalte werden über [Alternate][1] bis [Alternate][0] abgerufen, wobei die bisherigen Daten des Fensters überschrieben werden. Steht nicht genügend Speicherplatz zur Verfügung, um alle Puffer einzurichten, werden weniger als 10 Puffer aktiviert.

Wird die Diskette im aktuellen Laufwerk gewechselt, muss DISKUS hiervon durch Drücken der [Esc]-Taste unterrichtet werden.

Mit der [Tab]-Taste ist es möglich, zwischen einem Byte in der hexadezimalen Anzeige und der ASCII-Anzeige umzuschalten.

Die Tasten des Cursorblocks sind mit Sonderfunktionen belegt: Mit der [Home]-Taste wird das erste Byte der Anzeige angesprungen, mit [Shift][Home] wird der Cursor auf das erste Byte der Daten gesetzt. Die Cursortasten erlauben das Setzen des Cursors auf ein beliebiges Bytes innerhalb des Datenfensters. Wird zusätzlich zu den Cursortasten eine [Shift]-Taste betätigt, werden die Daten im Fenster zeilenweise aufwärts bzw. abwärts gescrollt. Seitenweises Scrollen erreicht man über die [Control]-Taste zusammen mit den Cursortasten. Die [Help]-Taste bietet eine Übersicht über einige wichtige, nicht offensichtliche Tastenbelegungen. In Verbindung mit der [Shift]-Taste ermöglicht es die [Help]-Taste, die zehn zuletzt gelesenen Sektoren bzw. Cluster erneut einzulesen. Dabei wird bei der ersten Benutzung von [Shift][Help] der zuletzt gelesene Sektor angezeigt, dann der vorletzte Sektor usw. Besonders praktisch ist diese Funktion, wenn man sich durch mehrere Doppelklicks auf Directory-Einträge in eine Ordnerstruktur hinein bewegt hat, und nun wieder "hinaus“ will.

Auch einige Tasten des Ziffernblocks haben spezielle Belegungen. Falls Sie hexadezimale Eingaben machen, bei denen neben den Ziffern von 0 bis 9 auch die Buchstaben A bis F benötigt werden, können Sie hierzu die Tasten [(], [)], [/], [*], [-] und [+] des Ziffernblocks für die Buchstaben A bis F benutzen.

11. Besonderheiten bei MiNT und MagiC

DISKUS läuft ohne Einschränkungen unter N.AES, MiNT und MagiC und bietet unter diesen Multitasking-Systemen zusätzliche Leistungsmerkmale. Einige davon sind direkt für den Anwender sichtbar. Manche Funktionen stehen nur auf bestimmten dieser Systeme bereit.

MiNT und MagiC unterstützen zusätzlich zu den Laufwerken A: bis P: auch die Kennungen Q: bis Z:, MiNT und Big-DOS darüber hinaus noch 1: bis 6:. Diese Partitionen werden nur dann erkannt, wenn ein geeigneter Festplattentreiber wie HDDRIVER installiert ist. Auch DISKUS unterstützt alle Kennungen von A: bis Z: sowie 1: bis 6:. Laufwerk U: wird als reserviert betrachtet, da es unter MiNT und MagiC eine Sonderrolle einnimmt. DISKUS kann daher grundsätzlich nicht auf Laufwerk U: zugreifen.

12. Der SCSI-Treiber

Um möglichst flexibel auf Peripheriegeräte zugreifen zu können, benutzt DISKUS für physikalische Zugriffe auf Peripheriegeräte sowie für weitere spezielle Operationen die SCSI-Treiber-Schnittstelle. Hierbei handelt es sich um eine Software-Schnittstelle, für die Treiber für alle Ataris und einige Clones und Emulatoren verfügbar sind. Der Vorteil dieser Lösung ist, dass DISKUS keiner Anpassung an spezielle Hardware bedarf, um direkten Zugriff auf neue Peripheriegeräte oder neue Busse zu erhalten. Voraussetzung ist lediglich, dass ein SCSI-Treiber für diese Hardware verfügbar ist. So ist beispielsweise auch der Zugriff auf USB-Massenspeicher möglich.

Für Ataris und kompatible Computer gibt es mehrere SCSI-Treiber, u. a. CBHD/SCSIDRV, der in HDDRIVER integrierte SCSI-Treiber, sowie den SCSI-Treiber für den Milan. Für Ataris mit USB-Hardware-Erweiterung, für die Atari-Emulatoren Hatari und ARAnyM sowie MagiCMac und MagiCPC sind ebenfalls SCSI-Treiber verfügbar. Mit dem SCSI-Treiber von HDDRIVER lässt sich am Atari nicht nur SCSI-Peripherie ansteuern, sondern auch ATAPI-, SATA- und IDE-Peripherie. Der SCSI-Treiber für den Milan schließlich wurde speziell für den PCI SCSI-Bus des Milan entwickelt. Weitere Informationen zu SCSI-Treibern liefert die HDDRIVER-Website auf https://www.hddriver.net.

Um alle Funktionen des **“Harddisk“**-Menüs von DISKUS nutzen zu können, ist die Installation eines SCSI-Treibers erforderlich. Im Idealfall kommt HDDRIVER zum Einsatz, auf dem Milan zusammen mit dem SCSI-Treiber für den Milan.

13. Daten auf Disketten, Festplatten und Speicherkarten

DISKUS ist nicht nur eine umfangreiche Sammlung von Routinen zur Bearbeitung von Platten, Disketten und Dateien. Der Anwender soll mit Hilfe dieses Handbuchs und den nachfolgenden Informationen einen Einstieg in das Thema der Datenorganisation auf Disketten, Festplatten und Speicherkarten finden können.

13.1. Welcher Diskettentyp ist geeignet?

Je nach Modell Ihres Atari kommen unterschiedliche Diskettentypen für die Datenspeicherung in Betracht. Beim Atari ST erfolgt die Datenspeicherung in der Regel auf Disketten doppelter Dichte (DD). Unter Verwendung spezieller Hardware-Schaltungen können auch HD/ED-Laufwerke und somit HD/ED-Disketten benutzt werden. Der Falcon030 sowie viele TTs und Clones besitzen ein HD-Laufwerk. Bei TTs, die ohne HD-Laufwerk ausgeliefert wurden, lässt sich ein HD-Laufwerk nachrüsten.

13.2. Das physikalische Diskettenformat

Eine formatierte Diskette besteht aus einzelnen konzentrischen Spuren (Tracks), insgesamt 80. Die Nummerierung dieser Spuren beginnt bei 0, so dass eine Diskette die Spuren 0 bis 79 besitzt. Jede Spur ist in Sektoren zu 512 Bytes unterteilt, beim Atari normalerweise 9 Sektoren pro Spur bei DD-Disketten. Bei HD-Disketten liegen meist 18 Sektoren pro Spur vor, bei ED-Disketten 36 Sektoren.

Festplatten und Wechselplatten sind prinzipiell ähnlich wie Disketten aufgebaut, auch hier gibt es also Spuren und Sektoren. Die Sektorgröße von Festplatten beträgt in der Regel 512 Bytes. Bei Wechselmedien hoher Kapazität sind - wie bei DVD-RAM-Medien und Daten-CDs - auch 2048 Bytes gebräuchlich.

13.3. Das logische Diskettenformat

Um sich die Verwaltung von Daten zu erleichtern werden von TOS in der Regel zwei Sektoren zu einer Einheit (Cluster) zusammengefasst. Jede Datei, sei sie auch noch so klein, belegt mindestens einen Cluster. Eine Datei mit einer Länge von 1025 Bytes benötigt bei einer Clustergröße von 1024 Bytes bereits 2 Cluster. Der zweite Cluster wird für Byte 1025 benötigt, die restlichen Bytes des Clusters bleiben ungenutzt.

In der Diskbelegungstabelle (File Allocation Table, FAT) wird für jede Diskette oder Partition vermerkt, welche Cluster bereits belegt sind, und welche noch zur Speicherung neuer Daten benutzt werden können. Die FAT befindet sich zwar in doppelter Ausführung auf dem Medium, aber bei allen Dateioperationen wird stets auf FAT 2 zurückgegriffen. FAT 1 wird beim Schreiben neuer Daten aktualisiert, bleibt aber ansonsten ungenutzt. DISKUS erlaubt es, für alle Operationen wahlweise FAT 1 oder FAT 2 als Grundlage zu nehmen. Dies ist besonders dann nützlich, wenn ein Sektor der zweiten FAT defekt ist.

Schließlich muss noch vermerkt werden, welche Dateien wo auf einem Medium beginnen. Dies geschieht im Wurzelverzeichnis und den Ordnern, die zu jeder Datei so wichtige Angaben wie Dateilänge, Dateiname usw. enthalten. Nur wenn sowohl die FAT als auch alle Verzeichnisse korrekte Daten enthalten, kann eine Datei gelesen werden. Wie die FATs und Verzeichnisse aufgebaut sind wird weiter unten beschrieben.

13.4. Der Bootsektor

Neben den Sektoren, die von den FATs und dem Wurzelverzeichnis belegt werden, existiert noch ein weiterer Sektor, der für das Betriebssystem reserviert ist: Der Bootsektor. Er enthält Informationen über das physikalische und logische Format eines Mediums, ohne die es nicht möglich ist, eine Diskette zur Datenspeicherung zu verwenden. Der Bootsektor ist der erste Sektor einer Diskette oder Partition.

Wichtig für TOS sind die ersten 30 Bytes des Bootsektors mit Angaben, die das Medienformat beschreiben. Diese Informationen werden durch Byte- und Wortkonstanten bestimmt. Worte sind im Intel-Format abgelegt, wie es von PCs benutzt wird. Die folgende Tabelle gibt einen Überblick über die Bootsektor-Daten. Beim Atari je nach Diskettentyp übliche Werte sind in Klammern angefügt.

