

TipsUndTricks

David Göhler and Hartwig Tauber

COLLABORATORS

	<i>TITLE :</i> TipsUndTricks		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	David Göhler and Hartwig Tauber	August 22, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	TipsUndTricks	1
1.1	Tips & Tricks	1
1.2	Linke Maustaste	2
1.3	Sand im Getriebe	3
1.4	CD-ROM-Kit von VOB	3
1.5	Schneller Diskettensatz	4
1.6	Overdrive CD-ROM	4
1.7	Beckertext-III	5
1.8	Richtig konfiguriert	5
1.9	Syquest, die dritte	5
1.10	Mosaic-Abstürze	6
1.11	Zufällig kodiert	7
1.12	ARQ patchen	8
1.13	Speicher sparen	8
1.14	Letzter Boot?	9
1.15	Pools für M2Amiga	9
1.16	Schnellstart Fields Of Battle	11
1.17	Zeichen setzen	11
1.18	Assign rückgängig	12
1.19	UPD auf Zack	12
1.20	GNU gezähmt	13
1.21	Fehler in OS-Includes	14
1.22	Suchen lassen	15
1.23	Final Writer auf Trapp	15
1.24	Programm-Lokalitäten	15
1.25	System richtig genutzt	16
1.26	AFS paßt doch	17
1.27	XiPaint-CD läuft	18
1.28	RockRidge ist besser	19
1.29	Bildertrick in AmigaGuide	19

1.30	Fonts in Guide-Dateien	20
1.31	AFS-Probleme gelöst	20
1.32	Assign ADD überlistet	21
1.33	Haste Töne!	22
1.34	Wie voll ist RAM:?	22
1.35	C geschliffen	23
1.36	ARexx oder C?	23
1.37	Audio-CD am Monitor	24
1.38	FreshFont-Probleme?	25
1.39	Index	26

Chapter 1

TipsUndTricks

1.1 Tips & Tricks

----- Tips & Tricks 10/95 - 2/96 -----

Ausgabe 10/95

~Linke~Maustaste~::~::~::~

~Sand~im~Getriebe~::~::~

~CD-ROM~Kit~von~VOB~::~

~Schneller~Diskettensatz~::

~Overdrive~CD-ROM~::

~Beckertext-III~::

~Richtig~konfiguriert~::

~Syquest,~die~dritte~::
Ausgabe 11/95

~Mosaic~Abstürze~::

~Zufällig~kodiert~::

~ARQ~patchen~::

~Speicher~sparen~::

~Letzter~Boot?~::
Ausgabe 12/95

~Pools~für~M2Amiga~::

~Schnellstart~Fields~Of~Battle~

~Zeichen~setzen~::~::~::~::~

~Assign~rückgängig~::~::~::~

~UPD~auf~Zack~::~::~::~

~GNU~gezähmt~::~::~::~

~Fehler~in~OS~Includes~::~

~Suchen~lassen~::~::~

~Final~Writer~auf~Trapp~
Ausgabe Spezial 1/96

Programm-Lokalitäten

System() richtig genutzt
Ausgabe 1/96

AFS paßt doch

XiPaint-CD läuft

RockRidge ist besser

Bildertrick in AmigaGuide

Fonts in Guide-Dateien

AFS-Probleme gelöst

Assign ADD überlistet
Ausgabe 2/96

Haste Töne!

Wie voll ist RAM:?

C geschliffen

ARexx oder C?

Audio-CD am Monitor

FreshFont-Probleme?

1.2 Linke Maustaste

Linke Maustaste

Auch die linke Maustaste bewirkt etwas im Workbenchmenü. Sie ist äußerst hilfreich, wenn man mehrere Befehle nacheinander auswählt. Diese spezielle Funktion ist sehr leicht auszuführen; wie gewohnt den Menüpunkt anwählen und diesen zusätzlich mit der linken Maustaste einmal anklicken. Jeden dieser Klicks merkt sich der Amiga und nach Loslassen der rechten Maustaste werden die entsprechenden Aktionen nacheinander ausgeführt. Beispielsweise werden so im Menüpunkt »Fenster« die Befehle »Inhalt aufräumen« und »fixieren alles« in angeklickter Reihenfolge ausgeführt, und zwar ohne zweimal das Menü auszuwählen. Im Menü vieler Anwenderprogrammen wirkt diese Funktion ebenfalls.

tf

1.3 Sand im Getriebe

Sand im Getriebe

Wenn sich der Mauszeiger nur noch ruckelnd über den Bildschirm bewegt wird so mancher Amiga-Besitzer unruhig. Dabei besteht kein Grund zur Panik. In fast allen Fällen ist die Ursache bei der Maus zu finden. Staub, Fussel und andere Schmutzteilchen lagern sich schnell an den Rollen ab. Es hilft also nur eins -- einfach die Maus umdrehen und die Abdeckung abschrauben. Nimmt man jetzt die Kugel heraus, sieht man im Inneren drei Andruckrollen. Der Schmutz läßt sich am besten mit einem Wattestäbchen entfernen (das man in hartnäckigen Fällen mit etwas Alkohol benetzt). Ist der Dreck abgerubbelt wird die Kugel mit einem weichen nicht fusselnden Tuch gereinigt und wieder ins Gehäuse eingesetzt. Nun noch die Abdeckung drauf und der Mauszeiger müßte jetzt wieder jede Bewegung mitmachen.

tf

1.4 CD-ROM-Kit von VOB

CD-ROM-Kit von VOB

Die Installation des Mitsumi-CD-ROM-Laufwerk FX001D zusammen mit dem CD-ROM-Kit der Firma VOB ergibt auf dem Bildschirm die Fehleranzeige »Cannot open VOBSCSI.device unit 0«.Eine Nachfrage bei VOB ergab, daß sich diese Kombination nicht mit Festplatten des Herstellers »Quantum« verträgt. Das von VOB angebotene Programm »SPEEDUP« versagt mit Quantum-Festplatten ebenfalls den Dienst. Nach Einbau des »Tandem«-Controller funktionieren das CD-ROM-Laufwerk und die Festplatten fehlerfrei.

Dietmar Kauka/tf

1.5 Schneller Diskettensatz

Schneller Diskettensatz

Auf CD und in Mailboxen werden komplette Disketten oft mit dem »Device-Masher-System« gepackt. Diese »DMS-Dateien« sind leicht an ihrer Endung ».dms« zu erkennen. Zum Entpacken wird immer ein Datenträger (zum Beispiel eine Diskette) benötigt. Die Datei auf Diskette entpacken, um sie »nur mal zu testen«, nimmt viel Zeit in Anspruch. Hat die Diskette dann auch noch einen »Schreib/Lesefehler«, ist der Ärger perfekt. Schneller und zuverlässiger geht's mit der resetfesten RAM-Disk »RAD:«.

