Dokumentation zum Ramdisk-XFS

Thomas Binder

COLLABORATORS			
	TITLE : Dokumentation zum Ramdisk-XFS		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Thomas Binder	December 17, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Dokumentation zum Ramdisk-XFS

1.1 Dokumentation zum Ramdisk-XFS fr MagiC

```
                Vorlufige Dokumentation zum Ramdisk-XFS vom @:"VERSION", Stand: @: ↩
                   DOCVERSION

Allgemeines
 Was ist das Ramdisk-XFS?
 Was kann das Ramdisk-XFS?
 Was kann das Ramdisk-XFS nicht?
 Installation
 Die Zusatzprogramme
 Bekannte Probleme
 Fragen und Antworten
 Zum Kopieren des Ramdisk-XFS
 Rechtliches
 Danksagungen

                Wer hat's verbrochen?
                 History

Technisches
 Die Datei RAMDISK.INF
 Die Debug-Version
 Die Sourcecodes
 Bemerkungen
```

1.2 was ist das ramdisk-xfs?

```
Was ist das Ramdisk-XFS?                                        Ramdisk-XFS
Das Ramdisk-XFS ist ein externes Filesystem fr MagiC ab Version 3 und
MagiCMac, wobei die Nutzung unter MagiC 4 bzw. MagiCMac ab Version
1.2.5 vorzuziehen ist (mehr dazu unter Bekannte Probleme und
Bemerkungen). Es stellt eine schnelle Ramdisk zur Verfgung, die
gegenber einer konventionellen Vertreterin einige Vorteile hat.

Durch die Untersttzung von langen Dateinamen mit der Untscheidung von
```

Gro- und Kleinschreibung erhlt man die Mglichkeit, eigene Programme in dieser Hinsicht zu testen, ohne dafr langsame Disketten nehmen oder eine Festplattenpartition opfern zu mssen.

Auch der Nur-Anwender kann von den Eigenschaften des Ramdisk-XFS profitieren, da es sich problemlos als Verzeichnis fr temporre Dateien nutzen lt, auf das schnell zugegriffen wird und das sich in der Gre dem jeweiligen Inhalt anpat.

Darberhinaus stellt das Ramdisk-XFS auch ein Beispiel fr die Programmierung von externen Filesystemen fr MagiC mit Hilfe des Pure-C-Interfaces dar. Aus diesem Grund werden smtliche Quellcodes in kommentierter Form mitgeliefert.

1.3 was kann das ramdisk-xfs?

Was kann das Ramdisk-XFS? Ramdisk-XFS
Das Ramdisk-XFS hat folgende Eigenschaften:

- Dateinamen mit einer Lnge von bis zu 32 Zeichen
- Unterscheidung von Gro- und Kleinschreibung, die Datei Test ist also nicht identisch mit der Datei TEST
- Dynamische Speicherverwaltung, d.h. die Ramdisk belegt immer nur so viel Speicher, wie zur Aufnahme des aktuellen Inhalts ntig ist
- Belegt den Speicher nicht bis zum letzten Byte, sondern lt immer einen gewissen Speicherblock frei, dessen Gre frei whlbar ist
- Eingeschrnkte Untersttzung von Unix-Filemodi
- Symbolische Links
- Freie Whlbarkeit der Laufwerkskennung (selbstsuchend oder durch den Anwender festgelegt)
- Der Name, unter dem das Ramdisk-XFS in Laufwerk U angesprochen werden kann, lt sich bereits in der Konfigurationsdatei festlegen
- Benutzter Speichertyp kann flexibel festgelegt werden
- Untersttzung eines Volume Labels ("Diskettenname")

Die Geschwindigkeit hngt natrlich stark von der benutzten Hardware ab. Grundstzlich lt sich jedoch sagen, da das Ramdisk-XFS beim Lesen etwas schneller, beim Schreiben etwas langsamer als eine gewhnliche Ramdisk ist. (Die geringere Geschwindigkeit beim Schreiben hngt dabei mit der dynamischen Anforderung des Speichers zusammen.)

Als kleiner Anhaltspunkt hier die Daten des Ramdisk-XFS auf einem Falcon030 mit 4MB ST-Ram, MagiC 4.01 und 1024x768x16 Bildschirmauflsung (gemessen mit How-Fast): Lesen ca. 1.8 MB/s, Schreiben ca. 0,6 MB/s. Bei abgeschaltetem Videochip: Lesen ca. 3 MB/s, Schreiben ca 1 MB/s.

1.4 was kann das ramdisk-xfs nicht?

Was kann das Ramdisk-XFS nicht? Ramdisk-XFS
Leider ist das Ramdisk-XFS nicht ohne Einschrnkungen, folgende Dinge funktionieren nicht oder nicht richtig:

- Keinerlei Filesharing, eine Datei kann immer nur hchstens einmal
  geffnet sein. Es ist also noch nicht einmal mglich, eine Datei
  zweimal zum Lesen zu ffnen, selbst wenn es der gleiche Proze ist
  (das liegt aber hauptschlich daran, da die MagiC-Doku hier einen
  leicht konfusen Eindruck macht).
- Die Unix-Filemodi werden nur soweit ausgewertet, wie es MagiC zult.
  Da smtliche MiNT-Aufrufe, die sich mit User-IDs beschftigen, in
  MagiC nicht vorhanden sind, benutzt das Ramdisk-XFS einen Kompromi:
  Es werden nur die Zugriffsrechte des Eigners berprft, und zwar fr
  jeden User. Dies ist ntig, weil man unter MagiC im Prinzip stndig
  Superuser ist und man sonst keinen richtigen Schreibschutz benutzen
  knnte (fr den Superuser gelten unter Unix keine Einschrnkungen).
- Das Sticky-Bit fr Verzeichnisse wird ignoriert, weil MagiC ohnehin
  keine Gruppen-IDs untersttzt.
- Gleiches gilt fr die Bits "Set User-ID/Group-ID on execution".
  Allerdings mten die sowieso vom Kernel ausgewertet werden, was mit
  dem XFS-Konzept von MagiC wohl nicht mglich sein drfte.
- Durch die dynamische Speicheranforderung wird der freie Speicher
  unter Umstnden stark fragmentiert. Allerdings kann man ja dafr
  sorgen, da stets ein gengend groer Speicherblock am Stck
  freibleibt. Der Ramdisk selbst macht fragmentierter Speicher nichts
  aus, da alle Files aus kleinen verketteten Blcken bestehen.