Byte	***Name***	***Bedeutung***
$00-$01		Verzweigungsbefehl auf das Bootprogramm ($6038, $E900)
$02-$07		Reserviert
$08-$0A		24-Bit umfassende Seriennummer des Mediums
$0B-$0C	BPS	Bytes pro Sektor (512)
$0D	SPC	Sektoren pro Cluster (1, 2)
$0E-$0F	RES	Reservierte Sektoren (1)
$10	FAT	Zahl der FATs (2)
$11-$12	DIR	Zahl der Datei-Einträge im Wurzelverzeichnis (144, 256)
$13-$14	SEC	Gesamtzahl der Sektoren (720, 1440, 2880, 5760)
$15	MEDIA	Media-Descriptor-Byte ($F9, $F8, $F0)
$16-$17	SPF	Sektoren pro FAT (5, 6, 9)
$18-$19	SPT	Sektoren pro Track (9, 18, 36)
$1A-$1B	SIDE	Anzahl der Seiten/Köpfe (1, 2)
$1C-$1D	HID	Versteckte Sektoren (0)

Man unterscheidet zwei Typen von Bootsektoren: ausführbare und nicht ausführbare. Ein ausführbarer Bootsektor enthält neben den aufgeführten Daten ein kleines Programm, das beim Einschalten oder nach einem Reset gestartet wird, wenn eine Diskette mit ausführbarem Bootsektor in Laufwerk A: liegt. Ob ein Bootsektor ausführbar ist wird durch die Summe aller Worte des Bootsektors bestimmt. Ergibt diese Summe den Wert $1234, handelt es sich um einen ausführbaren Bootsektor. Damit die Möglichkeit besteht, für jeden Bootsektor eine solche Prüfsumme zu erzeugen, ist das letzte Wort des Sektors reserviert. Unabhängig vom restlichen Inhalt des Bootsektors kann durch einen passenden Wert stets eine Sektorprüfsumme von $1234 erhalten werden. Ausführbare Bootsektoren enthalten neben den bereits aufgeführten Konstanten, die das Diskettenformat beschreiben, zusätzliche Angaben:

Byte	*Name*	*Bedeutung*
30-31	EXECFLG	Wird das Betriebssystem von Diskette gebootet, wird bei gesetztem Flag zusätzlich ein Programm namens COMMAND.PRG (falls vorhanden) geladen und gestartet
32-33	LDMODE	Ist dieses Flag gesetzt, werden beim Systemstart einzelne Sektoren geladen
34-35	SSECT	Erster Sektor, der geladen wird, falls LDMODE gesetzt ist
36-37	SECTCNT	Zahl der Sektoren, die geladen werden sollen
38-41	LDADDR	Adresse im Hauptspeicher, an die die oben definierten Sektoren oder eine Programmdatei geladen werden sollen
42-45	FATBUF	An diese Speicheradresse werden FAT und Directory der Bootdisk geladen
46-57	FNAME	Ist LDMODE gleich Null, werden nicht einzelne Sektoren, sondern eine komplette Datei geladen. In diesem Fall enthält FNAME den Dateinamen

Zum Erzeugen eines ausführbaren Bootsektors bietet DISKUS unter **Disk->Bootsektor** einige Erleichterungen. Mit **Bootprogramm installieren** lassen sich eigene Assemblerprogramme im Bootsektor unterbringen. Solche Programme dürfen die Größe des Bootsektors nicht überschreiten und enden in der Regel mit einem RTS, also einem Rücksprung ins Betriebssystem.

13.5. Die Diskbelegungstabelle (FAT)

TOS vermerkt in der FAT, welche Cluster einer Mediums belegt sind und ob ein Cluster defekt ist. In diesem Fall wird nicht versucht, diesen Cluster zur Datenspeicherung zu verwenden.

Bei Disketten belegt jeder Eintrag in der FAT 12 Bit. Dieses Format erlaubt es, in drei Bytes zwei FAT-Einträge unterzubringen (2 Einträge zu je 12 Bit = 24 Bit =3 Bytes). Bei Festplatten reichen 12 Bit nicht aus, um für jeden Cluster einen Eintrag bereitzuhalten. Deshalb ist hier jeder Eintrag 16 Bit (also zwei Bytes) lang. Ob ein Medium eine 12- oder 16-Bit-FAT besitzt, kann dem BPB (Bios Parameter Block) entnommen werden. Die von DISKUS nicht unterstützten FAT32-Partitionen besitzen eine 32-Bit-FAT.

Folgende FAT-Einträge sind möglich:

Inhalt	*Bedeutung*
$0000	Cluster ist nicht belegt
$0002-$7FFF	Cluster ist belegt
$8000-$FFEF	Ungültiger Eintrag
$FFF0-$FFF7	Cluster ist defekt
$FFF8-$FFFF	Cluster ist letzter Cluster einer Datei

Die oben aufgeführten Werte stellen 16-Bit-Worte dar. Bei 12-Bit-FATs liegen dreistellige hexadezimale Angaben vor. Die ersten beiden FAT-Einträge enthalten keine relevanten Informationen. Erst ab dem

zweiten Eintrag haben die FAT-Informationen eine Bedeutung. Die Zählung der Cluster beginnt somit auch erst bei Cluster 2. Cluster 0 und 1 existieren nicht, was mit der DOS-Kompatibilität zusammenhängt. Ist ein Cluster belegt, gibt der entsprechende Eintrag in der FAT an, welcher Cluster der nächste Cluster der Datei ist. Auf diese Weise wird eine logische Verkettung zwischen den Daten-Clustern einer Datei hergestellt.

Hierzu ein Beispiel. Ein Ausschnitt aus dem Beginn einer 16- Bit-FAT könnte so aussehen:

00 00 00 00 03 00 04 00 05 00 FF FF

Die ersten beiden Einträge (also die ersten vier Bytes) sind ungenutzt. Interessant wird es erst bei den beiden Bytes 03 00. Diese Bytes stellen den FAT-Eintrag für Cluster 2 dar. Wie schon bei den Daten des Bootsektors sind auch die FAT-Einträge im Intel-Format abgelegt. 03 00 heißt für den Atari also $0003, da low und high Byte vertauscht sind. Somit stellt Cluster 3 den Folgecluster von Cluster 2 dar. Bewegt man sich weiter durch die FAT erkennt man, dass auf Cluster 3 Cluster 4 und auf diesen Cluster 5 folgt. Cluster 5 wird durch den Eintrag $FFFF gekennzeichnet. Dieser Wert steht für das Dateiende. Cluster 5 ist also der letzte Cluster einer Datei, die die Cluster 2-5 belegt. Um welche Datei es sich handelt geht aus der FAT nicht hervor. Hierzu muss man sich die zugehörigen Informationen aus dem Directory besorgen. Hier findet man die Zuordnung zwischen dem Dateinamen und dem ersten FAT-Eintrag einer Datei.

Nun zu Disketten, bei denen man sich mit 12-Bit-FATs herumschlagen muss. Der oben dargestellte FAT-Ausschnitt sieht bei einer 12-Bit-FAT so aus:

F9 FF FF 03 40 00 05 F0 FF

(Der Inhalt der ersten drei Bytes kann je nach Diskettenformat unterschiedlich sein.)

Die zum selben Cluster gehörenden Daten sind jeweils einfach bzw. doppelt unterstrichen, um die Zuordnung zu verdeutlichen. Mit DISKUS müssen Sie sich jedoch nicht um solche Details kümmern. DISKUS erkennt das FAT-Format und passt alle FAT-Manipulationen selbstständig an.

13.6. Das Wurzelverzeichnis

Im Wurzelverzeichnis ist für jede Datei ein Eintrag von 32 Bytes reserviert. Die Bytes, aus denen sich ein solcher Datei-Eintrag zusammensetzt, bedeuten im Einzelnen:

Byte	***Bedeutung***
$00-$0A	Dateiname
$0B	Dateiattribut
$0C-$15	Reserviert
$16-$17	Uhrzeit der Dateierzeugung
$18-$19	Datum (gerechnet ab 1980) der Dateierzeugung
$1A-$1B	Erster Daten-Cluster der Datei
$1C-$1F	Dateilänge in Bytes

Auch Datei-Einträge in Ordnern sind nach diesem Schema organisiert. Im Gegensatz zum Wurzelverzeichnis, dessen Länge im Bootsektor vermerkt ist, können Ordner beliebig lang sein. (Bei FAT32-Partitionen hat übrigens auch das Wurzelverzeichnis eine variable Länge.)

Hier ein Beispiel für einen Eintrag, wie er in DISKUS auf dem Bildschirm erscheint:

```
0040   43 50 58 20 20 20 20 20-20 20 20 10 00 00 00 00  CPX        0
0050   00 00 00 00 00 00 2A AC-FA 16 04 00 00 00 00 00        *¼.6◇
```

Wieder liegen alle Angaben im Intel-Format vor. Es handelt sich um den Ordner CPX, wie man im ASCII-Teil der Anzeige erkennen kann. Der Punkt, der den Namen von der Extension trennt, taucht hier nicht auf. Das Dateiattribut ist $00, was in binärer Schreibweise %00000000 bedeutet. Jedes Bit des Attributs hat eine Bedeutung, die aus folgender Tabelle hervorgeht (das binär rechts erscheinende Bit ist Bit 0):

Bit	***Bedeutung***
0	Datei ist schreibgeschützt
1	Versteckte Datei, wird nicht vom Desktop angezeigt
2	Systemdatei, wird nicht vom Desktop angezeigt
3	Dieser Name ist der Name des Mediums
4	Es handelt sich um einen Ordner
5	Von dieser Datei wurde noch kein Backup angelegt (Archivbit)
6	Symbolischer Link (MiNT oder MagiC)

Das Archivbit wird erst ab TOS 1.04 korrekt verwendet. Es teilt Backup-Programmen mit, ob eine Datei seit dem letzten Backup verändert wurde.

Die Angabe des ersten Daten-Clusters einer Datei genügt, um auch die restlichen Cluster der Datei auffinden zu können. Sucht man in der FAT den Eintrag des Startclusters der Datei auf, findet man hier die Nummer des nächsten Clusters und ist so über alle weiteren Cluster im Bilde. Die Dateilänge in Bytes, die bei Ordnern stets 0 ist, rundet schließlich den Aufbau eines Verzeichnis-Eintrags ab.

Um diese Daten schnell auswerten zu können zeigt DISKUS beim Anklicken der Bytes eines solchen Eintrags mit der rechten Maustaste alle Informationen im Klartext an, solange die Maustaste gedrückt bleibt.

13.7. Der Aufbau von Ordnern

Ordner sind ähnlich wie das Wurzelverzeichnis aufgebaut. Die ersten beiden Einträge sind für das Betriebssystem reserviert und werden vom Desktop nicht angezeigt. Die "Dateinamen" dieser Einträge sind "." und "..". Die Startcluster-Angabe des Eintrags ".." enthält den Startcluster des übergeordneten Ordners oder eine 0, wenn es keinen übergeordneten Ordner gibt, weil sich der Ordner im Hauptverzeichnis befindet. Der Startcluster von "." enthält die Cluster-Nummer des Ordners selbst. Beide Einträge werden von TOS nicht genutzt und sind nur aufgrund der DOS-Kompatibilität vorhanden.