Um sie zu nutzen, gilt es sie erst einmal ins System einzubinden. Mit einem Doppelklick auf das Piktogramm »RAD0« in der Schublade »Storages/DosDrivers« der Workbench ist sie kurz darauf aktiviert. Nach öffnen einer Shell und Eintippen der Zeile

```
DMS write Name.dms to RAD0:
```

wird die DMS-Datei recht flott in die resetfeste RAM-Disk entpackt. Für »Name.dms« ist unbedingt der Name der Datei einzusetzen, die entpackt werden soll. Damit wird die resetfeste RAM-Disk genauso genutzt wie eine Diskette.

Das funktioniert allerdings nur, wenn sich das Programm »DMS« im Verzeichnis »C« der Workbench befindet. Die eingetippte Befehlszeile läßt sich nur dann fehlerfrei ausführen, wenn man sich in der Shell im gleichen Directory, wie die Datei »Name.dms« befindet.

Ohne viel Schreibaufwand und nur per Mausklick funktioniert's mit einem DOS-Tool wie »DosControl«.

tf

1.6 Overdrive CD-ROM

Overdrive CD-ROM

Oft ärgert man sich über das kurze Verbindungskabel zwischen dem Overdrive-CD-ROM-Laufwerk und dem PCMCIA-Port des Amiga. Das Kabel läßt sich mit wenig Aufwand sehr einfach verlängern. Es wird nur ein 25poliges Sub-D-Verlängerungskabel mit Stecker und Buchse benötigt. Der Fachhandel bietet diese Kabel für ungefähr 30 Mark an. Wichtig ist, daß das Kabel abgeschirmt und die Steckergehäuse aus Metall sind. Das Verlängerungskabel wird dann einfach zwischen Laufwerksbuchse und dem Overdrive Laufwerkskabel gesteckt.

Erich Schirle/tf

1.7 Beckertext-III

Beckertext-III

Manchmal widersetzen sich Tabulatoren, Kopf- und Fußzeilensymbole hartnäckig jedem Versuch, sie zu entfernen. Meist ist ein totaler Absturz des Rechners die Folge. Um die fertigen Dokumente doch noch zu retten, kann man die Zeilen vor und nach dem Übeltäter ausschneiden und auf eine neue Seite kopieren, sichern und am besten den Rechner neu starten. Die Kopf- und Fußzeilen-Symbole kann man im übrigen auch wie normalen Text behandeln, also ausschneiden, kopieren und in anderen Dokumenten wieder einsetzen.

Erich Schirle/tf

1.8 Richtig konfiguriert

Richtig konfiguriert

Bei der Verwendung des Directory-Tool »Multitool« ist es ärgerlich, daß man jedesmal nur ein Programm aufrufen kann. »Multitool« läßt sich erst dann wieder bedienen, wenn das aufgerufene Programm beendet wurde. Der Grund dafür ist in der Konfiguration zu suchen und läßt sich leicht ändern. Nach Aufruf des Voreinstellungseeditors wird vor jedem Programmaufruf der Befehl »Run« eingetragen. Damit wird »Multitool« nicht mehr blockiert. Der Grund ist, daß der Befehl »Run« einen eigenen Prozeß startet. Dieser wird seperat ausgeführt und gibt »Multitool« für weitere Aufgaben frei. Jetzt lassen sich beliebig viele Programme aus »Multitool« heraus starten. Das funktioniert genauso mit anderen Directory-Tools.

Alex Scheid/tf

1.9 Syquest, die dritte

Syquest, die dritte

In Ausgabe 3/95 waren Mountlist-Einträge für Syquest-Wechselplatten zu finden, die zwar funktionieren können, dies allerdings nur, wenn das Medium exakt genauso auf einem PC formatiert wurde. »CrossDos« kennt jedoch eine einfache und zugleich flexiblere Einbindung, als die

abgedruckte.

Solange »LowCyl=0« und »DosType=0x4d534800« gesetzt ist, liest CrossDos automatisch beim Mounten des Laufwerks die MD-DOS-Partitionsblöcke ein und konfiguriert sich eigenständig auf die richtigen Werte. Welche Partition, werden Sie sich vielleicht fragen (es können ja mehrere drauf sein)? Das ist genau der Clou: CrossDos entscheidet dies anhand des Laufwerksnamens, ausschlaggebend ist der letzte Buchstabe vor dem Doppelpunkt. »C« steht für die erste Partition, »D« für die zweite und so weiter. Ich habe so schon problemlos eine 1,2 GByte große MS-DOS-Festplatte mit 5 Partitionen sowie eine 270-MByte-Syquest eingebunden. Dabei habe ich das CrossDos der Workbench 2.1 verwendet.

Ein funktionierender DOSDrivers-Eintrag sieht folgendermaßen aus (die Angaben mit einem »*« sind beliebig, müssen aber angegeben werden, da »Mount« sie verlangt):

```
Device = xxx.device
FileSystem = L:CrossDOSFileSystem
Unit = x (0 - 7 einsetzen)
Flags = 0
BufMemType = 0
Mask = 0xFFFFFFFFE
MaxTransfer = 0x00FFFFFF
Mount = 1
DosType = 0x4d534800
LowCyl = 0
HighCyl = 255*
StackSize = 4000
GlobVec = -1
Buffers = 30
Surfaces = 64*
BlocksPerTrack = 32*
Reserved = 0*
Interleave = 0*
```

Volker Remuß/dg

1.10 Mosaic-Abstürze

AMosaic-Abstürze

Beim neuen »AMosaic 2.0« kommt es öfter vor, daß das Programm nach dem Anklicken einer Seite mit Eingabefeldern oder Download von Daten einfach abstürzt. Dies hat zwei Gründe.

Der Absturz nach dem Download war ein Programmfehler, der in der Beta-Release 3 nicht mehr auftreten sollte. Sie kann per FTP (File Transfer Protocol) über das Internet auf dem Rechner »max.physics.sunysb.edu« aus dem Verzeichnis »pub/amiga/amosaic« gezogen werden.

Bei Seiten mit Eingabefeldern benötigt AMosaic mindestens 32 KByte Stack. Dieser ist aber im mitgelieferten Icon auf 8 KByte gesetzt (was grob fahrlässig ist). Sollten Sie also Probleme mit AMosaic haben, erhöhen Sie unbedingt die Stackgröße. Klicken Sie dazu einfach auf das Icon des Programms und drücken Sie <Amiga_links i>. Danach erscheint das bekannte Informationsfenster der Workbench. Links oben ist ein Eingabefeld für die Stackgröße, in der »8192« steht. Tragen Sie dort »35000« ein und schließen Sie das Fenster mit einem Klick auf »Speichern«.

dg

1.11 Zufällig kodiert

Zufällig kodiert

Das abgedruckte Assembler-Programm kodiert jeden beliebigen Text nach dem Zufallsprinzip »Additive Congruential Method« und ist in »Algorithms (Second Edition)« von Robert Sedgewich ausführlich beschrieben. Ohne den Quelltext des Programms läßt sich der Text nicht wieder dekodieren. Das läßt sich erreichen, indem man alle »add«-Befehle durch »sub« ersetzt. Der Startzufallswert muß ungleich Null sein. Außerdem sollten Sie darauf achten, daß als Textlänge »Zeichenkettenlänge -- 1« anzugeben ist.