Mehr dazu findet sich unter Bekannte Probleme und Bemerkungen.

Ich habe hier sicherlich auch einige Sachen vergessen, also bitte nicht
aufregen, wenn etwas nicht klappt, obwohl es nicht hier aufgefhrt ist.

1.5 installation

Installation des Ramdisk-XFS Ramdisk-XFS
Um das Ramdisk-XFS benutzen zu knnen, mu lediglich die Datei
ramdisk.xfs in das Verzeichnis \gemsys\magic\xtension des Bootlaufwerks
kopiert werden. Beim nchsten Booten ist die Ramdisk dann aktiv.
Zustzlich sollte man gleichzeitig im selben Verzeichnis eine Datei
namens ramdisk.inf anlegen und das XFS damit fr seine Zwecke
konfigurieren (eine Beispieldatei ist im Archiv enthalten). Unterlt
man dies, installiert sich die Ramdisk auf dem ersten freien
Laufwerksbuchstaben, benutzt bevorzugt TT-RAM, lt mindestens 512K am
Stck frei, erlaubt keine Umlaute in Filenamen und legt kein Default-
Volume-Label an.

Alternativ kann man ramdisk.xfs auch in ramdisk.tos umbennenen und
direkt vom Desktop aus starten. Auf diese Weise kann man das XFS auch
nur kurz antesten, ohne es gleich richtig installieren zu mssen.

1.6 die zusatzprogramme

Zusatzprogramme fr das Ramdisk-XFS Ramdisk-XFS
Neben den .xfs-Dateien finden sich auch zwei Tools im gleichnamigen
Ordner des Archivs:

CheckFree:
Chckfree.app gehrt in den Ordner \gemsys\magic\start (bzw. in den Ordner, in dem die Autostart-Applikationen liegen) und wartet nach dem Start lediglich fortlaufend auf die Nachricht AP_TERM, die bei einem Shutdown an alle laufenden Applikationen verschickt wird. Es prft dann, ob die in der Datei chckfree.inf angegebenen Laufwerke (siehe Beispieldatei) leer sind; falls nicht, wird der Shutdown verweigert, als Fehlercode erhlt man die Nummer des Laufwerks, das nicht leer war (1 = A, 2 = B, ...) Man kann also auf diese Weise sicherstellen, da sich keine Dateien mehr auf einem Ramdisk-Laufwerk befinden, bevor man einen Reset auslst oder MagiCMac beendet.

GarbageCollector:
Garbcoll.app ist ein Programm, das wirklich nichts weiter macht, als sich gleich wieder zu beenden. Sinnvoll wird es erst dann, wenn man es als Single-Start-Programm startet, weil dann der Desktop beendet und damit aller von ihm belegter Speicher freigegeben wird. Auf diese Weise knnen u.U. Speicherlcken, die nach umfangreichen Lschaktionen auf der Ramdisk entstanden sind, wieder geschlossen werden.

1.7 bekannte probleme

Bekannte Probleme Ramdisk-XFS
Folgende Probleme bei Benutzung des Ramdisk-XFS sind bekannt (siehe auch Einschrnkungen und Bemerkungen):

- Das Ramdisk-XFS zusammen mit Outside oder VRAM ist tdlich (wenn das XFS den alternativen Speicher benutzen darf), da bei jedem Schreiben von Dateien die verkettete Liste der bisher belegten Blcke durchgegangen werden mu. Outside bzw. VRAM mu dann im schlimmsten Fall jedesmal auslagern, das (stckweise) Schreiben von lngeren Dateien kann dann unter Umstnden mehrere Minuten bis Stunden dauern.
- Beim Erweitern von Verzeichnissen werden die Index-Zeiger von ".." in den Strukturen der untergeordneten Directories nicht angepat. Das hat zwar prinzipiell keine schdlichen Auswirkungen auf den normalen Betrieb, sollte aber dennoch erwhnt werden.
- MagiC 3 und MagiCMac vor Version 1.2.5 untersttzen noch nicht den Systemaufruf Pdomain(), mit der das XFS feststellen kann, ob das laufende Programm korrekt mit langen Dateinamen und Unterscheidung von Gro/Kleinschreibung umgehen kann. Daher kann es bei diesen Systemen fters passieren, da Dateinamen bunt gewrfelt komplett gro oder komplett kleingeschrieben sind oder da ein Programm eine Datei nicht findet, eine falsche findet oder eine neue nicht anlegen kann. Wer also das Ramdisk-XFS sinnvoll einsetzen will, sollte besser aktuelle Versionen von MagiC bzw. MagiCMac verwenden.
- Thing (und viele andere Programme, die in den meisten Fllen mit den MiNTLibs gelinkt wurden) schalten nur unter MiNT mittels Pdomain() in die MiNT-Domain, um anzuzeigen, da sie lange Dateinamen korrekt behandeln. Als Folge zeigt Thing zwar korrekt lange Namen an, von ihm neu angelegte Dateien und Ordner werden aber immer komplett in Kleinbuchstaben gewandelt. Bei anderen Programmen, z.B. leider auch bei der LZHShell, fhrt das dazu, da verstmmelte Dateinamen benutzt werden, obwohl das Programm es eigentlich besser knnte.
- Ebenfalls im Zusammenhang mit dem letztgenannten Problem: Das