Ob diese Ordnerstruktur fehlerfrei ist, kann mit Hilfe von **Ordnerstruktur testen** unter **Spezial->Daten testen** ermittelt werden.

13.8. Der Bios Parameter Block (BPB)

Anhand der Daten im Bootsektor erkennt TOS das Diskettenformat. Diese Daten werden jedoch vor ihrer eigentlichen Verwendung noch aufbereitet. Die für TOS relevanten Bootsektor-Informationen werden für jedes angemeldete Laufwerk im sogenannten Bios Parameter Block im Speicher des Atari abgelegt.

Der BPB ist folgendermaßen aufgebaut:

Byte	***Name***	***Bedeutung***
$00	RECSIZ	Sektorgröße in Bytes
$02	CLSIZ	Clustergröße in Sektoren
$04	CLSIZB	Clustergröße in Bytes
$06	RDLEN	Länge des Wurzelverzeichnisses in Sektoren
$08	FSIZ	Größe der FATs in Sektoren
$0A	FATREC	Erste Sektornummer der zweiten FAT
$0C	DATREC	Sektornummer des ersten Daten-Clusters
$0E	NUMCL	Zahl der Daten-Cluster
$10	BFLAGS	Bit 0: 16-Bit-FAT, Bit 1: nur eine FAT
$12	TRACKS	Zahl der Spuren
$14	DNSIDES	Zahl der Seiten
$16	DNSPDT	Sektoren pro Spur (Summe von Vorder- und Rückseite)
$18	DNSPT	Sektoren pro Spur auf einer Seite
$1A	DHIDDEN	Zahl der versteckten Sektoren
$1C	DSERIAL	Drei Bytes für die Seriennummer des Mediums

Kann TOS aus den Daten des Bootsektors aufgrund fehlerhafter Informationen keinen BPB aufbauen, kann das Medium nicht von TOS, wohl aber von DISKUS verwendet werden. Falls ein Medium einen ungültigen BPB besitzt, wird dies durch den Knopf **BPB ungültig** signalisiert.

Mit der Funktion **Disk->Format definieren** können Sie einen neuen BPB definieren, wenn durch einen Fehler der ursprüngliche Bootsektor einer Diskette zerstört worden ist oder sich Fehler in den Daten des Bootsektors befinden. Dieser neu definierte BPB kann mit DISKUS auch in Form eines neuen Bootsektors auf eine Diskette geschrieben werden, um die Lesbarkeit des Mediums für TOS wieder herzustellen.

13.9. Alternative Dateisysteme

Das FAT16-Dateisystem ist inzwischen in die Jahre gekommen. Lange Dateinamen und nahezu beliebig große Partitionen gehören inzwischen fast zum Standard. Mit MagiC oder MiNT lassen sich einige der Einschränkungen von TOS umgehen. Diese Systeme erlauben die Verwendung alternativer Dateisysteme: VFAT, FAT32 und EXT2FS (Linux) bei MiNT sowie VFAT und FAT32 unter MagiC. Das VFAT- sowie das FAT32-Format werden auch von Windows, Linux und macOs unterstützt. Diese Formate sind also gut für den Datenaustausch zwischen PC und Atari unter Erhalt langer Dateinamen geeignet.

Bei der Verwendung fremder Dateisysteme auf dem Atari ist zu beachten, dass einige Programme, zu denen auch DISKUS zählt, lange Dateinamen und FAT32-Partitionen nicht unterstützen. Außerdem gibt es keine Möglichkeit, bei alternativen Dateisystemen Manipulationen vorzunehmen, wie sie DISKUS bei FAT16-Partitionen durchführen kann. Daher können Dateisysteme wie FAT32 oder EXT2FS innerhalb von DISKUS nicht auf Datei-Ebene angesprochen werden. VFAT-Dateisysteme hingegen können mit DISKUS nahezu ohne Einschränkungen bearbeitet werden. Lediglich Operationen, die sich direkt auf Dateinamen beziehen, sollten vermieden werden, da DISKUS nur die kurzen Namen erkennt. Der lange Name kann in solchen Fällen verloren gehen.

14. Das Beheben von Datenverlusten

Nun soll das Thema sein, wie Daten durch Fehler auf dem Datenträger, insbesondere durch Fehler in Bootsektor, FAT oder Directory, in Mitleidenschaft gezogen werden können, und wie man die betroffenen Daten eventuell noch retten kann.

Häufig sind von einem Fehler nicht nur wenige Bytes betroffen, sondern mindestens ein Sektor. Bevor man irgendwelche Manipulationen vornimmt, sollte eine schadhafte Diskette unbedingt kopiert werden. Manche der üblichen Kopierprogramme, die häufig spezielle Möglichkeiten zum Kopieren geschützter Disketten bieten, dürften sogar in der Lage sein, eine korrekte Kopie der Fehler auf den fehlerhaften Sektoren anzufertigen.

Bei einem fehlerhaften Sektor auf einer Fest- oder Wechselplatte ist es im Gegensatz zu Defekten auf Disketten nicht möglich, eine Kopie dieses Sektors zu erzeugen. Meldet also der Festplattentreiber, dass ein Sektor nicht mehr gelesen werden kann, besteht keine Möglichkeit, die Daten dieses Sektors zurückzuerhalten. Hier kann eine Reparatur lediglich darin bestehen, die restlichen Daten der betroffenen Datei oder Partition zu retten. Aktuelle Platten ersetzen defekte Sektoren automatisch durch Reservesektoren, ohne dass der Anwender dies bemerkt.

14.1. Fehler in Verzeichnissen

Ist ein Verzeichnis-Sektor fehlerhaft, kann TOS auf die Dateien, deren Einträge sich auf diesem Sektor befinden, nicht mehr zugreifen, selbst wenn die eigentlichen Daten noch vorhanden sind. Im ungünstigsten Fall führt ein Zugriff vom Desktop aus zu einem Absturz. Ähnlich ist die Situation, wenn der defekte Sektor Bestandteil eines Ordners ist. Hier gibt es aber erst dann Probleme, wenn dieser Ordner geöffnet wird. Es werden dann meist unsinnige Dateilängen und -namen angezeigt oder manche Dateien fehlen ganz.

Sind bis auf einen oder mehrere Verzeichnis-Sektoren keine weiteren Sektoren fehlerhaft, befinden sich die eigentlichen Daten weiterhin auf dem Medium, können jedoch unter TOS nicht mehr angesprochen werden. Ein Griff zu DISKUS kann hier weiterhelfen.

Zunächst sollte man sich vergewissern, ob auch DISKUS Schwierigkeiten hat, das Directory zu lesen. Es kann vorkommen, dass der Verzeichnis-Eintrag einer Datei in Ordnung ist und TOS dennoch nicht in der Lage ist, die Datei zu lesen, weil sich an anderer Stelle des Mediums Probleme ergeben. Können Sie in DISKUS den gewünschten Eintrag in der Dateiauswahlbox auffinden, kann die Datei mit **Datei->Kopieren** auf einem anderen Medium gesichert werden.

Sind gleich mehrere Sektoren einer Diskette defekt, kann sie dennoch von DISKUS ohne größeren Aufwand kopiert werden, wenn im Kopier-Dialog die Einstellung **Lesefehler ignorieren** aktiviert wird, damit es nicht ständig Fehlermeldungen hagelt.

Sind nicht alle Verzeichnis-Sektoren so zu reparieren, wie soeben beschrieben, ist die FAT aber fehlerfrei, kann DISKUS mit Hilfe der FAT das **Wurzelverzeichnis rekonstruieren** (**Spezial->Daten retten**). Anhand der FAT lässt sich ermitteln, welche belegten Cluster Start-Cluster von Dateien darstellen. Mit diesen Informationen kann DISKUS ein neues Directory anfertigen. Die Angaben über Dateinamen und

Dateilängen im Wurzelverzeichnis können jedoch nicht zurückgewonnen werden, da diese Angaben ausschließlich im Hauptverzeichnis abgelegt sind. Die Namen und Dateilängen von Dateien, die sich in Ordnern befinden, bleiben jedoch erhalten.

Die von **Wurzelverzeichnis rekonstruieren** gefundenen Dateien werden von DISKUS beginnend mit "FILE0" durchnummeriert. Ordner erhalten Namen ab "FOLDER0". Als Dateilängen werden Vielfache der Länge eines Clusters eingetragen. So ist sichergestellt, dass alle der ursprünglichen Datenbytes erfasst werden. Beim Laden einer so rekonstruierten Datei werden nun jedoch eventuell ein paar Bytes mehr geladen, als es der ursprünglichen Dateilänge entspricht. Bei Textdateien lassen sich überflüssige Zeichen am Ende des Textes mit Hilfe eines Texteditors entfernen und bei Programmdateien spielen ein paar Bytes zu viel in der Regel keine Rolle. Falls Sie wissen, wie lang eine mit **Wurzelverzeichnis rekonstruieren** gerettete Datei ursprünglich war, kann die korrekte Dateilänge mit Hilfe der Funktion **Datei->Information** nachgetragen werden.

Man sollte sich die so rekonstruierten Dateien "FILEXXX" anschließend ansehen (**Datei->Öffnen**) um festzustellen, um welche Datei es sich ursprünglich gehandelt hat. Bei Textdateien ist dies recht einfach, da man Texte normalerweise gut identifizieren kann. Schwierig ist es jedoch bei Programmdateien. Solche Dateien erkennt man daran, dass die ersten beiden Bytes den Wert $601A besitzen. DISKUS berücksichtigt dies, indem das Programm solche Dateien automatisch mit der Extension "PRG" versieht. Über einen Doppelklick kann man eine solche Datei starten und so feststellen, um welches Programm es sich konkret handelt. Anschließend kann per **Datei->Information** der korrekte Name vergeben werden.

14.2. Fehler in der FAT

Bei einer fehlerhaften FAT kann es vorkommen, dass Teile von Dateien verschwunden sind, Dateien durcheinander gewürfelt werden, oder der Atari bis in alle Ewigkeit versucht, eine Datei zu laden. Ist die FAT fehlerhaft, kann sich der Dateiinhalt beim Anzeigen einer Datei nach einer gewissen Zeit wiederholen und das Ende der Datei wird nie erreicht.