Ivan J. Eis/dg

Per Zufall: Mit diesem kurzen Assemblerstück kann man Texte per Zufallsalgorithmus en- und dekodieren:

```
*****
* $VER: RandomEncoding V1.0, von Ivan J. Eis © 1995 *
*****
* Parameter:      A0=Textzeiger      (APTR)          *
*                  D0=[Textlänge]-1 (UWORD)          *
*                  D1=Zufallswert    (UBYTE) Ungleich 0 *
*****

        lea      Text(pc),a0          * Adresse nach A0
        moveq   #[Ende-Text]-1,d0    * Textlänge in d0
        moveq   #7,d1                 * Zufallswert > d1
                                      * muss ungleich 0 sein.

Loop    move.b  d1,d2                 * Zufallswert in d2
        move.b  d1,d3                 * Zufallswert in d3

        and.b   #1,d2                 * Bit 0 raussuchen
        and.b   #2,d3                 * Bit 1 raussuchen

        lsr.b   #1,d3                 * Auf gleiche Höhe brin.
        eor.b   d3,d2                 * unterschiedlich?
```

```

    beq      SetNoBit    * Nein, dann nicht setzen

    or.b     #$80,d1    * Ja, Bit 7 setzen
    add.b    d1,(a0)+   * sub.b für Decoder
    lsr.b    #1,d1     * Zufallswert neu
                          * berechnen

    bra.b    NextChar

SetNoBit add.b    d1,(a0)+   * sub.b für Decoder
        lsr.b    #1,d1     * Zufallswert neu ber.

NextChar dbf      d0,Loop   * Länge Zähler
        rts                          * Programende

Text          dc.b    "Bitte decodiere diesen Text"
Ende

```

1.12 ARQ patchen

ARQ patchen

ARQ ist ein oft benutztes Utility, das die System-Requester wie »Benötige Datenträger xyz in einem Laufwerk« durch animierte und vor allem einfach per Tastatur zu bedienende Requester ersetzt. Leider kommt das Programm nur schlecht mit einer deutschen Workbench zurecht. So stimmen beispielsweise die Requester für den Drucker nicht mehr, da das Programm anhand der (englischen) Texte versucht herauszufinden, welche Animation es neben dem Text abspielen soll. Bei einer deutschen Workbench steht in dem Fenster aber Drucker und nicht Printer.

Lösen kann man das Problem mit einem Texteditor, der binäre Texte lädt und unverändert speichert (wie »CED« und »Edward« sowie »GoldED« nach Umkonfiguration es können!). Schalten Sie dazu auf »Überschreiben« und ersetzen Sie das Wort »Printer« durch »Drucker«. Hierbei muß man darauf achten, daß das neue Wort exakt die Länge des alten hat. Die Datei darf nicht länger oder kürzer werden! Sie können natürlich auch noch weitere Texte ersetzen. Vergessen Sie aber nicht, sich vorher eine Sicherheitskopie von ARQ zu machen.

Volker Georg Franke/dg

1.13 Speicher sparen

Speicher sparen

Spätestens ab Kickstart 3.0 werden zum Amiga diverse Monitortreiber mitgeliefert. Diese sind aber oft überflüssig, da sie vom vorhandenen Monitor gar nicht

dargestellt werden können oder aber auch durch die Treiber einer Grafikkarte ersetzt werden. In so einem Fall kann das Entfernen jedes einzelnen Treibers zwischen 10 und 20 KByte freien Speicher bringen; außerdem wird der Bootvorgang etwas beschleunigt. Letztendlich sollte man nur die NTSC- und PAL-Treiber und auf AA-Maschinen deren Dbl-Varianten unbedingt in der Schublade »SYS:Devs/Monitors« lassen, alle anderen kann man einmal auf ihren Gebrauchswert testen und dann gegebenenfalls löschen bzw. in »SYS:Storage/Monitors« ablegen. Alle Treiber befinden sich selbstverständlich auch noch auf den Systemdisketten und können von dort aus problemlos wieder ins System integriert werden.

Stefan Tiemann/dg

1.14 Letzter Boot?

Letzter Boot?

Falls man wissen möchte, wie lang das eigene System nun schon ohne Reset läuft, kann man sich eine Eigenschaft des AmigaDOS zunutze machen: Dieses speichert auf jedem Datenträger die Uhrzeit des Anlegens, also typischerweise die Zeit der Formatierung. Und genau dies geschieht auch bei der RAM-Disk, die das System während jedes Bootvorgangs neu anlegt. Sie besitzt deshalb auch die Uhrzeit des letzten Neustarts. Diese Zeit läßt sich über das Workbenchmenü »Piktogramm/Informationen« abfragen.

Stefan Tiemann/dg

1.15 Pools für M2Amiga

Pools für M2Amiga

Ab Kickstart 3.0 bietet das Betriebssystem Programmierern neue Routinen zur Speicherallozierung an, die bis zu zehn Mal schneller sein können als die alten und außerdem die Fragmentierung des Hauptspeichers deutlich verringern. Modula-2-Programmierer haben auf den ersten Blick das Nachsehen, in der entsprechenden Interface-Datei zur »exec.library« wurden die Poolfunktion offensichtlich vergessen. Da eine Korrektur und Neukompilierung dieser Dateien (beinahe) aussichtslos ist, bietet sich eine zweite Möglichkeit an, die übrigens auch für alle sonstigen Funktionen von Betriebssystembibliotheken benutzt werden kann. Man importiert als erstes einmal die »exec.library«:

```
IMPORT EL: ExecL;
```

Die eigentlichen Poolfunktionen kann man dann im Programm oder eigenen Modulen als ganz normale Prozeduren

deklarieren und den Autodocs entsprechend benutzen (s. Listing).

Stefan Tiemann/dg

Mit Pool: Diese Zeilen reichen aus, um Modula-2-Anwender die OS-3.0-Poolfunktionen zu erschließen

```
(*EntryExitCode:= FALSE *)
PROCEDURE AllocPooled(poolHeader{R.A0}: ADDRESS;
                     memSize{R.D0}: LONGCARD): ADDRESS;
BEGIN
  ASSEMBLE (MOVE.L A6, -(A7)
            MOVE.L EL(A4), A6
            JSR    -708(A6)
            MOVE.L (A7)+, A6
            RTS
            END);
END AllocPooled;

(*EntryExitCode:= FALSE *)
PROCEDURE FreePooled(poolHeader{R.A0}: ADDRESS;
                    memory{R.A1}: ADDRESS; memSize{R.D0}: LONGCARD);
BEGIN
  ASSEMBLE (MOVE.L A6, -(A7)
            MOVE.L EL(A4), A6
            JSR    -714(A6)
            MOVE.L (A7)+, A6
            RTS
            END);
END FreePooled;

(*EntryExitCode:= FALSE *)
PROCEDURE DeletePool(poolHeader{R.A0}: ADDRESS);
BEGIN
  ASSEMBLE (MOVE.L A6, -(A7)
            MOVE.L EL(A4), A6
            JSR    -702(A6)
            MOVE.L (A7)+, A6
            RTS
            END);
END DeletePool;

(*EntryExitCode:= FALSE *)
PROCEDURE CreatePool(memFlags{R.D0}: ED.MemReqSet;
                    puddleSize{R.D1}, treshSize{R.D2}: LONGCARD): ADDRESS;
BEGIN
  ASSEMBLE (MOVE.L A6, -(A7)
            MOVE.L EL(A4), A6
            JSR    -696(A6)
            MOVE.L (A7)+, A6
            RTS
            END);
END CreatePool;
```