Hilfsprogramm MGCOPY des MagXDesks luft ebenfalls nicht in der MiNT-Domain, daher gibt es auch hier nur kleingeschriebene neue Dateien und u.U. Probleme mit angeblich bereits existierenden Dateien.
- Gemini kann keine Dateien auf der Ramdisk anlegen, weil es immer den grten freien Speicherblock als Puffer benutzt und so nichts mehr fr das XFS brig bleibt. Eingeschrnkte Abhilfe schafft hier das MagiC-Utility LimitMem; besser wre es jedoch, Gemini wrde beim Kopieren nicht mehr so rcksichtslos vorgehen.
- Das Fhren des Logfiles der Debug-Version scheint mit MagiCMac nicht zu funktionieren. Von leeren Logfiles bis hin zu blen Systemabstrzen kommt offensichtlich alles vor. Unter MagiC 4 dagegen funktioniert es einwandfrei (das ist zumindest meine Erfahrung).

Auch hier kann es gut sein, da ich noch den einen oder anderen Punkt vergessen habe. Ich werde diese Liste also bei Bedarf noch erweitern und bin fr Hinweise immer dankbar.

1.8 fragen und antworten

Fragen und Antworten Ramdisk-XFS
Alles etwas bunt durcheinander gewrfelt, ich hoffe, es geht trotzdem...

F: Was fange ich mit diesen .xfs-Dateien an? Wohin mu ich die kopieren, oder wie rufe ich sie auf?
A: Wie beim Punkt Installation beschrieben, mu ramdisk.xfs in den Ordner \gemsys\magic\xtension des Bootlaufwerks kopiert werden. Die Datei ramdebug.xfs braucht man nur, wenn man Debug-Ausgaben protokollieren und mir zuschicken mchte.

F: Thing legt, obwohl ich MagiC 4/MagiCMac 1.2.5 benutze, nur kleingeschriebene Dateinamen an. Auch Umbennenen mit Grobuchstaben klappt nicht.
A: Thing benutzt unter MagiC leider nicht den Systemaufruf Pdomain(), um mitzuteilen, da er lange Dateinamen und die Unterscheidung von Gro-und Kleinschreibung beherrscht. Daher wandelt das XFS alle Dateinamen in Kleinbuchstaben um, da von solchen Programmen in der Regel komplett gro geschriebene Filenamen benutzt werden, was nicht so toll aussehen wrde.

F: Ich mchte mit Kobold arbeiten, mu ich dazu den GEMDOS-Modus fr das Laufwerk des Ramdisk-XFS einschalten?
A: Selbstverstndlich, denn intern werden komplett andere Strukturen als auf FAT-Filesystemen benutzt. Auf jeden Fall sollte man nach Mglichkeit aktuelle Versionen von Kobold benutzen, da diese meines Wissens mit erweiterten Filesystemen zurechtkommen.

F: Wenn ich mit MagiC 3 Dateien von einem FAT-Laufwerk mit dem Kobold auf das Ramdisk-XFS-Laufwerk kopiere, erscheinen sie alle komplett gro geschrieben. Mache ich das aber mit Thing, haben sie alle nur Kleinbuchstaben. Woran liegt das?
A: Da MagiC 3 den Systemaufruf Pdomain() noch nicht bereitstellt, kann das XFS nicht feststellen, ob das laufende Programm korrekt mit langen Dateinamen und der Unterscheidung von Gro/Kleinschreibung umgehen kann. Daher werden alle Namen unverndert bernommen. Kobold benutzt im GEMDOS-Modus immer komplett in Grobuchstaben gehaltene

```
   Dateinamen, whrend Thing merkt, da das Ziellaufwerk die
   Schreibweise unterscheidet und daher eine Wandlung in
   Kleinbuchstaben vornimmt.

F: Das Ramdisk-XFS untersttzt angeblich lange Dateinamen mit
   Unterscheidung von Gro- und Kleinschreibung. Mein Desktop zeigt
   aber immer nur normale 8+3-Namen an, die komplett gro geschrieben
   sind. Warum?
A: Der Desktop kommt offensichtlich nicht mit langen Dateinamen zurecht
   und benutzt die alte Methode zum Lesen von Verzeichnissen, die
   prinzipbedingt nur verstmmelte Dateinamen liefert.

F: Warum kann Gemini keine Dateien auf das Ramdisk-XFS kopieren?
A: Weil Gemini immer den grten verfgbaren Speicherblock als
   Kopierpuffer benutzt und somit dem XFS nichts mehr brig lt.
   Mgliche Abhilfe: Gemini mittels LimitMem weniger Speicher zur
   Verfgung stellen.

F: Warum kann das Programm XYZ nichts auf dem Ramdisk-XFS abspeichern?
A: In aller Regel aus dem gleichen Grund wie eine Frage vorher: Das
   Programm belegt allen freien Speicher und lt der Ramdisk daher
   keinen Platz fr neue Dateien mehr. Auch hier kann man unter
   Umstnden mit LimitMem Abhilfe schaffen.

Werden bestimmt noch mehr...
```

