

RSys12

COLLABORATORS

	<i>TITLE :</i> RSys12		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 30, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RSys12	1
1.1	RSys12.guide	1
1.2	RSys12.guide/Was ist RSys?	1
1.3	RSys12.guide/Installation von RSys	2
1.4	RSys12.guide/Aufruf vom CLI	3
1.5	RSys12.guide/Aufruf von der WorkBench	4
1.6	RSys12.guide/Die ENV-Variablen	6
1.7	RSys12.guide/Die Help-Datei	7
1.8	RSys12.guide/Die Actions-Datei	7
1.9	RSys12.guide/Gadgets - Menüs - das Icon	8
1.10	RSys12.guide/Das Hauptfenster	9
1.11	RSys12.guide/Die Gadgets und das ListView	9
1.12	RSys12.guide/Task-Liste der ExecBase	10
1.13	RSys12.guide/Library-Liste der ExecBase	11
1.14	RSys12.guide/Memory-Liste in der ExecBase	12
1.15	RSys12.guide/Port-Liste in der ExecBase	12
1.16	RSys12.guide/Volume-Liste der DOSBase	13
1.17	RSys12.guide/Assign-Liste der DOSBase	14
1.18	RSys12.guide/Font-Liste der GfxBase	14
1.19	RSys12.guide/Resourcen-Liste der ExecBase	15
1.20	RSys12.guide/Interrupt-Liste der ExecBase	16
1.21	RSys12.guide/Window-Liste der IntuitionBase	16
1.22	RSys12.guide/Screen-Liste der IntuitionBase	17
1.23	RSys12.guide/Inpuhandler-Liste	17
1.24	RSys12.guide/Allgemeine Systeminfo	18
1.25	RSys12.guide/Hardware-Liste der ExpansionBase	18
1.26	RSys12.guide/Abspeichern einer Liste	19
1.27	RSys12.guide/Springen zwischen Public Screens	19
1.28	RSys12.guide/Das Modifizieren	20
1.29	RSys12.guide/Das Menü	21

1.30	RSys12.guide/Project	21
1.31	RSys12.guide/Other Lists	22
1.32	RSys12.guide/Exec	22
1.33	RSys12.guide/DOS	24
1.34	RSys12.guide/Graphics	25
1.35	RSys12.guide/Diskfont	25
1.36	RSys12.guide/Intuition	26
1.37	RSys12.guide/Preferences	27
1.38	RSys12.guide/Other Libraries	29
1.39	RSys12.guide/Actions	30
1.40	RSys12.guide/Internal List	33
1.41	RSys12.guide/Flushing	34
1.42	RSys12.guide/Clip	34
1.43	RSys12.guide/Flags	34
1.44	RSys12.guide/Das Icon	35
1.45	RSys12.guide/Das Actioning	36
1.46	RSys12.guide/Die Hotkey-Aktivierung	36
1.47	RSys12.guide/Die Utilities von RSys	37
1.48	RSys12.guide/Verschlüsseln von Daten und Dateien	37
1.49	RSys12.guide/Dateien suchen	38
1.50	RSys12.guide/CRC-Checksummen berechnen	39
1.51	RSys12.guide/Fenster in IFF-Dateien speichern	39
1.52	RSys12.guide/Der Text-Formatter	39
1.53	RSys12.guide/RSys und Hintergrundinfos	40
1.54	RSys12.guide/Tips und Tricks	43
1.55	RSys12.guide/Weitere Pläne	45
1.56	RSys12.guide/Copyrights und Verwendung	46
1.57	RSys12.guide/Danksagungen	47
1.58	RSys12.guide/Verwendete Software	48
1.59	RSys12.guide/History	49
1.60	RSys12.guide/Anhang	53

Chapter 1

R Sys12

1.1 R Sys12.guide

```
#####          #####          #####          #####
##          ##  ##          ##          ##          ##
##          ##  ##          ##          ##          ##
##          ##  ##          ##          ##          ##
#####          #####          ##          ##  #####
##  ##          ##  ##          ##  ##          ##
##  ##          ##  ##          ##  ##          ##
##          ##  ##          ##  #####          ##
##          ###  #####          ##          #####
#####          ##          #####          #####
#####          ##          #####          #####
```

Was ist R Sys?
Installation von R Sys
Gadgets - Menüs - das Icon
Die Utilities von R Sys
Der Text-Formatter
R Sys und Hintergrundinfos
Tips und Tricks
Weitere Pläne
Copyrights und Verwendung
Danksagungen
Verwendete Software
History
Anhang

1.2 R Sys12.guide/Was ist R Sys?

Was ist R Sys?