1.16 Schnellstart Fields Of Battle

Schnellstart Fields Of Battle

Das neue Strategiespiel »Fields Of Battle (FOB)« nimmt sich gerade auf langsamen Amigas recht behäbig aus. Schon der Start des Programms kann mehrere Minuten dauern. Diesen Startvorgang kann man aber deutlich beschleunigen. FOB lädt zu jedem Neustart den CGTimes-Zeichensatz in den Größen 12, 16, 24 und 32. Diese liegen aber nicht direkt als Bitmapfonts vor, sondern müssen von der »diskfont.library« aufwendig aus dem Vektorfont »CGTimes« berechnet werden.

Abhilfe schafft hier das mit dem Amiga-OS gelieferte Programm »Intellifont« (»Fountain« unter Amiga-OS 2.0), das normalerweise in der Schublade »Sys:System« sein Dasein fristet oder sich noch auf den Systemdisketten befindet. Dieses startet man, klickt dann »Modify Existing Typefaces...« an und wählt nun aus der linken Tabelle den CGTimes-Zeichensatz aus. Bei Size trägt man nacheinander die Größen 12, 16, 24 sowie 32 ein und klickt jeweils »Add Size« an. Danach muß für jede neue Größe noch einmal »Create Bitmap« angewählt werden. Abgeschlossen wird der Vorgang mit »Perform Changes«. Abhängig vom Amiga-Modell kann man so bei jedem Start von FOB mehr als zwei Minuten Wartezeit sparen.

Stefan Tiemann/dg

1.17 Zeichen setzen

Zeichen setzen

Der Amiga-Zeichensatz beinhaltet einige sehr nützliche Zeichen, die nur wenige kennen und die über einfache Tastenkombinationen einfach zu erreichen sind.

Der Unterstrich <Shift Alt n> ist ein oben liegender, bei vielen Druckern durchgezogener Strich, der sich hervorragend zum Unterstreichen mit anschließendem kleinen Abstand eignet.

<Alt 8> erzeugt den kleinen viereckigen Punkt »·«, den man besonders gut bei Aufzählungen brauchen kann.

<Alt e> gibt das Copyright-Zeichen »©«, mit <Alt r> erhält man das Register-Zeichen »®«. Doch Vorsicht; diese Zeichen finden sich nicht in allen Zeichensätzen.

Mit den Kombinationen <Alt 0> und <Alt 9> erreicht man übrigens die französischen Anführungszeichen » und «, die neben den normalen laut Duden im Deutschen zulässig sind. Da ein Computer normalerweise aber keine »Anführungszeichen unten« kennt, sind sie die einzigen

Zeichen, die den Anfang und das Ende der wörtlichen Rede eindeutig anzeigen.

Wer schon mal die kleinen, hochgestellten Zahlen $\$^1\$$, $\$^2\$$ und $\$^3\$$ gesucht hat: sie erreicht man per `<Alt 1>`, `<Shift Alt 2>` und `<Alt 3>`.

dg

1.18 Assign rückgängig

Assign rückgängig

Hinter Assigns wie »C:« verbirgt sich im allgemeinen nur ein Verzeichnis, hinter solchen wie »LIBS:« ab Amiga-OS 3.0 aber mindestens zwei. Mit einer Zeile wie

```
Assign C: Ram:C Add
```

kann man weitere Verzeichnisse zu einem bestehenden Assign hinzufügen. Es ist aber auch möglich, diese genauso gezielt wieder zu entfernen. Statt des »Add« ist dann einfach nur »Remove« anzugeben.

```
Assign C: Ram:C Remove
```

Dies ist vor allem in Skripts von Vorteil, wenn man nur zeitweilig ein Assign benötigt und nach Gebrauch wieder aufheben möchte.

dg

1.19 UPD auf Zack

UPD auf Zack

Dem »ToolManager« von Stefan Becker liegt das Programm »UPD« von Jonas Petersson bei. Dabei handelt es sich um einen Soundplayer, der im Hintergrund die ARexx-Schnittstelle »Play« öffnet und auf ARexx-Kommandos wartet. Für Systemmeldungen arbeitet dieser mit einer ID-Datei, in der man für bestimmte Meldungen Klänge angeben und somit das Verhalten steuern kann.

Interessant ist dies vor allem zusammen mit ARQ, dem Systemrequester-Ersatz. Denn nach dem Öffnen eines Fensters sendet ARQ einen Befehl an den ARexx-Port »Play« ab. UPD seinerseits interpretiert den Befehl und spielt mit Hilfe der ID-Datei den passenden Klang ab. Die Beispiel-Datei »s:upd.ids« finden Sie im nebenstehenden Listing.

Hinter jedem Befehl steht der Pfad und evtl. eine Option

wie »-S« für Stereo. Damit UPD die ID-Datei auch nutzt, ist sie beim Start mit der Option »-f« anzugeben:

```
Run >NIL: upd -fs:upd.ids
```

Dies sollte aber erst geschehen, nachdem man »RexxMast« gestartet hat. Wenn UPD einmal läuft, läßt er sich auch als Sound-Player mißbrauchen:

```
address PLAY file <sound>
```

Volker Georg Franke/dg

Hier spielt die Musik: Die ID-Datei von UPD legt fest, wann welche Musik zu spielen ist

```
warn_write_protected  sound:closed -S
warn_insert_volume    sound:who_is -S
warn_delete           sound:really -S
warn_printer          sound:printit -S
error_task_held       sound:shit -S
error_read_write      sound:anotherone -S
error_general         sound:something -S
warn_general          sound:oops -S
query_general         sound:question -S
```

1.20 GNU gezähmt

GNU gezähmt

Auf Seite 47 in Ausgabe 8/95 hat ein Leser ein Problem mit dem GNU-C++-Compiler geschildert. Wie dort richtig erwähnt, versieht der C++-Compiler alle Funktionsnamen mit einer von den Parametern abhängigen Buchstabenkombination, die sich Signatur nennt. Dieses Anhängen der Signatur, das dann den Linker in die Irre führt, läßt sich recht einfach verhindern, indem man dem GNU-Compiler per Direktive »extern "C"« mitteilt, daß er sie für spezielle Funktionen nicht verwenden soll:

```
extern "C" struct Library *OpenLibrary(
    struct Library *, LONG);
```

Es ist auch möglich, hinter dem »"C"« eine geschweifte, geöffnete Klammer anzugeben, alle betroffenen Funktionen aufzuzählen und anschließend eine geschlossene Klammer anzugeben. Dieses Verfahren ist vor allem in Prototyp-Dateien im Verzeichnis »clib« der Amiga-OS-Includes anzuwenden.