1.9 zum kopieren des ramdisk-xfs

```
Zum Kopieren                                                Ramdisk-XFS
Das Ramdisk-XFS ist Freeware, darf und soll also ohne Bedenken benutzt
und kostenlos weitergegeben werden (der Vertrieb ber PD-Hndler ist
gestattet, solange der Preis pro Diskette DM 10 nicht bersteigt). Auch
der Upload in Mailboxen ist erlaubt und erwnscht, wenn fr den
Download keine Gebhren anfallen (auer den Telefongebhren, natrlich).

Es mssen in jedem Fall alle Dateien unverndert und vollstndig
kopiert werden (Archivierung ist erlaubt). Da es nicht wenige sind,
hier die Auflistung, wie genau der zu kopierende Verzeichnisbaum
auszusehen hat:

pc_xfs\
    |_ atarierr.h
    |_ mgx_xfs.h
    |_ pc_xfs.h
    |_ pc_xfs.inc
    |_ pc_xfs.s
    |_ sogehts.txt
ramdisk\
    |_ proto.h
    |_ ramdebug.prj
    |_ ramdisk.c
    |_ ramdisk.h
    |_ ramdisk.prj
    |_ ramutil.c
    |_ version.h
```

```
tools\
    |_ chckfree.app
    |_ chckfree.inf
    |_ garbcoll.app
ramdebug.xfs
ramdisk.hyp
ramdisk.inf
ramdisk.ref
ramdisk.stg
ramdisk.xfs
```

Da viele Quellcodes enthalten sind, erfolgt die Originaldistribution als Archiv im Archiv. Das uere enthlt lediglich die Dateien archive.lzh und archive.sig. Erstere ist dabei das innere Archiv mit den oben aufgelisteten Dateien, und archive.sig ist ein PGP-Signaturfile fr archive.lzh. Auf diese Weise kann jeder nachprfen, ob die Dateien (insbesondere eben die Sourcecodes) wirklich direkt von mir stammen und nicht in irgendeiner Weise manipuliert wurden. Mein ffentlicher Schlssel ist selbstverstndlich auf Anfrage erhltlich.

Es ist nicht erlaubt, das Ramdisk-XFS einem anderen Produkt ohne meine ausdrckliche schriftliche Genehmigung beizulegen. Wer so etwas vorhat, darf sich ruhig bei mir melden, in aller Regel werde ich die Zustimmung nicht verweigern.

Das Ramdisk-XFS hat einen nicht unerheblichen Teil meiner Freizeit beansprucht; trotzdem habe ich es als Freeware inklusive aller Quelltexte verffentlicht. Ich bitte daher alle, die es regelmig benutzen, mir meine Arbeit durch eine kleine Spende zu honorieren, was auch die Weiterentwicklung des XFS sichert.

1.10 rechtliches

Rechtliches Ramdisk-XFS

Das Ramdisk-XFS und die dazugehrenden Utilities wurden mit groer Sorgfalt entwickelt und eingehend getestet. Dennoch kann nicht ausgeschlossen werden, da sie noch Fehler enthalten. Ich kann daher weder die einwandfreie Funktionsfhigkeit, noch die Tauglichkeit zu einem bestimmten Zweck garantieren.

Desweiteren erfolgt die Benutzung des Ramdisk-XFS und der beiliegenden Utilities auf eigene Gefahr! Ich, Thomas Binder, bernehme keinerlei Haftung fr Schden jeglicher Art, die direkt oder indirekt durch die sach- oder unsachgeme Anwendung des Ramdisk-XFS und/oder der Utilities entstanden sind oder entstanden sein knnten.

Jegliche nderung an den Sourcecodes, auer fr den Privatgebrauch, ist untersagt. Daraus folgt direkt, da vernderte Quellen nicht verbreitet werden drfen. Wer etwas verbessert oder erweitert hat, darf mir gerne ein Diff schicken; sinnvolle Sachen nehme ich gerne auf, natrlich mit gebhrender Erwhnung des jeweiligen Autors.

Wer Teile der Sourcen fr ein eigenes XFS bernehmen will, kann dies gerne tun, mu dann aber ausdrcklich darauf hinweisen. Dies sollte in der Doku, in der Einschaltmeldung, und, bei Sourcedistributionen, auch

an den entsprechenden Stellen im Quellcode geschehen.

1.11 danksagungen

Danksagungen Ramdisk-XFS
Ich danke den folgenden Leuten, ohne deren Untersttzung das Ramdisk-XFS nicht in der jetzigen Form existieren wrde (alphabetische Reihenfolge):

- Oliver Buchmann
  Fr das Weiterleiten von Interessenten.

- Alexander Clauss
  Fr das Betatesten.

- Stefan Damerau
  Fr das Betatesten.

- Phillip Frank
  Fr das Betatesten.

- Dirk Hagedorn
  Fr das Betatesten.

- Gtz Hoffart
  Fr das Betatesten und seine Vorschlge.

- Dirk Klemmt
  Fr das Betatesten.

- Andreas Kromke
  Fr die MagiC-Programmiererdoku (auch wenn sie in vielen Punkten besser sein knnte) und seine Hilfe bei Problemen.

- Egbert Mattikau
  Fr das Betatesten.

- Thomas Much
  Fr das Betatesten (speziell mit Texel).

- Rainer Riedl
  Fr das Betatesten.

- Frank Rske
  Fr das Betatesten und einige Vorschlge.

- Peter Rosengatter
  Fr das Betatesten.

- Herwig Schelauske
  Fr das Betatesten.

- Sren Schnee
  Fr das Betatesten.

- Uwe Seimet
  Fr das Betatesten.

- Manfred Ssykor
  Fr das Betatesten.

- Arno Welzel
  Fr das Betatesten.