Diese Effekte rühren daher, dass fehlerhafte FAT-Einträge die Verkettung der Daten-Cluster in Unordnung bringen. Um solche Fehler festzustellen gibt es in DISKUS die Funktion **FAT testen** im Rahmen von **Spezial->Daten testen**. Wurden durch einen FAT-Fehler Dateien zerstört, kann die Anwendung von **Datei rekonstruieren** die Daten eventuell noch retten. Andernfalls bleibt nur der Weg, die verlorenen Daten "per Hand" zu suchen. Dieses oft recht zeitaufwändige Verfahren wird weiter unten beschrieben.

Ist die FAT völlig zerstört, das Wurzelverzeichnis aber noch intakt, kann die Funktion **FAT-Prototyp erzeugen** unter **Spezial->Daten retten** weiterhelfen. Hier wird zunächst eine FAT erzeugt, bei der jeder FAT-Eintrag auf den nachfolgenden Eintrag zeigt. Daraus resultiert eine FAT, die lediglich eine einzige, große Datei darstellt, die sich über das gesamte Medium erstreckt. Mit **Daten testen->Dateien testen** bei eingeschalteter automatischer Reparatur kann DISKUS nun eine neue FAT erzeugen, die in großen Teilen fehlerfrei ist. Der Erfolg dieser Operation hängt allerdings davon ab, wie fragmentiert das Medium ist. Die so erzeugte neue FAT lässt sich bei Bedarf per Hand bearbeiten.

Bei Änderungen in der FAT sollte man große Vorsicht walten lassen. FAT-Änderungen werden daher von

DISKUS nur auf Wunsch (**Spezial->FATs schreiben**) gesichert.

14.3. Fehler im Bootsektor

Bei einem unlesbaren oder fehlerhaften Bootsektor ist TOS nicht mehr in der Lage, irgendwelche Daten von der Diskette zu holen. Sind die restlichen Sektoren der Diskette weitgehend fehlerfrei, kann man die Diskette unter DISKUS wieder lesbar machen, indem man dem Programm die Informationen, die der fehlerfreie Bootsektor enthalten müsste, mit der Funktion **Disk->Format definieren** mitteilt. Dazu ist es natürlich notwendig, dass das Format der zerstörten Diskette bekannt ist. **Format definieren** gibt das normalerweise auf dem Atari verwendete Diskettenformat für Disketten doppelter Dichte vor. Ist die Diskette abweichend formatiert, müssen diese Angaben entsprechend geändert werden.

Um den Bootsektor für TOS wieder akzeptabel zu machen, können Sie das in DISKUS definierte Format auf die Diskette schreiben. Haben Sie DISKUS das korrekte Diskettenformat mitgeteilt, empfiehlt es sich jedoch, alle Daten des Mediums anderweitig zu sichern (**Datei->Kopieren** oder **Disk->Kopieren**, eventuell mit aktivierter **Lesefehler ignorieren**-Option). Es ist ja nicht auszuschließen, dass die betroffene Diskette noch weitere fehlerhafte Sektoren enthält.

15. Das Wiederherstellen von Dateien

Das Wiederherstellen gelöschter Daten stellt ein sehr umfangreiches Thema dar. Nicht immer ist ein automatisches Reparieren von irrtümlich gelöschten Dateien möglich. Ein Restaurieren "per Hand" kann recht zeitaufwändig sein. Oft hat man jedoch keine andere Wahl, will man einen Datenverlust begrenzen.

15.1. Automatisches Restaurieren gelöschter Dateien

Eine wichtige Voraussetzung für die komplette Wiederherstellung einer Datei ist, dass noch keine neuen Daten auf den Datenträger geschrieben wurden, die bereits die Cluster der gelöschten Datei überschrieben haben könnten. Sind bereits Schreibzugriffe erfolgt, ist ein vollständiges Retten der verlorenen Daten meist nicht mehr möglich. Falls nach dem Löschen einer Datei noch keine Schreiboperationen erfolgt sind, kann die Datei in vielen Fällen automatisch restauriert werden. Hierfür ist die Funktion **Datei restaurieren** unter **Spezial->Daten retten** zuständig.

Beim Restaurieren von Dateien sucht DISKUS zunächst den ersten Cluster der gelöschten Datei, der anhand des noch vorhandenen Datei-Eintrags ermittelt werden kann. (Beim Löschen einer Datei wird lediglich das erste Zeichen des Dateinamens verändert, die restlichen Verzeichnis-Daten bleiben erhalten.) Anschließend verkettet DISKUS die dem ersten Cluster der Datei physikalisch folgenden und als unbelegt gekennzeichneten Cluster so lange miteinander, bis die ursprüngliche Dateilänge erreicht wurde.

Nach diesem Restaurierungs-Versuch empfiehlt es sich, die Cluster der betroffenen Datei durchzusehen (falls es sich um eine Textdatei handelt), um so festzustellen, ob die Reparatur erfolgreich war. Wurde eine Programmdatei repariert, kann die Funktionsfähigkeit in der Regel durch das Starten des Programms überprüft werden.

Wurde die Datei erfolgreich wiederhergestellt und ist der Datenverlust durch eine fehlerhafte Diskette aufgetreten, sollten Sie die restaurierte Datei auf einem anderen Datenträger unterbringen. Soll die reparierte Datei nicht auf einer anderen Disk untergebracht werden, sondern auf dem aktuellen Laufwerk verbleiben, müssen Sie die FAT nach dem Restaurieren mit **Spezial->FATs schreiben** zurückschreiben.

Woran kann es liegen, wenn eine Datei nicht mehr automatisch restauriert werden kann? Dieser Fall kann eintreten, wenn nach dem Löschen der Datei Schreibzugriffe stattgefunden haben oder die Datei fragmentiert ist, also nicht alle Daten-Cluster der Datei physikalisch aufeinander folgen. Dies ist der Fall, wenn die Cluster durch häufiges Schreiben und Löschen von Daten in Unordnung geraten sind. Dann ist nicht sichergestellt, dass die von DISKUS gefundenen unbelegten Daten-Cluster tatsächlich Bestandteile der gelöschten Datei sind. (Abhilfe schafft ein regelmäßiges Defragmentieren mit **Spezial->Daten optimieren**.)

Da beim Löschen einer Datei deren Cluster als unbelegt gekennzeichnet werden, stehen diese sofort wieder zur Datenspeicherung zur Verfügung. Nachfolgende Schreibzugriffe können die ursprünglichen Daten zerstören, so dass die Daten der zu restaurierenden Datei nur noch zum Teil gerettet werden können. Sind Teile einer Programmdatei bereits von neuen Daten überschrieben worden, ist ein Restaurieren zwecklos, da Programme nur dann lauffähig sind, wenn alle Daten komplett sind. Bei reinen Textdateien ist es dagegen nicht allzu tragisch, wenn Teile der Datei fehlen. Fehlende Teile des Textes können nachträglich mit einer

Textverarbeitung oder einem Texteditor wieder eingefügt werden.

15.2. Das Restaurieren "per Hand"

Es kann vorkommen, dass beim automatischen Restaurieren einer Datei zwar genügend Cluster gefunden werden, um die ursprüngliche Dateilänge zu erreichen, aber nicht alle diese Cluster tatsächlich zur gelöschten Datei gehören. Dieser Fall kann dann eintreten, wenn die Datei fragmentiert war. Hier kann ein Restaurieren "per Hand" einen Teil oder alle Daten retten. Ein jederzeit gültiges Rezept hierfür kann zwar nicht gegeben werden, jedoch ist die im folgenden beschriebene Vorgehensweise empfehlenswert.

Wurde eine Datei gelöscht und nicht aufgrund eines Fehlers auf dem Speichermedium zerstört, brauchen für einen Restaurierungsversuch nur Daten-Cluster in Betracht gezogen werden, die als unbelegt gekennzeichnet sind. Dies liegt daran, dass beim Löschen einer Datei alle von dieser Datei belegten Cluster wieder zur erneuten Unterbringung von Daten freigegeben werden. Cluster, die nicht freigegeben wurden, gehören mit Sicherheit zu anderen Dateien, brauchen also nicht berücksichtigt zu werden. Bei einem Datenverlust durch einen Diskettenfehler hingegen kommen nur als belegt markierte Cluster als Bestandteile der Datei in Frage, da in diesem Fall die Datei nicht korrekt gelöscht wurde, die Daten-Cluster somit nicht freigegeben sind.

Nun zur Vorgehensweise beim Restaurieren per Hand. Zunächst sollte der erste Cluster der zerstörten Datei gesucht werden. Ist der Verzeichnis-Eintrag der Datei noch vorhanden, können Sie diesen Cluster in die Anzeige bringen, indem Sie die Funktion **Datei->Öffnen** benutzen. In der Dateiauswahlbox muss hierzu der **gelöscht**-Knopf selektiert werden, falls die betroffene Datei vorher gelöscht wurde.

Ist der ursprüngliche Datei-Eintrag nicht mehr vorhanden, da bereits neue Daten auf das Medium geschrieben wurden, ist die Suche nach dem ersten Cluster schwieriger. Erzeugen Sie in diesem Fall zunächst mit **Datei->Leerer Eintrag** einen leeren Eintrag im Wurzelverzeichnis, der später auf die restaurierte Datei verweisen wird.

Handelte es sich bei der betroffenen Datei um eine Textdatei und ist Ihnen ein Teil des Textes bekannt, besteht die Möglichkeit, diesen Text von DISKUS suchen zu lassen, um so den Dateibeginn (oder auch andere Teile der Textdatei) aufzufinden.

Ist der erste Cluster nicht mehr zu lokalisieren, ist ein komplettes Restaurieren der Datei ausgeschlossen. Nur bei Textdateien macht es in einem solchen Fall noch Sinn, einen Rettungsversuch zu unternehmen. Versuchen Sie hierzu, einen Cluster zu finden, der, wenn er auch nicht der erste der Datei ist, möglichst nah am Anfang der Datei liegt. Mit den Knöpfen in Form von Pfeilen neben dem Knopf **phys. Cluster** lässt sich das Medium clusterweise durchsuchen. Hierbei ist es bei gelöschten Dateien hilfreich, sich nur die als unbelegt markierten Cluster anzusehen. Betätigen Sie hierzu die Knöpfe mit der rechten Maustaste.