Wer im Wechsel mit GNU-C und GNU-C++ arbeitet, sollte die zusätzlichen Anweisungen mit einer Preprozessor-Direktive umschließen, damit sie nur dann ausgeführt werden, wenn der Compiler auch im C++-Modus ist (denn in ANSI-C gibt es das Konstrukt »extern« nicht in dieser Form). Im nebenstehenden

Listing sieht man, wie dies z.B. in der Datei
»clib/timer_protos.h« zu bewerkstelligen ist.

Frank Meyer/dg

Der GNU-C++-Trick: Damit kann der Linker auch im C++-Modus
mit einfachen C-Funktionen noch etwas anfangen

```
#ifndef CLIB_TIMER_PROTOS_H
#define CLIB_TIMER_PROTOS_H
#ifndef DEVICES_TIMER_H
#include <devices/timer.h>
#endif

#ifdef __cplusplus
extern "C" {
#endif

void AddTime(struct timeval *,struct timeval *);
void SubTime(struct timeval *,struct timeval *);
LONG CmpTime(struct timeval *,struct timeval *);
ULONG ReadEClock(struct EClockVal *dest );
void GetSysTime(struct timeval *dest );

#ifdef __cplusplus
}
#endif

#endif /* CLIB_TIMER_PROTOS_H */
```

1.21 Fehler in OS-Includes

Fehler in OS-Includes

Der Versuch, zwei simple BOOPSI-Gadgets mit dem Tag
»ICA_Targets« zur Zusammenarbeit zu bewegen, hat fast zwei
Wochen Arbeit und viele Nerven gekostet -- und dann war der
Fehler klar: Er liegt in den Amiga-OS-3.0-Includes von
Commodore! Ich benutze das Assembler-Paket »DevPac 3.04«
und die mitgelieferten Includes der Version 39.108. Um den
Fehler zu beheben, ist in der Datei »intuition/icclass.i«
die Zeile 26:

```
ICA_Dummy EQU $40000
```

durch

```
ICA_Dummy EQU (TAG_USER+$40000)
```

zu ersetzen. Danach funktionieren BOOPSI-Gadgets, wie sie es
sollen. In den Amiga-OS-3.1-Includes ist der Fehler bereit
entfernt.

Peter Schmidt/dg

1.22 Suchen lassen

Suchen lassen

Wem ist das noch nicht passiert: Man weiß, daß eine bestimmte Datei noch auf der Festplatte liegt, die man jetzt benötigt. Aber leider ist einem entfallen, wo sie liegt. Statt selbst zu suchen (oder nach PD-Programmen Ausschau zu halten), sollte man suchen lassen: mit »Search« des Amiga-OS. Dies kennt die Optionen »FILE«, um die Dateisuche zu aktivieren und »ALL«, um auch alle Unterverzeichnisse durchzukämmen:

```
Search <Verzeichnis> <Datei> ALL FILE
```

Gibt man statt »Verzeichnis« nur zwei Anführungszeichen oben an, sucht Search im aktuellen Verzeichnis. Sie können bei dem Dateinamen auch Platzhalter einsetzen, müssen dann aber noch die zusätzliche Option »PATTERN« angeben.

Patrick Bernau/dg

1.23 Final Writer auf Trapp

Final Writer auf Trapp

Möchten Sie mit »Final Copy II« oder »Final Writer« Grafiken in den Text einfügen? Dann sollten Sie dies erst tun, wenn Sie den Text komplett eingegeben haben; aus mehreren Gründen:

Wenn Sie die Option »Grafik sichern« der IFF-ILBM-Einstellungen angeklickt haben, sichert das Programm alle eingebundenen Grafiken mit dem Text zusammen in eine Datei, was wesentlich länger dauert und bei Diskettenbetrieb schier unerträglich ist.

Enthält das Dokument Grafiken, um die Text herumfließt, und fügen Sie Text am Anfang ein, dann benötigt Final Copy II sehr lange, um Tastatureingaben zu verarbeiten.

Befinden sich im Dokument Texte, die sich direkt auf nebenstehende Grafiken beziehen, können sich durch Korrekturen Text und Grafik voneinander »entfernen«. Sie müssen anschließend alle Textbezüge noch einmal korrigieren oder alle Grafiken neu positionieren.

Manfred Jörg/dg

1.24 Programm-Lokalitäten

Programm-Lokalitäten

Wie fast jedes Dateisystem kann man auch beim AmigaDOS Dateien »absolut« und »relativ« angeben. Absolut bedeutet, daß eine Dateibezeichnung immer aus einem Laufwerksnamen, einem Doppelpunkt, einigen Verzeichnisnamen und schließlich dem Dateinamen besteht. Diese Angabe nennt sich »voller Pfad«. Statt des Laufwerksnamen läßt sich auch der Name eines Assigns angeben.

Läßt man die Bezeichnung der Partition weg, ist die Dateiangebe relativ zum aktuellen Verzeichnis. Fehlen jegliche Partitions- und Verzeichnisnamen, ist die Datei im aktuellen Verzeichnis gemeint. Dies ist nützlich, wenn man mehrere Dateien zusammenstellt, aber nicht bekannt ist, wie die Platten und Partitionen desjenigen heißen, der die Dateien bekommt.

Möchte man etwa in MainActor Broadcast eine Animation mit einem Sound versehen, sollten man in den Dateirequestern nur relative Dateinamen eintragen -- am besten kopiert man die Animation und die Sounddatei in ein Verzeichnis, beendet MainActor, wechselt in das Verzeichnis und startet das Programm erneut aus diesem Verzeichnis heraus. Dann stelle man die Animation mit Sound zusammen (lösche dabei Pfadangaben, falls diese im Dateirequester vom Programm eingetragen wurden) und speichere das Projekt.

Das Projekt kann nun problemlos an andere weitergegeben werden. Sie können dann per Doppelklick Animation und Sound starten, ohne mit »Please insert xyz: in any drive« belästigt zu werden. Diese Vorgehensweise ist übrigens für viele Programme sinnvoll und vermeidet unnötigen Frust beim Anwender.

dg

1.25 System richtig genutzt

System() richtig genutzt

Oft ist es praktisch, nicht alles selbst zu programmieren, sondern auf Bewährtes zurückzugreifen. So macht es Sinn, aus einem eigenen Programm z.B. Shell-Kommandos aufzurufen und somit zu nutzen. Dies erlaubt seit Amiga-OS 2.0 die Funktion »System()« einfach und komfortabel.

Wird dabei das eigene Programm von der Workbench aus gestartet, besitzt es allerdings von vornherein kein Console-Fenster. Nun lassen sich zwar für die Ein- und Ausgabe »Dateihandles« beim Aufruf von System() angeben, die vorher durch Öffnen von Dateien zu initialisieren sind, diese müssen aber leider verschieden sein. Man kann also nicht ein Console-Fenster öffnen und das eine Dateihandle für Ein- und Ausgabe angeben -- sehr lästig, wenn man nicht nur Texte ausgeben möchte, sondern auch Eingaben erwartet.