1.12 autor

Autor Ramdisk-XFS
Wer einen Fehler gefunden hat, Vorschlge, Verbesserungen oder Kritik
loswerden mchte, meinen ffentlichen PGP-Schlssel braucht oder
einfach nur mit mir "reden" will, kann diese Adresse benutzen:

Thomas Binder
Johann-Valentin-May-Strae 7
64665 Alsbach-Hhnlein
Deutschland

Per EMail bin ich im InterNet und im Mausnetz ber folgende Adressen
erreichbar:

binder@rbg.informatik.th-darmstadt.de
gryf@hrz.th-darmstadt.de (wird nicht so oft geprft)
Thomas Binder @ HD (ich tausche nur Mittwochs und Samstags)

Wer ab und zu (oder hufiger ;)) im IRC hngt, sollte nach Gryf gucken.

Wer das Ramdisk-XFS hufiger oder regelmig benutzt, sollte so fair
sein und meine Arbeit durch eine kleine Spende honorieren. Meine
Bankverbindung:

Dresdner Bank AG Frankfurt am Main
Konto-Nummer: 9 024 050 00
Bankleitzahl: 500 800 00

Vielen Dank!

1.13 history

History Ramdisk-XFS
Release 1 vom 23.04.1996

1.14 release 1

Release 1 vom 23.04.1996 Ramdisk-XFS
- Erste ffentliche Version, leider noch mit ein paar Einschrnkungen

1.15 die datei ramdisk.inf

```
Konfiguration mit Hilfe der Datei RAMDISK.INF                Ramdisk-XFS
Das Verhalten des Ramdisk-XFS lt sich weitreichend ber eine
Konfigurationsdatei bestimmen. Der Name dieser Datei bildet sich aus
dem Basisnamen, unter dem das XFS gestartet wurde, plus der Endung
".inf". Lautet der Dateiname des Filesystems also ramdisk.xfs, wird
nach der Datei ramdisk.inf gesucht, was der bliche Fall sein sollte.

Die Datei wird zunchst im Ordner \gemsys\magic\xtension des aktuellen
Laufwerks gesucht, danach in dessen Wurzelverzeichnis und schlielich
noch im aktuellen Verzeichnis. Wird sie nirgendwo gefunden, werden
sinnvolle Defaulteinstellungen benutzt.

Bei der Konfigurationsdatei handelt es sich um eine simple ASCII-Datei,
in der drei Typen von Zeilen vorkommen knnen:

1. Leerzeilen (sollte klar sein...)
2. Kommentarzeilen; das sind alle Zeilen, die mit einem Doppelkreuz (#)
   beginnen
3. Konfigurationszeilen, die immer das Format Variable=Wert haben mssen

Alle anderen Zeilen sind fehlerhaft und werden gemeldet; das gilt auch
fr Konfigurationszeilen mit falschen Parametern. Das XFS wartet in
solchen Fllen auf einen Tastendruck, damit man die fehlerhafte Zeile
in Ruhe lesen kann.

Folgende Variablen knnen in den Konfigurationszeilen vorkommen
(Gro- und Kleinschreibung wird nicht unterschieden):

drive     - Legt das Laufwerk fest, auf dem sich die Ramdisk installiert
mountname - Legt den Verzeichnisnamen fest, unter dem die Ramdisk auf
            Laufwerk U zu finden ist
label     - Default-"Diskettenname" fr das Ramdisk-XFS
ramtype   - Bestimmt den zu benutzenden Speichertyp
leavefree - Gre des mindestens freizuhaltenden Speicherblocks
8bit      - Legt fest, ob ASCII-Zeichen > 127 (also Umlaute und andere
            Sonderzeichen) in Dateinamen erlaubt sind

Die Defaulteinstellungen
```

1.16 Die Variable drive

```
drive                                                        Ramdisk-XFS
Defaultwert: Undefiniert, das XFS sucht sich die erste freie Kennung

Mit Hilfe der Variablen drive kann man festlegen, welches BIOS-Laufwerk
das Ramdisk-XFS benutzen soll (leider ist es mit der momentanen Version
der XFS-Schnittstelle von MagiC nicht mglich, ohne auszukommen, da
man nicht direkt mounten kann).

Als Wert wird einfach der gewnschte Laufwerksbuchstabe angegeben, also
z.B. drive=Z. Enthlt die Datei ramdisk.inf keine drive-Zeile, sucht
sich das XFS selbstttig das erste freie BIOS-Laufwerk und belegt
```

dieses. Sollte keines mehr frei oder das durch drive festgelegte
bereits vergeben sein, erhlt man eine entsprechende Meldung, das XFS
installiert sich dann natrlich nicht.

1.17 Die Variable mountname

mountname Ramdisk-XFS
Defaultwert: Laufwerksbuchstabe

Unter MagiC sind alle BIOS-Laufwerke auch als Unterverzeichnisse von
Laufwerk U ansprechbar, das Laufwerk J findet sich also auch im
Directory U:\j. Beim Ramdisk-XFS kann man nun gezielt einen anderen
Namen fr dieses Verzeichnis whlen, der dann bei mountname angegeben
werden mu, also z.B. mountname=RamXFS. Fehlt diese Zeile, heit das
Verzeichnis wie gewhnlich und hngt dementsprechend davon ab, unter
welcher Laufwerkskennung sich das Ramdisk-XFS installiert hat.

Hinweis: Zur Zeit sind im Laufwerk U von MagiC nur 8+3-Namen in
Grobuchstaben mglich, d.h. das obige Beispiel (mountname=RamXFS)
wrde zu dem Verzeichnis U:\RAMXFS fhren.

1.18 Die Variable label

label Ramdisk-XFS
Defaultwert: Kein Label

Das Ramdisk-XFS untersttzt die von TOS-kompatiblen Laufwerken und ISO-
9660-CDROMs bekannten Volume Labels ("Diskettennamen"). Da diese
natrlich nicht dauerhaft gespeichert werden knnen, lt sich mit
Hilfe der Variablen label ein solcher Name bereits beim Start festlegen.