Jeder gefundene Cluster muss nun mit dem vorhergehenden logischen Cluster der Datei verkettet werden. Dies kann mit **Datei->Verketten** geschehen. Alternativ dazu können Sie einen Cluster-Verweis direkt editieren, wenn Sie den **Cluster belegt**-Knopf anklicken und den FAT-Eintrag per Hand ändern.

In der Regel orientiert sich die logische Reihenfolge der Cluster an der physikalischen Reihenfolge auf dem

Medium, da TOS beim Schreiben von Dateien stets versucht, Cluster der Reihe nach zu vergeben. Erst dann, wenn das Ende eines Mediums erreicht ist, wird die Suche nach freien Clustern am Anfang des Datenträgers fortgesetzt.

Ist die Clusterzahl oder die Dateilänge bekannt, lässt sich anhand dieser Angaben feststellen, wie viele Cluster verkettet werden müssen, bis das Dateiende erreicht ist. Bei bekannter Dateilänge erfährt man die Zahl der Daten-Cluster, indem man die Länge durch die Clustergröße dividiert und den erhaltenen Wert um 1 erhöht, falls bei der Division ein Rest bleibt. Wie viele Bytes ein Daten-Cluster umfasst erfahren Sie mit **Disk->Struktur**.

Sind Sie mit dem Ergebnis Ihrer Bemühungen zufrieden, können Sie die restaurierte Datei auf ein anderes Medium kopieren, oder die FAT des aktuellen Laufwerks zurückschreiben (**Spezial->FATs schreiben**). Wurde der Datenfehler durch eine fehlerhafte Diskette verursacht, sollte jedoch ein anderer Datenträger verwendet werden.

Das Restaurieren "per Hand" kann recht zeitaufwändig sein. Dies ist besonders dann der Fall, wenn die Dateien eines Laufwerks arg fragmentiert sind. Dies kann jedoch durch ein regelmäßiges Defragmentieren des Mediums mittels **Spezial->Daten optimieren** verhindert werden.

15.3. Lokalisieren von Fehlern

Bisher war von Fehlern die Rede, die sich recht einfach lokalisieren ließen. Oft ist ein genaues Lokalisieren des Fehlers nicht ohne Weiteres möglich. Dies ist beispielsweise dann der Fall, wenn ein Medium keine physikalischen Defekte besitzt, sich die Daten aber dennoch nicht korrekt laden lassen. Hier gilt es, sich schrittweise an die Fehlerursache heranzutasten.

Insbesondere bei Disketten sollte beim Auftreten solcher Fehler zunächst überprüft werden, ob der Bootsektor gültige Daten enthält. Falsche Angaben über die Größe der FATs oder des Wurzelverzeichnisses führen leicht zu einem scheinbaren Chaos. DISKUS benutzt die im Bootsektor vermerkten Angaben, um beim Anklicken der Knöpfe **FAT1**, **FAT2**, **Directory** und l**og. Cluster** den ersten Sektor der entsprechenden Bereiche anzuzeigen.

Von der Größe der FATs hängt die Position der Directorysektoren ab. In der Regel wird man ungefähr wissen, welche Dateien sich im Hauptverzeichnis eines Mediums befinden. Werden beim Anklicken des Knopfes **Directory** keine oder die falschen Dateinamen angezeigt, ist die Wahrscheinlichkeit, dass der Bootsektor eine falsche Angabe über die Sektorzahl und Länge der FATs enthält, recht hoch. Die minimale Größe der FAT lässt sich ermitteln, indem man die Sektorzahl des Mediums durch die Anzahl der Bytes pro Sektor teilt, diesen Wert halbiert und anschließend nach oben rundet.

Falsche Werte für die Länge von FAT oder Directory sorgen auch dafür, dass sich der Beginn der Datensektoren verschiebt. Sind FAT und Directory optisch fehlerfrei, enthalten Dateien aber dennoch Unsinn, ist der Fehler eventuell in falschen Angaben über die Größe der Bereiche mit Verwaltungssektoren zu suchen. Um die Länge des Directories zu überprüfen genügt oft schon der Anhaltspunkt, dass die letzten Sektoren des Wurzelverzeichnisses, soweit es nicht komplett belegt ist, ausschließlich Nullbytes enthalten.

15.4. Das Verhindern von Datenverlusten

Es sollen noch einige Hinweise folgen, wie man einem Datenverlust vorbeugen kann.

Zum Testen von Disketten oder Festplatten-Partitionen lassen sich diese sektor- oder dateiweise auf Lesefehler testen. Beim sektorweisen Testen (**Lesetest** unter **Spezial->Daten testen**) werden alle Sektoren, also auch die Verwaltungssektoren (Bootsektor, FAT-Sektoren, Directory) und die unbelegten Sektoren, auf Fehler überprüft. Fehlerhafte Sektoren können markiert werden und werden danach von TOS nicht mehr zur Speicherung neuer Daten verwendet. Darüber hinaus ist das Testen von Dateien möglich (**Dateien testen** unter **Spezial->Daten testen**). Hier werden nur die von Dateien belegten Sektoren überprüft.

Auch das Testen der FAT (**FAT testen** unter **Spezial->Daten testen**) liefert Aufschlüsse darüber, ob sich ein Datenverlust andeutet. Ist die FAT fehlerhaft, sollte man sofort versuchen, den Fehler zu beheben. Hierzu sind jedoch detaillierte Kenntnisse über die Datenverwaltung seitens TOS notwendig. Sind Sie in dieser Beziehung kein Experte, sollten einfach alle Dateien auf ein anderes Speichermedium gerettet werden, so weit dies die fehlerhafte FAT noch zulässt.

Von Festplatten sollte auf einer Diskette eine Kopie des Rootsektors gesichert werden. Hierzu wird der Rootsektor, also der physikalische Sektor 0, mit **Sektor->Physikalisch lesen** geladen und anschließend mit **Spezial->Daten speichern** auf einem anderen Laufwerk (nicht auf der Platte!) als Datei gesichert. Sollten die Daten des Rootsektors durch einen Fehler verloren gehen, lässt sich mit der umgekehrten Reihenfolge, also dem Laden der Daten und dem anschließenden Schreiben des Rootsektors, der Datenverlust beheben.

Das Defragmentieren eines Mediums (**Spezial->Daten optimieren**) bringt neben einem schnelleren Datenzugriff ebenfalls mehr Datensicherheit. Da die Cluster defragmentierter Dateien physikalisch nahtlos hintereinander angeordnet sind, ist es kein Problem, gelöschte Daten zu rekonstruieren, solange noch kein Schreibzugriff erfolgt ist. In diesem Fall ist das automatische Wiederherstellen einer gelöschten Datei (**Datei restaurieren**) mit Sicherheit möglich.

Um sich zu vergewissern, ob alle Daten eines Mediums fehlerfrei sind, ist es das Einfachste, den Punkt **Spezial->Daten testen** aufzurufen und die Buttons **FATs vergleichen**, **FAT testen**, **Dateien testen** sowie **Ordnerstruktur testen** zu aktivieren. Findet DISKUS bei diesem Test keinen Fehler, können Sie davon ausgehen, dass die Daten auf dem getesteten Medium fehlerfrei sind.

Schließlich kann nicht oft genug auf Eines hingewiesen werden: Die beste Absicherung gegen Datenverluste sind regelmäßige Backups, beispielsweise auf Wechselplatten, Speicherkarten oder DVD-RAM.

16. Das Arbeiten mit Disketten

Im Gegensatz zu Festplatten, bei denen der Anwender das Plattenformat, wenn überhaupt, nur in recht engen Grenzen variieren kann, bestehen bei Disketten eine Reihe von Manipulationsmöglichkeiten.

16.1. Erhöhung der Diskettenkapazität

Wenn man sich mit der vom Desktop zur Verfügung gestellten Formatier-Funktion zufrieden gibt, verschenkt man wertvollen Diskettenplatz. Durch ein geändertes Diskettenformat kann man pro Diskettenseite 50 bis 100 KByte zusätzlichen Speicherplatz erhalten, ohne dass die Datensicherheit beeinträchtigt wird.

Um den Speicherplatz einer Diskette zu vergrößern, gibt es zwei Wege: Einerseits kann man das physikalische Diskettenformat ändern, andererseits sind Manipulationen am logischen Format möglich.

16.2. Änderung des physikalischen Formats

Eine normal formatierte Diskette doppelter Dichte besteht aus 80 Spuren zu je 9 Sektoren. Bei HD-Disketten sind es 18 Sektoren, bei ED-Disketten 36 Sektoren. Durch Erhöhung der Zahl der Spuren und/oder der Zahl der Sektoren pro Spur kann die Kapazität einer Diskette vergrößert werden. Maximal sind mit DISKUS auf DD-Disketten 87 Spuren mit jeweils 10 Sektoren pro Spur möglich. Bei HD-Disketten liegt das Maximum bei 21 Sektoren. ED-Disketten schließlich erlauben bis zu 40 Sektoren pro Spur.

Was die Zahl der Spuren betrifft, lassen nur wenige Laufwerke 87 Spuren zu. Meist macht der Kopf einer Floppy bereits deutlich vor dem Erreichen der 87. Spur Bekanntschaft mit dem inneren Anschlag des Laufwerks. Dies macht sich durch ein Knacken bemerkbar und sollte tunlichst vermieden werden, da sich sonst der Lesekopf dejustieren kann. 83 Spuren werden jedoch von vielen Laufwerken akzeptiert.

Eine Erhöhung der Sektorzahl von 9 auf 10 Sektoren pro Spur bringt bei DD-Disketten eine stattliche Kapazitätssteigerung. Behält man 80 Spuren bei und formatiert eine Diskette mit 10 Sektoren pro Spur, gewinnt man bei doppelseitigen Disketten 80 KByte an Speicherplatz.

16.3. HD-Laufwerke am ST

Beim Betrieb von HD-Laufwerken am Atari wird die Taktfrequenz des Floppy-Controllers von 8 auf 16 MHz erhöht. Auf einer HD-Diskette lassen sich ca. 1.4 MByte an Daten unterbringen. Dieses Diskettenformat ist PC-kompatibel. Der Floppy-Controller WD1772 der STs und mancher TTs ist eigentlich nicht für Taktfrequenzen jenseits von 8 MHz gedacht. Ein Kühlkörper kann hier helfen. Manche Hardware-Anbieter haben selektierte Controller im Programm, die für hohe Taktfrequenzen geeignet sind. Die beste Lösung für einen fehlerfreien HD-Betrieb stellt aber die Verwendung des AJAX Floppy-Controllers von Atari dar.