Die Lösung ist einfacher als befürchtet: In dem String mit dem Befehlsnamen, der System() zu übergeben ist, kann man Ein- und Ausgabe wie gewohnt mit den Zeichen »<« und »>« umleiten. Diese beiden Zeichen lassen sich zum Glück kombinieren, wie in:

```
<>CON:60/60/400/100/Aha!
```

Dann öffnet sich nur ein Fenster für Ein- und Ausgaben. Wie bei jedem Console-Fenster läßt sich durch weitere Angaben wie »/WAIT/CLOSE/AUTO« hinter dem Titel (im Beispiel »Aha!«) steuern, ob das Fenster nach Kommandoende offen stehen bleibt, ein Schließschalter erhält und sich erst bei der ersten Ausgabe öffnen soll.

Auch wenn in der Shell keinerlei Ein-/Ausgaben gewünscht sind, kann man sich mit wie beschrieben helfen und einfach ein »<>NIL:« einsetzen.

dg

1.26 AFS paßt doch

AFS paßt doch

Beim Installieren des AmiFileSave-Dateisystems kann es passieren, daß der Amiga meint, er habe nicht genug Platz im RDB (Rigid Disk Block), um das Dateisystem noch aufzunehmen. Hat man dort viele Dateisysteme gespeichert, reicht es meist, ein ungenutztes (und nur ein solches!) zu löschen, um für genug Platz zu sorgen. Bitte achten Sie aber dabei unbedingt darauf, kein Dateisystem zu löschen, das noch von einer bestehenden Partition benutzt wird. Dies kann fatale Folgen beim nächsten Hochfahren haben.

Manchmal reicht aber auch das Löschen nicht, da kleinere Platten nur wenige Sektoren pro Spur besitzen. In diesem Fall gibt es nur eins: Die erste Partition auf der Festplatte muß verschoben oder

verkleinert werden, damit am Anfang der Festplatte mehr Platz entsteht. Leider führt dies automatisch zum totalen Datenverlust dieser Partition -- wenn man aber sowieso gerade auf AFS umschwenkt, bleibt einem das sowieso nicht erspart. Damit wird also ein Backup fällig.

Zur Platzgewinnung ist die erste Partition und anschließend der Schalter »Advanced Options« anzuklicken. Daraufhin erscheinen neue Schalter. Für die Anpassung wichtig ist der Wert hinter »Start Cyl:«. Meist steht dort »2«. Versuchen Sie es einfach mit einem größeren Wert (vielleicht reicht ja schon 3 oder 4). Wichtig: Je größer der Wert, umso kleiner wird der verbleibende Platz für die Partitionen auf der restlichen Festplatte.

dg

1.27 XiPaint-CD läuft

XiPaint-CD läuft

Seit kurzem wird das 24-Bit-Malprogramm XiPaint 3.2 auf CD vertrieben (s. Kurzttest in dieser Ausgabe Seite xx). Die Unterstützung der verschiedenen Grafikkarten ist in Form von Libraries geschehen, die bei Bedarf geladen werden. Die Namen der Libraries auf CD können -- wie beim Amiga üblich -- groß oder klein geschrieben sein. Allerdings verträgt es das Amiga-OS nicht, wenn der Name der Library im Speicher statt klein- großgeschrieben wird.

Unglücklicherweise sind die Namen auf der CD aber großgeschrieben, was dazu führt, daß XiPaint sie bei Verwendung der meisten CD-ROM-Dateisysteme falsch installiert und anschließend nicht laden kann (vielmehr lehnt das Amiga-OS sie ab). Wenn Sie also ein solches Problem haben, sollten Sie nach der Installation von XiPaint 3.2 die großgeschriebenen Libraries in »Libs« alle umbenennen und dabei komplett kleinschreiben. Anschließend wird XiPaint ohne Probleme starten und funktionieren.

Diese Schwierigkeit kann übrigens auch durchaus bei anderen Programmen auftreten. In dem Fall lohnt es sich, in »Libs:« oder den entsprechenden Verzeichnissen nachzuschauen, ob die Libraries nicht vielleicht großgeschrieben sind.

dg

1.28 RockRidge ist besser

RockRidge ist besser

CD-ROMs sind Datenträger, die fast jedes Computer-System lesen kann. Wie leider so oft, hat man sich deswegen auf einen recht kleinen, gemeinsamen Nenner geeinigt, der sich ISO 9660 nennt. Seit einiger Zeit gibt es als abwärtskompatible Erweiterung das RockRidge-Format, das wesentlich mehr Möglichkeiten bietet (mehr als der Amiga verdauen kann).

Beim Freeware-CD-ROM-Dateisystem »AmiCDROM« kann man zwischen beiden Systemen wählen, indem man in den Startup-String ein »R« einfügt (um RockRidge anzuschalten). Fehlt diese Option, erscheinen auf einigen CDs alle Dateinamen nur großgeschrieben. Mit der Option dagegen gibt es sowohl Groß- als auch Kleinbuchstaben in Dateinamen. Dies kann unter Umständen einige Probleme beseitigen (s. auch XiPaint-Tip).

dg

1.29 Bildertrick in AmigaGuide

Bildertrick in AmigaGuide

AmigaGuide-Dateien haben einen Nachteil: in ihnen kann man Bilder und Text nicht mischen. Außerdem ist es (regulär) nicht möglich, per Klick von einer Seite mit Text auf eine mit einem Bild zu wechseln. Ein Trick erlaubt dies aber doch.

Jede Verzweigung in einem AmigaGuide-Dokument kann auf einen Knoten (engl. »Node«) im gleichen Dokument oder einen in einer externen Datei zeigen. Sie unterscheiden sich nur dadurch, daß ein externer Verweis einen Dateinamen enthält, der durch einen Schrägstrich »/« und den Knotennamen ergänzt wird, etwa

"Hallo" link "Text.guide/main"

Nutzt man aber statt AmigaGuide »Multiview« (oder AmigaGuide ab Amiga-OS 3.0 mit den Datatypes), läßt sich statt »Text.guide« auch der Name einer Bilddatei angeben. Dabei darf das »/main« aber nicht fehlen:

```
"Hallo" link "Bild.iff/main"
```

Dann versucht das Programm die Datei zu laden, merkt, daß es ein Bild ist und zeigt per Datatype das Bild an. Daß etwas nicht stimmt, sieht man schon daran, daß noch die Schalter für Textelemente über dem Bild zu sehen sind, obwohl ein Bild dargestellt wird.

dg

1.30 Fonts in Guide-Dateien

Fonts in Guide-Dateien

Ab Amiga-OS 3.0 gibt es für Hypertexte im AmigaGuide-Format einige neue Kommandos, die dem Aussehen mehr Pep verleihen. Dazu gehören die Anweisungen

font

wordwrap und

proportional

Hinter font kann man z.B. »helvetica.font 13« angeben, um diesen für das Dokument oder auch nur die folgende Seite einzusetzen. Die Anweisung kann zwischen Knoten beliebig oft stehen und wirkt sich auf die noch folgenden Seiten aus. Sie zeigt aber keine Wirkung, wenn man versucht, den Zeichensatz mitten in einer Seite umzuschalten. Hinter wordwrap kann »on« oder »off« stehen, um ein automatisches Anpassen an die Fensterbreite zu erlauben, was insbesondere dann Sinn macht, wenn man einen Zeichensatz mit variabler Zeichenbreite verwendet und folgerichtig hinter proportional ebenfalls »on« angegeben hat.

dg

1.31 AFS-Probleme gelöst

AFS-Probleme gelöst

Bei der Installation von AmiFileSafe V.2.2 Pro hatte ich zwei Probleme, die vielleicht auch anderen zu schaffen machen.