Das Label des Ramdisk-XFS darf selbstverstndlich auch bis zu 32
Zeichen lang sein (berlngen werden abgeschnitten) und Gro- und
Kleinbuchstaben enthalten. Leider benutzen viele Desktops noch nicht
den korrekten Betriebssystemaufruf (Dreadlabel()), um diesen Namen ohne
8+3-Verstmmelung ermitteln zu knnen, daher wird von diesen dann
beispielsweise nur "Ramdisk-" statt "Ramdisk-XFS-Label" angezeigt
(hnliches gilt natrlich auch fr das ndern des Labels).

1.19 Die Variable ramtype

ramtype Ramdisk-XFS
Defaultwert: altorst

Mit Hilfe dieser Variablen kann man genau bestimmen, welchen
Speichertyp das Ramdisk-XFS fr seine Daten benutzen soll. Folgende
Werte sind mglich:

stonly: Nur ST-kompatibles RAM
altonly: Nur Alternate RAM ("Fast RAM")

```
storalt: Egal, aber lieber ST-kompatibel
altorst: Egal, aber lieber Alternate RAM
```

1.20 Die Variable leavefree

```
leavefree                                                    Ramdisk-XFS
Defaultwert: 512

Das Ramdisk-XFS achtet immer darauf, da es nicht den kompletten freien
Speicher belegt und einen zusammenhngenden Speicherblock einer
gewissen Mindestgre freilt. Mit der Variablen leavefree kann nun
festgelegt werden, wieviele Kilobytes dies sein sollen. Mchte man
also, da das XFS immer mindestens 2 Megabyte am Stck fr andere
Programme freilt, mu man die Zeile leavefree=2048 in seine
ramdisk.inf aufnehmen.

Natrlich kann man hier auch 0 eintragen, dann darf man sich aber auch
nicht wundern, wenn pltzlich kein Speicher mehr frei ist und sich der
Desktop dann aus genau diesem Grund weigert, eine Datei von der Ramdisk
zu lschen...

Wenn die Ramdisk Speicher braucht, schaut sie zunchst nach, wie gro
der grte zusammenhngende Speicherblock des gewnschten RAM-Typs ist
(siehe auch ramtype). Danach wird der bentigte Speicher angefordert.
Diese Anforderung gilt nur dann als fehlgeschlagen, wenn eine der
folgenden Bedinungen erfllt ist:

- die Anforderung ist tatschlich fehlgeschlagen, es gibt also sowieso
  nicht mehr genug freien Speicher
- der grte zusammenhngende Speicherblock ist jetzt kleiner als bei
  leavefree angegeben
- der grte zusammenhngende Speicherblock war schon vor der
  Anforderung zu klein und ist jetzt noch kleiner geworden

In den beiden letzten Fllen wird der gerade angeforderte Speicher
wieder freigegeben.
```

1.21 Die Variable 8bit

```
8bit                                                         Ramdisk-XFS
Defaultwert: false

Mit der Variablen 8bit kann man festlegen, ob in Dateinamen auf dem
Ramdisk-XFS Umlaute und andere Sonderzeichen erlaubt sind, deren ASCII-
Code grer oder gleich 128 ist. Ist das gewnscht, mu man die Zeile
8bit=true in die ramdisk.inf aufnehmen.
```

1.22 die defaulteinstellungen

Defaultwerte der Variablen Ramdisk-XFS
Wenn eine bestimmte Variable in der Datei ramdisk.inf nicht auftaucht, werden passende Standardwerte angenommen, die hier nochmals in einer bersicht zu sehen sind. Fehlt die Datei ganz, werden alle hier stehenden Werte bernommen.

```
drive=<Erste freie BIOS-Laufwerkskennung>
mountname=<Buchstabe des durch drive festgelegten Laufwerks>
label=<keines vorhanden>
ramtype=altorst
leavefree=512
8bit=false
```

Werte in spitzen Klammern sind natrlich nur eine Umschreibung des tatschlichen Inhalts, der nicht direkt dargestellt werden kann.

1.23 die debug-version

Debug-Version Ramdisk-XFS
Wer glaubt, einen Fehler nachvollziehbar gefunden zu haben, kann mir durch das Zuschicken eines Debug-Protokolls viel Arbeit abnehmen. Zu diesem Zweck mu man statt ramdisk.xfs die Datei ramdebug.xfs installieren oder zustzlich starten. Diese Version gibt haufenweise Protokollausgaben in die Datei c:\gemsys\magic\xtension\ramdebug.log aus, was leider auch einen gewaltigen Geschwindigkeitseinbruch bedeutet.

Aus diesem Grund kann man in der INF-Datei, die dann natrlich ramdebug.inf heit, zustzlich die Variablen logfile und logback benutzen. Mit logfile lt sich der komplette Zugriffspfad (immer absolut und mit Laufwerksangabe!) fr das Logfile anlegen, ber logback kann ein weiterer Zugriffspfad angegeben werden, in den das alte Logfile beim Start umbenannt wird (mu sich auf dem gleichen physikalischen Laufwerk befinden!) Gibt man logfile alleine oder nach einer logback-Zeile an, wird kein Backup erzeugt. Auf den Drucker umlenken geht auch, dazu einfach U:\dev\prn fr logfile angeben (natrlich dann ohne logback).

Wer den Geschwindigkeitsverlust ertrglich halten will, kann das Logfile auf eine andere Ramdisk legen, auf keinen Fall aber auf die Debug-Ramdisk selbst, sonst gibt es eine Endlosrekursion!

Das Logfile kann zwischenzeitlich bedenkenlos gelscht werden, wenn man den ganzen Krempel, der durch das Anlegen eventueller Testdateien entstanden ist, loswerden will, bevor man den eigentlichen Fehler reproduziert.