'RSYS' ist von allem ein bißchen `:-)`

Die Hauptanregung zu 'RSys' bekam ich durch die verschiedenen Systemutilities, wie 'Xoper', 'ARTM', 'TaskX' und andere. Die einen erlaubten einen tiefgreifenden Einblick in das Betriebssystem ('Xoper'), waren aber etwas unkomfortabel in der Bedienung, andere wiederum boten zwar viel Information ('ARTM'), waren dann aber nicht ganz absturzsicher und boten einfach zuviel, worunter die Übersicht litt. Wieder andere boten noch andere Features, wie WorkBench-Unterstützung und Application Icons (AppIcons), bzw. -Windows (Toolmanager), aber keine Systeminformationen. Bei fast allen Programmen aber war kein Quellcode dabei, weswegen wirklich gute Programmierlösungen nur den Programmierern vorbehalten war.

Aus diesen Überlegungen heraus entstand 'RSYS'. 'RSYS' hat etwas von 'ARTM' und 'Xoper', aber auch ein wenig vom 'Toolmanager' und von 'Icon'. Eines hat aber 'RSYS' den anderen Programmen voraus: der Sourcecode ist dokumentiert und im Lieferumfang dabei.

Bei der Source legte ich Wert auf eine gute Dokumentation und die Lesbarkeit der Programmteile. Allerdings war ich auch etwas faul, was eine ausführliche Dokumentation angeht. Das Programm habe ich völlig in Public Domain gestellt, bitte aber dennoch (natürlich :-)) um konstruktive Kritik. Sie soll dazu beitragen, Verbesserungen einzubauen und Fehler auszumerzen.

Ziemlich am Ende dieser Dokumentation habe ich auch einige Hintergrundinfos zu den einzelnen Systemlisten und -objekten geschrieben (RSys und Hintergrundinfos). Außerdem ist in den Quelltexten einiges an Informationen zu den einzelnen Programmpunkten ausgeführt, so daß sich ein Blick in das Programm immer lohnt.

1.3 RSys12.guide/Installation von RSys

Installation von RSys

Die Installation von 'RSYS' gestaltet sich recht einfach. Zum einen ist 'RSYS' ein CLI-Programm, d.h. es kann vom CLI wie ein AmigaDOS-Befehl aufgerufen und gleich wieder verlassen werden.

Zum anderen ist es ein Befehl für das 'wbstartup'-Verzeichnis der Bootpartition. Wird das Icon und das Programm in dieses Verzeichnis gelegt, so erscheint ein neues Icon auf der WorkBench, das wie eine verkleinerte Form des Hauptfensters von 'RSYS' aussieht. Durch einen Doppelklick auf das Icon oder einen CLI-Aufruf verschwindet das Icon und das Hauptfenster erscheint. Wenn 'RSYS' schon im Hintergrund läuft, d.h. vom 'wbstartup'-Verzeichnis aus geladen wurde, wird das Programm nach einem weiteren Aufruf aus dem CLI natürlich viel schneller erscheinen.

Ab Version 1.2 kann die Installation vollständig mit dem 'Installer' vorgenommen werden. Dem Programmpaket liegt ein entsprechender Script-File bei ('RSys12.Install'). Mit ihm könnt Ihr nicht nur alle Programmteile und Dateien installieren, sondern auch einzelne Teile des Programmpaketes nachinstallieren. Der 'Installer' ist seit kurzem frei verfügbar, liegt aber

'RSYS' nicht bei. Er sollte in jeder guten Mailbox oder bei einem PD-Händler erhältlich sein.

Entsprechend der verschiedenen Verwendungsmöglichkeiten gestaltet sich auch die Installation von 'RSYS' (falls sie von Hand vorgenommen werden soll), die jetzt im Einzelnen besprochen wird.

Aufruf vom CLI
 Aufruf von der WorkBench
 Die ENV-Variablen

1.4 Rsys12.guide/Aufruf vom CLI

Aufruf vom CLI
 =====

Möchtet Ihr 'RSYS' als reinen CLI-Befehl installieren, so kopiert einfach das Programm 'RSYS' in das 'C:'-Verzeichnis Eures Bootdevices, also i.a. in 'SYS:C'. 'RSYS' kann dann wie ein normaler CLI-Befehl verwendet werden.