Bei der Installation streng nach Handbuch sollte man nach erfolgreicher RDB-Installation erneut

hochfahren (booten) und anschließend die betroffene Partition formatieren. Jedoch war nach dem Reboot kein Icon zu sehen. Nach langer Suche fand ich schließlich heraus, daß das SpeedUp-System von Breitfeld Computersysteme sich nicht gleichzeitig benutzen läßt. Nachdem der Befehl »Speedup« aus der Startup-Sequenz gelöscht war, erschien nach dem Booten das benötigte Icon und die Partition ließ sich problemlos formatieren.

Bei der Floppy-Installation funktionierte alles auf Anhieb -- scheinbar. In DirOpus 4.11 gab es jedoch das Problem, daß das Programm Diskettenwechsel von AFS-Disketten nicht mehr erkannt hat. Die Lösung: Laden Sie den Mountlist-Eintrag von AF0: (oder einer ähnlichen Bezeichnung, die das AFS für Disketten bei Ihnen einbindet) in einen Texteditor und setzen Sie den Wert von »Priority« von 10 auf 11. Damit erkennt auch DirOpus jeden Diskettenwechsel sofort.

Werner Lisseck/dg

1.32 Assign ADD überlistet

Assign ADD überlistet

Mit dem Befehl »Assign« hat das Amiga-OS anderen Systemen (auch und gerade Windows 95) vieles voraus. Besonders interessant sind dabei die Optionen »PATH« und »DEFER«, die es ab Amiga-OS 2.0 gibt. Sie sorgen dafür, daß die Verbindung, die ein Assign-Befehl zwischen logischer Bezeichnung und Ziel zieht, erst bei der ersten Benutzung überprüft und tatsächlich anlegt.

Steht also in der User-Startup

Assign Bla: Wechsel: PATH

muß das Laufwerk »Wechsel:« bei Befehlsausführung noch nicht bekannt sein. Erst wenn ein Programm das erste Mal auf »Bla:« zugreift, sucht das Amiga-OS nach »Wechsel:«. Dies ist nützlich bei Laufwerken, die nicht immer angeschlossen sind, bei Wechselplatten und CDs.

PATH und DEFER unterscheiden sich darin, daß ein PATH-Assign nach der Benutzung gleich wieder aufgelöst wird, die andere Variante jedoch bestehen bleibt und nur per Option »REMOVE« aufgelöst werden kann.

Allerdings kann man an ein Assign vom Typ PATH nie ein weiteres per »ADD« hinzufügen. Bei DEFER geht

es, man muß aber vorher dafür sorgen, daß es schon einmal benutzt wurde. Dies ist übrigens auch der Grund, warum bei der Sequenz, die MUI (das Magic User Interface) in die User-Startup einfügt, die zwei Zeilen

```
if exists HELP:dummy
endif
```

auftauchen. Sie dienen nur dazu, den Assign HELP: (vom Typ DEFER) aufzulösen, bevor man ein »Assign ... ADD« darauf anwenden kann.

dg

1.33 Haste Töne!

Haste Töne!

Andere Computer -- andere Sitten. Auf anderen Rechnern werden Soundsamples leider nicht im IFF-8SVX-Format gespeichert. Bei Windows ist es Wave, beim Macintosh AIFF. Möchte man diese wandeln, muß man wahrscheinlich nach einem Konverter suchen. Falsch gedacht!

Was viele bisher nicht bemerkt haben: Multiview kann auch speichern. Es tut dies grundsätzlich in IFF-Formaten -- und das gilt nicht nur für Bilder. Hat man das passende Datatype, kann Multiview die Daten laden. Beim Speichern entsteht dann automatisch eine IFF-Datei. Dies funktioniert auch bei Soundsamples tadellos. Auch wenn man eine 16-Bit-Wave-Datei mit einer Sample-Frequenz von 44 kHz lädt, entsteht eine 8-Bit-8SVX-Datei mit der passenden Frequenz. Alles, was man dazu braucht, sind die richtigen Datatypes. Eine sehr gut sortierte Auswahl findet sich auf der neuen »Meeting Pearls Vol. III«.

dg

1.34 Wie voll ist RAM:?

Wie voll ist RAM:?

Trivial oder nicht trivial: Die Frage war lediglich, wieviel Platz »ENV:« in der RAM-Disk einnimmt. Nichts einfacher als das, sollte man meinen: Auf der Workbench die RAM-Disk öffnen und im Fenstertitel die Größe ablesen, denn die

Verzeichnisse »T« und »Clipboards« sind nahezu leer. Dort stand 928 KByte. Laut Directory-Tool »RO« belegt »ENVARC:« auf der Festplatte und ENV: aber nur etwa 300 KByte. Laut DU von Matt Dillon (beim DICE dabei) sind es 468 KByte, DUD (aus dem Aminet) meint, es seien etwa 580 KByte, ein LS-Programm kommt auf etwa 400, das List von Amiga Technologies liegt bei 967 Blöcken á 1 KByte. »Ja, was nun?« ist da die Frage.

In solchen Fällen hilft nur eines: Mit »Avail Flush Total« den momentan freien Speicher bestimmen, ENVARC: nochmals in die RAM-Disk kopieren (etwa nach ENV2), wieder mit »Avail Flush Total« nachmessen und die Differenz bilden.

Und siehe da, keiner hatte Recht: Es waren etwa 385 KByte, die nach dem Kopieren fehlten -- schon erstaunlich.

dg

1.35 C geschliffen

C geschliffen

Wenn man in einer Hochsprache wie »C« programmiert, macht man sich nicht oft Gedanken darüber, was der Compiler aus den schönen Anweisungen macht, die die Sprache anbietet.

So gibt es bei Schleifen in C grundsätzlich zwei Varianten: die for- und die while-Schleife. Jede läßt sich durch die jeweils andere ersetzen. Aber ein Compiler macht daraus nicht immer die gleichen Anweisungen. So war es früher beim Aztec-C unsinnig, mit »for« zu programmieren, da dieses meist fast doppelt soviel Zeit benötigte wie ein while-Konstrukt.

Wer also bei bestimmten Algorithmen ein paar Prozente herausholen möchte (wie bei Sortieralgorithmen), sollte die in Frage kommenden Sprachelemente auf Geschwindigkeit hin austesten oder sich einfach mal ansehen, was der Compiler draus macht.

dg

1.36 ARexx oder C?

ARexx oder C?