Hlt man beim Start der Debug-Version eine der Umschalttasten gedrckt, werden alle Protokollausgaben direkt auf den Bildschirm geschrieben.

Wichtig: Mit MagiCMac scheint das protokollieren der Debug-Ausgaben leider berhaupt nicht zu funktionieren, mal hngt sich das System auf, mal ist die Datei leer. Die Ausgabe auf den Bildschirm scheint jedoch zu klappen.

1.24 die sourcecodes

Die Sourcecodes Ramdisk-XFS
Wie bereits an anderer Stelle erwhnt, soll das Ramdisk-XFS auch als Grundlage fr interessierte Programmierer dienen, die prinzipiell ein Filesystem fr MagiC programmieren wollen, dies aber wegen mangelnder Beispiele (das CDROM-XFS gibt es ja nur als Binary) bisher nicht in Angriff genommen haben.

Darberhinaus sind die Quellcodes auch ein Beispiel fr die Benutzung der Pure-C-Schnittstelle, mit der die Entwicklung eines Filesystems fr MagiC deutlich bequemer wird. Assembler in allen Ehren, aber ich wollte keinen ext2fs-Treiber in Maschinensprache basteln mssen...

Damit die Sourcen auch wirklich hilfreich sind, habe ich sie ausfhrlich kommentiert (vielleicht habe ich dabei auch etwas bertrieben...) Zustzlich zu den reinen Kommentaren, was bestimmte Codeabschnitte bewirken, finden sich viele Tips und Anmerkungen fr die Programmierung eigener Filesysteme fr MagiC.

Wer will, kann Teile der Sourcen bernehmen, mu das dann aber auch an passender Stelle (Doku, Einschaltmeldung und bei Source-Distributionen auch in den Quellen) erwhnen. Eine nderung der Sourcen bzw. eine Komplettbernahme fr ein eigenes Ramdisk-XFS ist jedoch nicht gestattet! Nheres hierzu im Abschnitt Rechtliches.

1.25 bemerkungen

Bemerkungen Ramdisk-XFS
Hier finden sich einige Anmerkungen zu bestimmten Verhaltensweisen des Ramdisk-XFS.

Konvertierung der Filenamen fr Fsfirst()/Fsnext().

Suchen von Dateien

Verwaltung der Zugriffsrechte

1.26 konvertierung der filenamen

Behandlung von Dateinamen bei Fsfirst()/Fsnext() Ramdisk-XFS
Die Systemaufrufe Fsfirst()/Fsnext() (und Dreaddir() im sogenannten Kompatibilittsmodus) liefern Filenamen, die denen auf FAT-Filesystemen entsprechen; sie haben also maximal acht Zeichen Prefix, eventuell einen Punkt und dahinter maximal drei Zeichen Suffix (alles in Grobuchstaben). Da ein XFS fr MagiC diese Konvertierung selbst vornehmen mu, finden sich hier die Regeln, nach denen das Ramdisk-XFS diese Umwandlung der Dateinamen erreicht. Sie ist brigens auch fr das Suchen von Dateien von Bedeutung, um auch lteren Programmen die Chance zu geben, Files zu finden.

1. Die Filenamen "." fr das aktuelle und ".." fr das bergeordnete

```
   Verzeichnis werden direkt bernommen
2. Alle Zeichen werden in Grobuchstaben gewandelt
3. Alle in FAT-Dateinamen nicht erlaubten Zeichen werden in ein X
   umgewandelt (betrifft insbesondere Leerzeichen)
4. Alle Punkte im Filenamen, auer dem letzten, werden in ein Komma
   verwandelt; ist der letzte Punkt gleichzeitig das letzte Zeichen des
   Filenamens, wird er ignoriert (zhlt aber trotzdem als letzter Punkt)
5. Ist das erste Zeichen ein Punkt und dies ist der einzige Punkt im
   Filenamen, wird er doch in ein Komma verwandelt
6. Die ersten acht Zeichen vor dem letzten Punkt werden bernommen
   (wenn es keine acht sind, entsprechend weniger)
7. Die ersten drei Zeichen nach dem letzten Punkt (sofern einer
   existiert) werden mitsamt dem Punkt bernommen

Ein paar Beispiele:

"." -> "." (Regel 1)
"Sehr_langer_Dateiname" -> "SEHR_LAN" (Regeln 2 und 6)
"Mein Bild.tiff" -> "MEINXBIL.TIF" (Regeln 2, 3, 6 und 7)
".profile" -> ",PROFILE" (Regeln 2, 5 und 6)
"Viele.Punkte.im.Namen" -> "VIELE,PU.NAM" (Regeln 2, 4, 6 und 7)
"Punkt am Ende." -> "PUNKTXAM" (Regeln 2, 3, 4 und 6)
```

1.27 suchen von dateien

```
Vorgehensweise bei der Dateisuche                                Ramdisk-XFS
Das Suchen von Dateien auf der Ramdisk ist unter MagiC 3 oder bei
Programmen, die in der TOS-Domain laufen, eine etwas schwierigere
Angelegenheit. Man erwartet schlielich, da ein altes Programm Dateien
ffnen kann, auch wenn sie durch die Dateiauswahlbox verstmmelt wurden.

Dazu ein Beispiel: Eine Datei auf der Ramdisk heit Beispielname.txt
und soll nun in einem lteren Programm geffnet werden. In der
Dateiauswahlbox wird der Name zu BEISPIEL.TXT und wird dementsprechend
so vom Programm gesucht. Wrde das XFS jetzt nicht auch mit 8+3-
Vergleichen arbeiten, knnte die Datei nicht geffnet werden.

Folgende Tabelle zeigt eine bersicht, wann das Ramdisk-XFS wie sucht.
Dabei wird zwischen "Zugriff" und "Existenz" unterschieden. Im ersten
Fall soll auf die Datei zugegriffen werden, im zweiten soll lediglich
festgestellt werden, ob bereits eine Datei gleichen Namens existiert.

                Suchen                          Existenz

     MagiC3 MagiC4-TOS MagiC4-MiNT   MagiC3 MagiC4-TOS MagiC4-MiNT
(1)     ja       ja         ja          ja      nein        ja
(2)    nein      ja        nein        nein      ja        nein
(3)     ja       ja        nein        nein     nein       nein

(1) = Direkter Vergleich der Filenamen ohne irgendwelche Umwandlungen
(2) = Vergleich der Filenamen in Kleinbuchstaben
(3) = Vergleich der Filenamen in 8+3-Schreibweise

TOS und MiNT hinter MagiC4 bezieht sich hier auf die Domain, in der das
suchende bzw. prfende Programm luft.
```