16.4. HD-Laufwerke am TT und Falcon030

An den TT kann ein externes HD-Laufwerk ohne größeren Aufwand direkt angeschlossen werden. TTs, die mit TOS 3.06 ausgeliefert wurden, haben bereits ein HD-Laufwerk sowie einen für HD-Disketten

geeigneten AJAX Floppy-Controller eingebaut. TTs älteren Datums sind noch mit einem WD1772 Controller ausgestattet. Auch hier muss also mit Wärmeproblemen gerechnet werden. Der Falcon030 sowie alle Atari-Clones sind mit einem HD-Laufwerk ausgestattet.

Vor der Verwendung von HD-Disketten sollte bei älteren TTs ein DIP-Schalter entsprechend eingestellt werden, damit im Formatier-Dialog des Desktops ein Knopf mit der Bezeichnung **Hohe Schreibdichte** erscheint. Die DIP-Schalter befinden sich unterhalb der Festplatte. Es muss lediglich Schalter 6 auf ON gesetzt werden. Beim TT kann der Knopf **Hohe Schreibdichte** erst ab TOS 3.05 angewählt werden.

16.5. ED-Laufwerke

Der AJAX Floppy-Controller kann mit bis zu 32 MHz getaktet werden. Daher waren Hardware-Lösungen erhältlich, die die Verwendung von ED-Laufwerken am Atari erlauben. Mit diesen können Disketten mit deutlich mehr als 3 MByte an Kapazität formatiert werden.

16.6. Der Interleave-Faktor

Dieser Parameter legt fest, in welcher Reihenfolge die Sektoren einer Spur beim Formatieren auf die Diskette geschrieben werden. Normalerweise ist diese Reihenfolge auf einer normal formatierten Disk 1,2,3,4,5,6,7,8,9. Ändert man den Interleave jedoch auf 2, erhält man die Reihenfolge 1,3,5,7,9,2,4,6,8. Zwischen zwei Sektornummern liegt also ein Abstand von 2. Abhängig vom Diskettenformat kann eine Änderung des Interleave-Faktors eine Beschleunigung des Diskettenzugriffs mit sich bringen.

Je mehr Sektoren sich auf einer Spur befinden, umso eher kann es vorkommen, dass der Floppy-Controller den aktuellen Sektor nicht komplett übertragen kann, bevor der folgende Sektor erreicht wird. Befinden sich zwei Sektoren mit aufeinander folgenden logischen Sektornummer physikalisch nicht direkt hintereinander auf der Diskette (Interleave ungleich 1), bleibt dem Controller mehr Zeit für die Datenübertragung. Bei zu großen Werten für den Interleave-Faktor sinkt jedoch die Geschwindigkeit der Übertragung deutlich. In den meisten Fällen ist für den Atari ein Interleave-Faktor von 1 optimal.

16.7. Atari-Disketten und PC-Kompatibilität

Vom grundsätzlichen Aufbau her sind Disketten für den Atari PC-kompatibel. Lediglich das Media Descriptor Byte und ein paar Bytes am Anfang des Bootsektors müssen geändert werden, damit ein PC Atari-Disketten verarbeiten kann. Ab TOS 1.04 wird bereits mit dem korrekten PC-Format formatiert.

PC-kompatible Disketten erfordern am Beginn des Bootsektors die Bytefolge $EB34xx oder $E9xx, die einen Sprungbefehl für die Intel-Prozessoren darstellt. Zwar fördert das Schreiben dieser Bytes die PC-Kompatibilität, aber leider wird es dadurch je nach TOS-Version unmöglich, diese Disketten mit einem auf dem Atari ausführbaren Bootsektor zu versehen. Erst ab TOS 1.04 muss am Anfang eines Bootsektors kein Sprungbefehl auf das Bootprogramm mehr stehen. Hier genügt bereits eine korrekte Prüfsumme, um ein Bootsektor-Programm ausführen zu lassen. Bei älteren TOS-Versionen wird jedoch ein solcher Sprungbefehl, der durch die Bytes $60xx repräsentiert wird, unbedingt benötigt. In diesem Fall heißt es: PC-Kompatibilität schließt einen ausführbaren Bootsektor aus.

Sollen Disketten, die für den Atari formatiert wurden, nachträglich PC-kompatibel gemacht werden, geschieht dies bei DISKUS mit der Funktion **PC-Kompatibilität** unter **Disk->Bootsektor**.

17. Das Arbeiten mit Festplatten und Speicherkarten

Der folgende Abschnitt wird Sie mit den Datenstrukturen auf Fest- und Wechselplatten sowie Speicherkarten vertraut machen.

17.1. Der Rootsektor

Damit der Plattentreiber ermitteln kann, wie eine Platte logisch organisiert ist, sind einige wichtige Daten auf einem besonderen Sektor der Platte vermerkt. Dieser Sektor wird als Rootsektor bezeichnet und ist der Sektor 0 der Platte. Wie viele Partitionen es gibt, wie groß diese sind und wo sie sich befinden, alle diese Angaben enthält der Rootsektor. Um diese Informationen abzurufen, steht bei DISKUS die Funktion **Harddisk->Partitionen** zur Verfügung.

In der Regel werden die Partitionsdaten beim Atari im sogenannten Master Boot Record-Partitionsschema (MBR) abgelegt, das nachfolgend beschrieben wird. Inzwischen existiert mit der UEFI GUID-Partitionstabelle (GPT) ein neueres, flexibleres Partitionsschema. GUID-Partitionstabellen werden auf dem Atari von HDDRIVER unterstützt.

Wie ein Bootsektor kann auch ein Rootsektor ausführbar sein und ein Programm enthalten, das beim Booten automatisch gestartet wird. Um einen Rootsektor ausführbar zu machen stellt **Harddisk->Rootsektor** Hilfsmittel zur Verfügung.

Da der Rootsektor der wichtigste Sektor der Festplatte ist, sollten seine Daten unbedingt auf einem anderen Datenträger gesichert werden. Falls die Daten auf dem Rootsektor beschädigt werden, besteht dann die Möglichkeit, die Format-Informationen des Rootsektors auf die Platte zurückzuschreiben. Beim GPT-Partitionsschema werden alle Partitionsdaten grundsätzlich doppelt abgelegt, um das Risiko eines Datenverlustes zu verringern.

Hier die für die Atari-Festplatten relevanten Angaben in einem MBR-Rootsektor:

Byte	***Name***	***Bedeutung***
$1C6	p0_flag	Flag für Partition 0 Bit 0: Partitionsinformationen sind gültig Bit 6: bootbare TT SVR4-Partition (Atari Unix) Bit 7: bootbare TOS-Partition
$1C7	p0_id	Dreibuchstabige Kennung
$1CA	p0_st	Physikalische Sektornummer des ersten Sektors
$1CE	p0_size	Größe in Sektoren
$1D2-$1F2		Beschreibung der Partitionen 1-3, s. o.
$1F6	bsl_st	Sektornummer des ersten Sektors der Soft-Defektliste
$1FA	bsl_cnt	Größe der Soft-Defektliste in Sektoren

Byte	*Name*	*Bedeutung*
$1FE	chksum	Ausgleichsbytes für die Prüfsumme

17.2. Aufteilung in Partitionen

Da TOS ohne Erweiterungen wie MagiC oder MiNT nur Laufwerke bis zu einer gewissen Clusterzahl verwalten kann, müssen Medien mit großer Kapazität in mehrere logische Laufwerke (Partitionen) aufgeteilt werden. Die Informationen über diese Aufteilung enthält der Rootsektor. Partitionen, die unter TOS verwendet werden sollen, müssen die Kennung GEM oder BGM besitzen. Andere Kennungen erfordern zusätzlich zum Festplattentreiber spezielle Treiber für das jeweilige Dateisystem.

Die Aufteilung in logische Laufwerke geschieht beim Partitionieren. Dabei wird die Platte in mehrere logische Laufwerke unterteilt. Jedes dieser Laufwerke muss so klein sein, dass die Zahl der Daten-Cluster eine gewisse Obergrenze nicht überschreitet. Diese Grenze hängt von der TOS-Version und dem Dateisystem ab.

17.2.1. GEM-Partitionen

Kleine Festplatten-Partitionen bestehen aus Sektoren zu je 512 Bytes und besitzen in der Regel die Kennung GEM. Da je nach TOS-Version eine Obergrenze für die Anzahl der Cluster pro Laufwerk existiert, ist die maximale Größe einer solchen Partition auf 16 MByte (ab TOS 1.04 auf 32 MByte) beschränkt.

17.2.2. BGM-Partitionen

Große TOS-kompatible Partitionen besitzen stets die Kennung BGM, die aber auch für kleinere Partitionen verwwendet werden kann. Die Sektorgröße ist in der Regel ein Vielfaches von 512 Bytes, maximal je nach TOS-Version bis zu 32768 Bytes. BGM-Partitionen können eine Kapazität von bis zu 256 MByte aufweisen, ab TOS 1.04 bis zu 512 MByte. 1 GByte pro BGM-Partition sind mit TOS 4.0 sowie MiNT, MagiC oder Big-DOS möglich.

17.2.3. XGM-Einträge

Da im primären MBR-Rootsektor nur Platz für 4 Partitions-Einträge vorgesehen ist, wurde neben den Kennungen GEM und BGM, die "richtige" Festplatten-Partitionen kennzeichnen, die XGM-Kennung eingeführt.

Die Einträge, die bei normalen Partitionen Startsektor und Länge einer Partition angeben, erhalten im Falle von XGM-Einträgen eine neue Bedeutung. An Stelle des Startsektors wird hier die Sektornummer eines weiteren Rootsektors angegeben. Der Eintrag für die Länge der Partition wird durch eine Angabe über die Zahl der Sektoren, die in diesem Rootsektor vergeben werden, ersetzt. Dieser neue Rootsektor kann wiederum einen XGM-Eintrag enthalten, so dass theoretisch eine beliebige Anzahl an Rootsektoren und dadurch beliebig viele Partitionseinträge realisiert werden können. Mehr als 14 Partitionen (mit HDDRIVER 16) können unter Standard-TOS nicht benutzt werden, da TOS nicht mehr als 16 Laufwerke (inklusive der beiden Floppies) verwalten kann. Abhilfe schafft die Verwendung von MiNT, MagiC oder Big-DOS.