ARexx-Programmierer loben -- nicht zu Unrecht -- oft die mächtigen String-Funktionen ihrer Sprache. Was viele aber nicht wissen: Auch C beinhaltet seit der Verabschiedung des ANSI-C-Standards einige Funktionen, mit denen sich Strings sehr einfach parsen und durchsuchen lassen.

Besonders hervorzuheben sind dabei: »strtok()«, »strchr()«, »strpbrk()« und »strspn()«. strtok() dient dazu, eine Zeichenkette in Teile zu zerlegen (Neudeutsch: zu parsen). Dabei benötigt es als Parameter einen Zeiger auf den zu zerlegenden String und eine Zeichenkette, die aus den Buchstaben besteht, die als Trennungszeichen dienen (bei ARexx wäre es das Leerzeichen). Bei jedem Aufruf erhält man das jeweils nächste Stück.

»strchr()« erlaubt die Suche einer Zeichenkette in einer anderen. »strpbrk()« funktioniert wie strtok(), nur daß nichts zuerlegt wird, sondern man als Ergebnis einen Zeiger auf das erste, gefundene Trennungszeichen oder Null zurückerhält.

»strspn()« erwartet ebenfalls zwei Zeichenketten: die zu durchsuchende und eine, die eine Zeichenmenge enthält. Die Funktion sucht weiter, solange im ersten String Zeichen verwendet werden, die in der Zeichenmenge vorkommen. Anschließend erhält man die Länge zurück, für die diese Bedingung im ersten String gilt. Gibt man etwa als Zeichenmenge die Ziffern »0« bis »9«, Komma und Punkt an, lassen sich Zahlen sehr leicht aus einem String herausfiltern. Zu strspn() gibt es noch das Gegenstück »strcspn()«, bei dem die Zeichenmenge alle Ausschlußzeichen enthält.

Das ist lange noch nicht alles, was C bezüglich Strings zu bieten hat. Ein Blick in ein gutes C-Buch fördert noch viel zu Tage. Mein persönlicher Favorit: ANSI-C, A Lexical Guide, Prentice Hall, ISBN 0-13-037814-3.

dg

1.37 Audio-CD am Monitor

Audio-CD am Monitor

Haben Sie ein CD-Laufwerk? Wollen Sie Ihre Musik-CDs über den Monitorlautsprecher hören? Dann kaufen Sie sich ein Kabel »2 Cynchstecker auf 3poligen Klinckenstecker 3.5 mm«, es kostet ca. sieben

Mark. Verbinden Sie den Kopfhörerausgang vom CD-Laufwerk mit dem Audio-Eingang des Monitors. Zumindest funktioniert es einwandfrei mit einem Mitsumi-FX-CD-ROM-Laufwerk und dem Commodore-Monitor 1942. Die Lautstärke am Laufwerk sollte vorsichtshalber nicht voll aufgedreht werden. Wer das Kabel selbst basteln möchte, sollte folgende Anschlußbelegung beachten: Die Spitze des Klinkensteckers mit links verbinden, die Mitte mit rechts, der hintere Kontakt ist Masse.

Herbert Pittermann/dg

1.38 FreshFont-Probleme?

FreshFont-Probleme?

Wer (nicht skalierbare) BitMap-Fonts direkt von einer FreshFont-CD nutzen möchte, wird enttäuscht sein, wenn sein Dateisystem die CD als ISO 9660 behandelt. Dann werden alle Dateinamen groß geschrieben. Dies hat zur Folge, daß das Betriebssystem die Zeichensätze nicht mehr als solche erkennt (ein ähnliches Problem gibt es auch bei Libraries, s. unbedingt Tips & Tricks 1/96). Falls das Dateisystem dazu in der Lage ist, sollte man die »RockRidge«-Erweiterung einschalten. Bei AmiCDROM geschieht dies durch Einfügen eines »R« in den Startup-String.

Es ist aber auch möglich, die Fonts zu kopieren und anschließend per DOS-Befehl »Rename« klein zu schreiben. Wer dies gerne automatisiert, findet im nebenstehenden ARexx-Listing »rename.rx« ein Programm, das alle Dateinamen automatisch in Kleinbuchstaben wandelt und die Datei auch gleich umbenennt. Wenn Sie das Skript nach »REXX:« kopieren und das s-Bit setzen (mit »Protect rename.rx +s«) können Sie die Umwandlungen für ein Verzeichnis wie folgt erledigen:

```
list #?.font lformat
```

```
"rename.rx %s" to ram:exhopp
```

```
Execute ram:exhopp
```

```
Delete ram:exhopp
```

Die ersten beiden Zeilen müssen Sie dabei zu einer zusammenfassen und alle drei in der Shell ausführen lassen.

Das ARexx-Listing:

```
/* Rename.rx macht aus Groß- Kleinbuchstaben und
```

```

    benennt alle übergebenen Dateinamen um */

options results
addlib("locale.library",0,-30,38)
parse arg filenames

do while filenames ~= ''

    parse var filenames name filenames
    lenge = length(name)
    neuname = ""

    do lauf = 1 to lenge
        neuname = neuname ||,
                ConvToLower(SubStr(name,lauf,1))
    end

    address command 'rename "'name'" "'neuname' "'
end

```

Herbert Pittermann/dg

1.39 Index

Index aller Verzweigungen

AFS paßt doch

AFS-Probleme gelöst

ARexx oder C?

~ARQ~patchen~::~::~::~::~::~::~::~::~::~

Assign ADD überlistet

~Assign~rückgängig~::~::~::~::~::~::~::~

Audio-CD am Monitor

~Beckertext-III~::~::~::~::~::~::~::~

Bildertrick in AmigaGuide

C geschliffen

~CD-ROM-Kit~von~VOB~::~::~::~::~::~

~Fehler~in~OS~Includes~::~::~::~

~Final~Writer~auf~Trapp~::~::~

Fonts in Guide-Dateien

FreshFont-Probleme?
~GNU~gezähmt~::~::~::~::~::~::~::~::~::~
Haste Töne!
~Letzter~Boot?~::~::~::~::~::~::~::~::~::~
~Linke~Maustaste~::~::~::~::~::~::~::~::~::~
~Mosaic~Abstürze~::~::~::~::~::~::~::~::~::~
~Overdrive~CD-ROM~::~::~::~::~::~::~::~::~::~
~Pools~für~M2Amiga~::~::~::~::~::~::~::~::~::~
Programm-Lokalitäten
~Richtig~konfiguriert~::~::~::~::~::~::~::~::~::~
RockRidge ist besser
~Sand~im~Getriebe~::~::~::~::~::~::~::~::~::~
~Schneller~Diskettensatz~::~::~
~Schnellstart~Fields~Of~Battle~
~Speicher~sparen~::~::~::~::~::~::~::~::~::~
~Suchen~lassen~::~::~::~::~::~::~::~::~::~
~Syquest,~die~dritte~::~::~::~::~::~::~::~::~::~
System() richtig genutzt
~UPD~auf~Zack~::~::~::~::~::~::~::~::~::~
Wie voll ist RAM:?
XiPaint-CD läuft
~Zeichen~setzen~::~::~::~::~::~::~::~::~::~
~Zufällig~kodiert~::~::~::~::~::~::~::~::~::~