Wenn Methode (3) gebraucht wird, um eine Datei zu suchen, kann es passieren, da eine falsche ermittelt wird. Wenn es im obigen Beispiel noch eine Datei beispieldatei.txt gibt, deren Eintrag physikalisch vor dem von Beispielname.txt liegt, wird beim ffnen von BEISPIEL.TXT immer die erste gefunden, da sie auf 8+3-Basis bereits den gleichen Namen hat, auch wenn eigentlich die andere Datei gemeint war.

1.28 verwaltung der zugriffsrechte

Zugriffsrechte Ramdisk-XFS
Da das Ramdisk-XFS so weit wie mglich ein Unix-Filesystem nachbildet, MagiC aber bislang keinerlei Mehrbenutzerfhigkeit besitzt, werden die Zugriffsrechte etwas anders als beispielsweise von MiNT gewohnt ausgewertet.

Unter MagiC gibt es nur einen einzigen User, der damit auch "Superuser" ist. Dieser hat unter Unix normalerweise keinerlei Einschrnkungen im Bezug auf Dateizugriffe, kann also alle Dateien einsehen, verndern, neu anlegen, lschen, etc. (eine Ausnahme davon sind z.B. Dateien in einem AFS-Baum, was hier aber nicht weiter von Interesse ist).

Wrde das Ramdisk-XFS jetzt den "Superuser" wie gewohnt behandeln, knnte man berhaupt keine Zugriffsbeschrnkungen realisieren. Daher werden die Rechte immer berprft, und zwar die, die fr den Eigner eingetragen sind.

Auf diese Weise kann man mit Hilfe des GEMDOS-Calls Fchmod() den Zugriff auf Dateien und Verzeichnisse regeln. Lscht man beispielsweise das Schreibbit, knnen keine neuen Eintrge in dem Verzeichnis mehr angelegt werden. Ein gelschtes Lesebit verbietet es, Dateien zu lesen oder Verzeichnisse anzuzeigen.

Eine weitere Besonderheit ist das x-Bit. Bei Dateien zeigt es eigentlich nur an, ob es sich um ein ausfhrbares File handelt (auch Skripte). Es wird bei Programmen, die in der TOS-Domain laufen, automatisch fr neue Dateien gesetzt, die mit .tos, .ttp, .prg, .app, .gtp oder .acc enden.

Bei einem Verzeichnis hingegen gibt das x-Bit an, ob es "berschritten" werden darf, d.h. ob es in einer Pfadangabe als Teilstck vorkommen darf. Ein Beispiel: Ist das x-Bit des Directories Z:\foo\bar gelscht, kann man sich zwar den Inhalt anschauen, nicht jedoch auf Dateien oder Unterverzeichnisse zugreifen. Ein Versuch, die Datei Z:\foo\bar\foobar.txt zu ffnen, schlgt also fehl.

Auch das Lesebit bietet bei Verzeichnissen noch eine kleine berraschung: Ist es gelscht, das x-Bit aber gesetzt, kann man den Inhalt nicht mehr anzeigen lassen. Trotzdem kann man auf Dateien und Unterverzeichnisse, deren Namen man kennt, zugreifen. Ist also das Lesebit bei Z:\foo\bar gelscht (und gleichzeitig das x-Bit gesetzt), schlgt ein ls -l Z:\foo\bar fehl. Die Datei Z:\foo\bar\foobar.txt lt sich jedoch problemlos ffnen (sofern sie existiert und lesbar ist).

Der GEMDOS-Befehl Fattrib() erlaubt es, zumindest das Schreibbit (ber

das Nur-Lesen-Bit) von Dateien und Directories zu beeinflussen. Daneben wird auch das Archivbit fr Dateien untersttzt, obwohl es das auf Unix-Filesystemen nicht gibt. (Als "Ersatz" fhren diese, wie auch das Ramdisk-XFS, ein Datum der letzten Modifikation, was fr Backup-Zwecke weitaus besser geeignet ist, da es im Gegensatz zum Archivbit nach erfolgter Sicherung der Datei nicht manipuliert werden mu.)

Das Schreibbit verhlt sich aus Kompatibilittsgrnden leicht anders als unter Unix: Ist es nicht gesetzt, kann eine Datei nicht gelscht werden. Unter Unix wre das problemlos mglich, da die Zugriffsrechte des dazugehrigen Verzeichnisses entscheiden, ob eine Datei gelscht werden kann oder nicht (das rm-Kommando fragt aber im Normalfall bei solchen Dateien vorher nach).

1.29 Hilfe

Hilfsseite Ramdisk-XFS
Dies ist eine Hypertextanleitung zum Ramdisk-XFS, der flexiblen dynamischen Ramdisk fr MagiC und MagiCMac, die auch lange Dateinamen untersttzt.

Vielen herzlichen Dank bei dieser Gelegenheit an Holger Weets & Co. fr den ST-Guide und die dazugehrenden Utilities!