17.3. DOS/Windows-konforme Partitionierung

Die Aufteilung einer Festplatte in Partitionen unter DOS oder Windows geschieht ähnlich wie beim Atari. Es existieren Partitionstypen, die im Großen und Ganzen dem entsprechen, was sich unter TOS hinter den GEM-, BGM- oder XGM-Kennungen verbirgt. Lediglich das Format dieser Kennungen sowie die Anordnung der Informationen auf dem Rootsektor unterscheiden sich in dem, was man von TOS her kennt.

Der Festplattentreiber HDDRIVER erkennt die DOS/Windows-konformen Partitionstypen $01, $04, $05, $06, $0B, $0C, $0E und $0F. Windows-partitionierte Platten lassen sich daher sowohl auf dem Atari als auch auf PCs zur Datenspeicherung und zum Datenaustausch einsetzen. Partitionen größer als 32 MByte gemäß Typ $06, $0E oder $0F lassen sich auf dem Atari nicht ohne weiteres verwenden, da TOS darauf nicht vorbereitet ist. Diese Einschränkung kann allein vom Festplattentreiber nicht umgangen werden, wohl aber mit HDDRIVER in Verbindung mit MagiC, MiNT oder Big-DOS. Einem einfachen Datenaustausch mit Windows- oder auch Linux-PCs und Macs steht so nichts im Wege.

17.4. TOS/Windows-kompatible Partitionierung

Soll ein Medium sowohl auf einem Atari, einem PC als auch einem Mac benutzbar sein, kann es mit der Festplattensoftware HDDRIVER TOS/Windows-kompatibel eingerichtet werden. Für den Datenaustausch zwischen TOS und Windows, Linux oder macOS wird dann keine zusätzliche Software benötigt.

17.5. GPT-Partitioniierung

Ein Datenaustausch zwischen TOS und Linux ist außerdem bei Medien mit TOS-Partitionen und GPT-Partitionsschema möglich. Insbesondere unter Linux mit Kernel-Versionen ab 6.0 ist dieses Schema zu empfehlen. Bei einem GPT-Partitionsschema ist HDDRIVER als Festplattentreiber erforderlich.

17.6. Hinweise zur Partitionierung

Bevor man eine Platte partitioniert sollte man sich klar darüber werden, welche Aufteilung für die eigenen Bedürfnisse geeignet ist. Aus Gründen der Datensicherheit kann es sich als sinnvoll erweisen, Partition C: möglichst klein zu wählen und auf dieser Partition keine wichtigen Daten unterzubringen.

Je größer eine Partition wird, umso größer wird die Größe eines Datenclusters, also der kleinsten Einheit, die zur Datenspeicherung vergeben wird. Bei einer logischen Sektorgröße von 16384 Bytes pro Sektor werden pro Datei stets Vielfache von 32768 Bytes verbraucht. Es ist daher ungünstig, auf großen Partitionen viele kleine Dateien unterzubringen, denn so sind selbst große Platten schnell voll. Daher sollte man darauf achten, große Partitionen vorwiegend für große Dateien zu verwenden und Ordner, die in erster Linie kleine Dateien enthalten, auf kleineren Partitionen anlegen.

17.7. Physikalische Zugriffe

Auch wenn eine Platte meist in mehrere Partitionen aufgeteilt ist, befinden sich diese logischen Laufwerke auf einem physikalischen Medium. Wie auch bei einer Diskette wird auf eine Harddisk oder Speicherkarte sektorweise zugegriffen. Dabei besteht die Möglichkeit eines logischen oder physikalischen Zugriffs. Ein

logischer Zugriff orientiert sich stets an einer konkreten Partition und die Sektornummer wird relativ zum Startsektor dieser Partition angegeben. DISKUS unterstützt bei physikalischen Zugriffen 64-Bit-Sektornummern und kann so in Verbindung mit HDDRIVER auch auf Medien mit einer Kapazität von mehr als 16 Terabyte zugreifen.

Bei einem physikalischen Festplatten-Zugriff verliert die Einteilung in Partitionen an Bedeutung. Alle Sektoren der Platte werden mit 0 beginnend durchnummeriert. Im Regelfall weiß man bei einem physikalischen Zugriff nicht, auf welcher Partition sich der betroffene Sektor befindet. Aus diesem Grund sind physikalische Zugriffe nur in speziellen Fällen sinnvoll. Bei Reparaturen nach Datenverlusten auf der Festplatte kann es wichtig sein, eine Möglichkeit zum physikalischen Zugriff zu besitzen.

Ein Beispiel soll den Zusammenhang zwischen logischen und physikalischen Sektoren verdeutlichen. Nehmen wir an, ein Medium ist in zwei Partitionen C: und D: aufgeteilt. Der erste Sektor von Partition C: (also der Bootsektor) liegt in der Regel auf dem physikalischen Sektor 2 der Platte. Hat Partition C: eine Größe von 10000 Sektoren, liegt der Bootsektor von Partition D: auf dem physikalischen Sektor 10002. Greift man nun logisch auf Sektor 2 der Partition D: zu, entspricht dies dem Lesen des physikalischen Sektors 10004. Greift man physikalisch auf Sektor 2 der Festplatte zu, spricht man den Bootsektor von Partition C: an, also den logischen Sektor 0 dieser Partition. Ein Zugriff auf den Rootsektor ist ausschließlich physikalisch möglich. Dies liegt daran, dass dieser Sektor zu keiner Partition gehört.

Da die Daten im Rootsektor für den Harddisk-Treiber zum Lesen und Schreiben von Daten notwendig sind, können Änderungen dieser Daten zu Datenverlusten führen. Deshalb muss hiervon dringend abgeraten werden, es sei denn, Sie wissen, was Sie tun. Eine sinnvolle Änderung des Rootsektors ist z. B. das Ändern einer Partitionskennung. Auch die Aufteilung der Platte in Partitionen kann nachträglich manipuliert werden, ohne die Festplatte neu partitionieren zu müssen. Dieses Unterfangen ist jedoch recht kompliziert.

17.8. Die Plattenstatistik

Eine besondere Eigenschaft einiger alter Festplatten-Controller ist eine Statistik, die sie über die während des Arbeitens mit der Harddisk durchgeführten Such- und Leseoperationen und eventuell aufgetretene Lesefehler führen. Eine solche Statistik kann wertvolle Hinweise bei der Überprüfung des Laufwerksverhaltens liefern. DISKUS erlaubt es, Einblick in die Laufwerksstatistik zu nehmen (**Fehlerstatistik** unter **Harddisk->Kommando senden**). Diese Funktionalität ist nicht zu verwechseln mit der SMART-Technologie neuerer Platten.

Der Controller unterscheidet bei Lesefehlern zwei Situationen: Korrigierbare und nicht korrigierbare Lesefehler. Nicht korrigierbare Lesefehler können seitens des Atari eine Fehlermeldung hervorrufen, da es in diesem Fall nicht möglich war, die gewünschten Daten korrekt zu lesen. Die meisten Treiber starten bis zu vier Leseversuche, bevor ein Fehler gemeldet wird.

Anders sieht es bei korrigierbaren Lesefehlern aus. Hier gab es zwar Probleme beim Lesen, aber es war dem Controller dennoch möglich, z. B. durch einen zusätzlichen Leseversuch oder über die Prüfsumme der Daten, einen fehlerfreien Datensektor zu erhalten. Korrigierbare Lesefehler weisen jedoch trotz fehlender Fehlermeldung darauf hin, dass es in naher Zukunft Probleme mit den Daten auf der Festplatte geben

könnte. Vom korrigierbaren bis zum nicht mehr korrigierbaren Lesefehler muss es kein großer Schritt sein.

17.9. SCSI-, IDE/ATAPI- und USB-Peripherie am Atari

17.9.1. Geräte am ACSI- und SCSI-Bus

Atari ST, MegaST^{E} und TT sind mit dem sogenannten ACSI-Bus (Atari-Computer System Interface) ausgestattet, über den Geräte wie Festplatten, Speicherkarten und CD-ROM/DVD/BD-Laufwerke angesprochen werden können. Der ACSI-Bus ist verwandt mit dem SCSI-Bus (Small Computer System Interface). Allerdings erfüllt der ACSI-Bus nicht die SCSI-Spezifikation, so dass man manche SCSI-Geräte nur eingeschränkt nutzen kann. Unter anderem ist die nutzbare Kapazität einer Platte am ACSI-Bus im Normalfall auf 1 Gigabyte beschränkt. Abhilfe schafft hier ein geeigneter SCSI-Hostadapter wie der LINK96/97 oder die Adapter von Inventronik in Verbindung mit einem Festplattentreiber wie HDDRIVER. Lösungen wie SCSI2Pi, PiSCSI oder BlueSCSI besitzen ebenfalls keine Einschränkungen. Der Hostadapter passt das SCSI-Busprotokoll an Ataris ACSI-Hardware an.

Beim TT sowie STs oder Falcons mit Fast-RAM-Eweiterung haben Geräte am ACSI-Bus keinen direkten Zugriff auf das TT-kompatible RAM. Daher muss die Datenübertragung intern über das ST-kompatible RAM erfolgen, was einen gewissen Zeitverlust mit sich bringt. Der ACSI-Bus ist aber ohnehin nicht für hohe Übertragungsraten geeignet. Bei einigen TTs treten sogar Übertragungsfehler auf, wenn eine schnelle SCSI-Platte am ACSI-Bus betrieben wird.

TT und Falcon besitzen einen handelsüblichen SCSI-Bus ohne Einschränkungen. Es lässt sich beliebige 8-Bit-SCSI-Peripherie anschließen, bevorzugt SCSI-2 oder neuer.

17.9.2. IDE-Festplatten

IDE-Festplatten (auch als PATA, Parallel ATA, bekannt) besitzen zwar nicht die Intelligenz von SCSI-Geräten, waren dafür aber preisgünstiger. Im Gegensatz zu SCSI-Platten arbeiteten sehr alte IDE-Platten beim Lesen und Schreiben von Sektoren nicht mit physikalischen Sektornummern, sondern benötigten zusätzliche Angaben über Spur und Zylinder, auf dem ein Sektor zu finden ist. Neuere IDE-Platten und alle SATA-Platten dagegen erlauben den Zugriff auch über physikalische Sektornummern (LBA). Nur mit Festplattentreibern wie HDDRIVER, die LBA unterstützen, lassen sich bei IDE- und SATA-Platten mehr als 8 GByte Kapazität nutzen.