Mit dem Aufruf

'RSYS' ?

bekommt ihr gemäß den AmigaDOS-Konventionen eine Kommandoschablone (Template) angezeigt. Dieses hat folgendes Format:

```
'RSYS' 1.0 (07-Aug-92) [11:47] (c) by Rolf Boehme
T=TASKS/S,L=LIBRARIES/S,M=MEMORY/S,P=PORTS/S,V=VOLUMES/S,A=ASSIGNS/S,
F=FONTS/S,R=RESOURCES/S,I=INTERRUPTS/S,W=WINDOWS/S,S=SCREENS/S,
H=INPUTHDL/S
```

Ihr könnt beim Aufruf entweder die Kurzform angeben, oder die vollständige Form, z.B.

'RSYS' T oder 'RSYS' TASKS

Beide Aufrufe haben den gleichen Effekt. Der Typ '/S' sagt, daß es sich um Schalter handelt, d.h. jeder Parameter veranlaßt 'RSYS' zu einer entsprechenden Ausgabe im ListView des Hauptfensters (Die Gadgets und das ListView).

Es können mehrere Schalter angegeben werden, ist aber ziemlich zwecklos, da nur der erste Parameter gilt und "behandelt" wird. Die Aufrufe

'RSYS' T L P R A und 'RSYS' T

haben also den gleichen Effekt.

Das Programm könnt Ihr dann mit einem Druck auf die ESC-Taste sofort verlassen. Falls Ihr 'RSYS' einfach mal kleiner machen wollt, könnt ihr das Zoomgadget rechts oben verwenden. Allerdings bleibt das Start-CLI bis zur

Beendigung des Programms immer noch gesperrt.

Ab 'RSYS' 1.2 kann man das Programm nicht mehr in den Iconify-Zustand bringen, wenn es von einem CLI aus aufgerufen wurde. Iconify ist jetzt nur möglich, wenn 'RSYS' vom 'wbstartup'-Verzeichnis oder über die WorkBench aufgerufen wurde.

1.5 RSys12.guide/Aufruf von der WorkBench

Aufruf von der WorkBench

=====

Dazu wird 'RSYS' und die mitgelieferte Info-Datei 'RSys.info' in das 'wbstartup'-Verzeichnis der Bootpartition oder -diskette kopiert.

Der Aufruf von der WorkBench aus über das 'wbstartup'-Verzeichnis bietet ein paar mehr Möglichkeiten. Zunächst aber zu den Parametern vom Icon, den *ToolTypes*. 'RSYS' unterstützt in dieser Version mehrere ToolTypes. Im folgenden sind sie alle aufgelistet, zunächst die Tastenkombinationen der verschiedenen Utilities von 'RSYS':

- Tooltype: CX_HOTKEY (DEFAULT: 'lshift rshift y')
Mit dieser Tastenkombination kann 'RSYS' aktiviert werden, falls es im System schon existiert. Es erscheint dann das Hauptfenster von 'RSYS'.
- ToolType: CX_CRYPT (DEFAULT: 'lshift rshift k')
Es erscheint nach dieser Tastenkombination das 'Crypt Files'-Window von 'RSYS'.
- ToolType: CX_FINDFILE (DEFAULT: 'lshift rshift s')
Diese Tastenkombination startet die Benutzeroberfläche zum Suchen von Dateien.
- ToolType: CX_FORMAT (DEFAULT: 'lshift rshift f')
Es erscheint ein kleines Window, in dem der 'Format'-Befehl einfach konfiguriert und gestartet werden kann.
- ToolType: CX_HUNKLIST (DEFAULT: 'lshift rshift h')
Diese Tastenkombination läßt ein Fenster erscheinen, in dem die Hunkstruktur eines Executables aufgelistet werden kann. Sehr nützlich für Programmierer.
- ToolType: CX_CRC (DEFAULT: 'lshift rshift c')
Nach dieser Tastenkombination erscheint ein kleines Fenster, in dem die CRC-Checksummen von Dateien berechnet werden können.
- ToolType: CX_SAVEWINDOW (DEFAULT: 'lshift rshift w')
Durch diese Tastenkombination erscheint ein Window, in dem Windows, die im System geöffnet worden sind, als IFF-Dateien abgespeichert werden können.