Bei Atari-Computern finden IDE-Platten als interne Festplatten im STBook und Falcon030 Verwendung. Auch die meisten Atari-Clones unterstützen IDE/SATA-Peripherie. An den ST und TT lassen sich IDE-Platten über Hardware-Zusätze von Fremdherstellern anschließen. Außerdem sind Speicherkarten-Adapter (CF und SD) auf IDE-Basis weit verbreitet. IDE-Platten lassen sich mit speziellen Adaptern auch am SCSI-Bus betreiben. Die erforderlichen Adapter sind allerdings relativ teuer.

17.9.3. ATAPI- und SATA-Peripherie

Ebenfalls an die IDE-Schnittstelle werden ATAPI- und SATA-Geräte angeschlossen. Bis auf Festplatten handelt es sich bei Geräten an der IDE-Schnittstelle stets um solche, die den ATAPI-Standard erfüllen,

insbesondere CD-ROM-Laufwerke oder CD-Brenner. ATAPI- und optische SATA-Geräte werden auf Software-Ebene ähnlich angesteuert wie SCSI-Peripherie. SATA-Geräte lassen sich am IDE-Port über preisgünstige Adapter betreiben. Weniger preiswert, aber technisch möglich, ist der Betrieb von SATA-Geräten am SCSI-Bus. ATAPI- und SATA-Geräte lassen sich in Verbindung mit HDDRIVER unter DISKUS wie SCSI-Geräte ansprechen.

17.10. USB-Festplatten

Inzwischen gibt es auch mehrere USB-Hardware-Erweiterungen für den Atari. Falls für die entsprechende Hardware ein SCSI-Treiber exististiert (USB-Storage-Treiber), kann DISKUS ohne Einschränkungen auf die angeschlossenen USB-Platten zugreifen.

17.11. Hinweise zum Kauf von Peripheriegeräten

Moderne Festplatten und Speicherkarten sind so schnell, dass ihre maximale Übertragungsrate beim Anschluss an Atari-Computer längst nicht erreicht werden kann. Als Anhaltspunkt mag dienen, dass beim ACSI-Bus Werte von bis zu 1.4 MByte/s erreicht werden können. Der SCSI-Bus von Falcon030 und TT erlaubt ca. 1.8 MByte/s. Mit der IDE-Schnittstelle eines unbeschleunigten Falcon lassen sich ohne Zusatzhardware ca. 2.3 MByte/s erreichen, am Milan oder am TT mit IDE-Hardware-Erweiterung deutlich mehr. Alle diese Werte variieren je nach Gerät und hängen nicht zuletzt davon ab, mit welcher Software sie gemessen wurden. Daher sollte man die Bedeutung solcher Angaben nicht überbewerten.

Grundsätzlich lassen sich Platten und Speicherkarten bis zu 2 TByte problemlos am Atari betreiben. Größere Platten lassen sich zwar noch über den SCSI-Treiber und DISKUS, aber nicht mehr direkt über TOS ansprechen, da TOS lediglich 32-Bit-Sektornummern unterstützt. Es ist zu beachten, dass der Atari einen 8 Bit breiten SCSI-Bus besitzt, so dass sich SCSI-Platten mit 16- oder 32-Bit-Interface nicht ohne weiteres anschließen lassen. Die Busbreite lässt sich bei solchen Platten durch handelsübliche Adapter auf 8 Bit reduzieren. SCSI-Platten mit einer Kapazität von mehr als einem GByte lassen sich am ST und STE nur mit geeigneten Hostadaptern und Treibern wie HDDRIVER in der vollen Kapazität nutzen. Im Idealfall kommt ein LINK96/97 oder ein Adapter von Inventronik zum Einsatz. Der Festplattentreiber AHDI kann grundsätzlich nur das erste GByte einer SCSI-Festplatte ansprechen, auch beim TT oder Falcon.

Am ACSI-Port des Atari können bis zu sieben Geräte betrieben werden, an der SCSI-Schnittstelle des Falcon oder TT sind sieben weitere Geräte möglich. Die Nutzung sogenannter LUNs (logische Untereinheiten), wie sie von HDDRIVER unterstützt werden, ermöglicht sogar noch mehr SCSI-Geräte. Pro IDE-Port lassen sich zwei Geräte (IDE, ATAPI oder SATA) betreiben. Neben Festplatten kommen auch Streamer, Scanner oder CD-ROM/DVD/BD-Laufwerke oder -Brenner in Frage. Um beim Anschluss mehrerer Geräte Fehler auszuschließen, muss besonders beim ACSI-Bus auf kurze Kabel geachtet werden. Wichtig bei SCSI-Peripherie ist, dass die physikalisch letzten Geräte, also die beiden Geräte am Ende des Kabels, terminiert sind. (Auch der Computer ist ein SCSI-Gerät.) Die restlichen Geräte dürfen nicht terminiert sein. Schließlich sind hochwertige SCSI-Kabel von großer Bedeutung.

Es gibt zahlreiche Speicherkarten-Lösungen für den ACSI-Bus. Hier sollte man nicht unbedingt zur

billigsten Variante greifen. Billiglösungen sind oft nur eingeschränkt kompatibel zu den relevanten Spezifikationen. Außerdem werden Medienwechsel unter Umständen nicht automatisch erkannt, was zu Datenverlusten führen kann.

Beim Anschluss von ACSI-Geräten an den ST ist zu beachten, dass je nach Hostadapter nur Geräte betrieben werden können, die ohne Paritätsprüfung arbeiten. Häufig sind Platten von Werk aus auf Paritätsprüfung eingestellt. Diese Einstellung lässt sich manchmal über einen Jumper oder per Software mit DISKUS deaktivieren. Um die Paritätsprüfung per Software abzuschalten, muss eine Platte zu diesem Zweck kurzzeitig an einen Falcon oder TT angeschlossen werden. Erst nach der Neukonfigurierung wird sie vom ST erkannt. Hostadapter neueren Typs haben keine Probleme, was die Paritätsprüfung angeht.

Manche SCSI-Geräte benötigen einen Treiber wie HDDRIVER, der die sogenannte Initiator-Identifizierung unterstützt, bei der auch der Computer eine SCSI-ID erhält. Dies ist allerdings nur am SCSI-Bus möglich, nicht aber am ACSI-Bus. Platten, die Initiator-Identifizierung voraussetzen, lassen sich am ST nur mit dem Hostadapter LINK96/97 oder den Adaptern von Inventronik verwenden, weil diese Adapter die Initiator-Identifizierung durch spezielle Hardware realisieren. Booten kann TOS von solchen Platten bei TT und Falcon nicht.

Dass es neue SCSI-Platten schon lange nicht mehr zu kaufen gibt, ist heutzutage überhaupt kein Problem. Es existieren zahlreiche Hardware-Lösungen, die SCSI-Geräte emulieren, beispielsweise SCSI2Pi oder PiSCSI für den Raspberry Pi sowie BlueSCSI. Für den Atari besteht kein Unterschied zwischen diesen Emulationen und echter SCSI-Hardware, denn es handelt es sich bei dieser speziellen Hardware um echte SCSI-Geräte, auch wenn diese anders aussehen als handelsübliche Massenspeicher.

Index

Bios Parameter Block 49

Block-Menü

- Alles markieren 30
- Ausschneiden 30
- Einfügen 30
- Kopieren 30
- Überschreiben 30

Bootsektor 45

BPB 49

CBHD 43

Cluster 44

Cursortasten 41

Datei-Menü

- Eintrag löschen 17
- Eliminieren 15
- Ende 17
- Information 14
- Kopieren 15
- Leerer Eintrag 17
- Löschen 15
- Neuer Ordner 17
- Öffnen 15
- Programmflags 16
- Verketten 15
- Vertauschen 16

Datei/Ordner restaurieren 34

Dateiattribute 5

Dateiauswahlbox 12

Dateien sortieren 38

Dateisysteme 50

Datenbanken 6

Datenverlust 32, 51, 57

DD-Disketten 44

Defragmentierung 39

Desktop 2

- Bootsektor 3
- BPB ungültig 9
- Cluster belegt 8
- Cursorposition 11
- Directory 5
- FAT1 3
- FAT2 4
- Harddisk 8
- Laufwerke 9
- Laufwerkskennung 11
- log. Cluster 6
- phys. Cluster 7
- Position 10
- Schreibschutz 9
- Sektor 7

Disk-Menü

- Bootsektor 20
- Format definieren 21
- Formatieren 21

Information 18
Kopieren 19
Löschen 20
Sektorgröße ändern 22
Struktur 18
Diskbelegungstabelle 46
Diskettenformat 44
Diskettenkapazität 58
Diskettentyp 44
DOS-Emulator 19
ED-Disketten 44
ED-Laufwerke 59
Editieren 2
EXT2FS 50
FAT 46
FAT-Fenster 3
FAT-Prototyp erzeugen 36
FAT32 50
Harddisk-Menü
Busse neu scannen 23
Geräteauswahl 23
Geräteinformation 23
Kommando senden 24
Partition initialisieren 27
Partitionen 25
Rootsektor 25
Sektoren kopieren 27
HD-Disketten 44
HD-Laufwerke 58
HDDRIVER 10, 23, 26, 42f., 65
Interleave 59
Klemmbrett 30
Lesefehler 19
Linux 50
MagiC 42
MagiCMac 43
MagiCPC 43
Milan 23, 43
MiNT 42
Ordner 48
Partitionen 62
Partitionierung 62
PC-Kompatibilität 59
Peripherie 64
Plattenstatistik 64
Puffer 41
Reparieren 54
Rootsektor 61
Schriftgröße 2
SCSI-Treiber 23, 29, 31, 43
SCSIDRV 43
Sektor zurückholen 36
Sektor-Menü
Physikalisch lesen/schreiben 29
Sektorfolge laden/speichern 29
Sektorfenster 3
Sonderzeichen 2
Spezial-Menü
Daten drucken 31
Daten laden/speichern 31

Daten optimieren 37
Daten retten 34
Daten schützen 36
Daten suchen 31
Daten testen 32
FATs schreiben 39
Medium kopieren 37
Optionen 40
Umrechnung 39
Weitersuchen 32
Suchoperationen 6
Tastaturbelegung 41
Verzeichnisse aufräumen 38
VFAT 50
Wurzelverzeichnis 47
Wurzelverzeichnis rekonstruieren 35
XHDI 10