Im weiteren folgen die Tooltypes, mit denen die Benutzungseigenschaften von

'RSYS' beeinflusst werden können:

- ToolType: NOFASTMODE (DEFAULT: OFF)
Das Haupt-ListView von 'RSYS' (Die Gadgets und das ListView) wird normalerweise erst dann aktualisiert, wenn alle Einträge erzeugt worden sind. Wird dieses Tooltype gesetzt, wird die Anzeige während der Erzeugung der Listeneinträge aktualisiert. Bei größeren Listen kann es dann etwas länger dauern, bis die Anzeige fertig ist (deshalb *NO*FASTMODE).
 - ToolType: NOSORT (DEFAULT: OFF)
Alle Listen, bis auf die, die in der Liste selber noch Unterpunkte oder Überschriften haben, werden alphabetisch sortiert angezeigt. Mit diesem Flag kann die Sortierung der Listeneinträge abgeschaltet werden.
 - ToolType: WORKINGBAR (DEFAULT: OFF)
Mit diesem Flag wird die Anzeige eines Statusbalkens eingeschaltet. Während eine Liste im ListView aktualisiert wird, läuft ein Statusbalken mit. Für den Balken wird der Bereich des 'Informationen / Messages'-Gadgets (Die Gadgets und das ListView) verwendet.
 - ToolType: MOUSEWINDOW (DEFAULT: ON)
Ist dieses Flag gesetzt, werden alle Fenster von 'RSYS' relativ zur Mausposition geöffnet.
 - ToolType: TOPAZFONT (DEFAULT: OFF)
'RSYS' ist so programmiert, daß es theoretisch mit *jedem* Font verwendet werden kann. Ist der Font zu groß, wird automatisch der Systemfont 'Topaz 8' verwendet. Ist dieses Tooltype aktiv, wird sofort 'Topaz 8' verwendet. Das ist dann nützlich, wenn als Bildschirmzeichensatz entweder ein zu großer Font, oder aber ein proportionaler Font gewählt wurde. Die ListView-Einträge werden mit Leerzeichen formatiert, da Tabulatoren in fast jedem Font als kleine Rechtecke gezeichnet werden. Infolgedessen sieht die Formatierung unter einem proportionalen Font etwas zerrissen aus.
 - ToolType: OUTWINDOW (DEFAULT: -)
Über das AppIcon von 'RSYS' können Aktionen gestartet werden, insbesondere durch den Aufruf von externen Programmen (Die Actions-Datei). Dieser Tooltype legt das Ein- und Ausgabefenster von 'RSYS' fest.
 - ToolType: NOAPPICON (DEFAULT: EIN)
Damit wird verhindert, daß ein AppIcon (Das Icon) auf der WorkBench erscheint. Natürlich ist damit auch das Actioning nicht mehr möglich.
 - ToolType: BACKPATTERNCOL (DEFAULT: 0)
Das Hauptfenster kann ähnlich wie bei den Systemrequestern, mit einem Hintergrundmuster versehen werden. Die Farbe kann mit diesem Tooltype festgelegt werden.
 - ToolType: BACKGROUNDCOL (DEFAULT: 0)
Die Hintergrundfarbe des Pattern-Musters kann hier gesetzt werden. Abhängig von der Auflösung des Bildschirms kann beispielsweise ein leicht gelbliches Muster erzeugt werden, wenn BACKPATTERNCOL auf
-

Schwarz und BACKGROUNDCOL auf Weiß gesetzt wird.

- ToolType: BACKPATTERN (DEFAULT: 5555AAAA)

Hiermit wird das Hintergrundmuster selbst festgelegt. Die anzugebende long-Hexzahl legt die "erste" und "zweite Reihe" des Musters fest, das dann für das Füllen des Fensters verwendet wird. Welches Muster dabei heraus kommt, kann man sich auf folgende Weise einfach vorstellen. Die Hexzahl schreibt man sich in der binären Darstellung mit 32 Stellen hin. Für den Default-Wert sieht die Darstellung so aus:

5555AAAA = 01010101010101011010101010101010

Nun teilt man diese long-Zahl in zwei Hälften und schreibt sie untereinander, also so:

5555AAAA = 0101010101010101
1010101010101010

Die gesetzten Pixel sind die Einsen. Man sieht, daß sich aus diesem Wert ein 'Schachbrett-Muster' ergibt, das auch bei den Systemrequestern verwendet wird. Mit dem Muster 0000FFFF kann man sich übrigens ein waagerecht gestreiftes Muster, mit CCCCCCCC ein senkrecht gestreiftes erzeugen lassen.

- ToolType: DETAILPEN (DEFAULT: 0)

- ToolType: BLOCKPEN (DEFAULT: 1)

- ToolType: MENUPEN (DEFAULT: 0)

Mit diesen ToolTypes kann man die Farben des Hauptwindows und des Menüs des Hauptwindows festlegen. Eine nette Farbvariation ist 2, 3, 1, Reihenfolge, wie oben angeführt.

Wurde 'RSYS' von der WorkBench aus gestartet, erscheint ein neues Icon. Wird es doppelgeklickt, erscheint das Hauptfenster von 'RSYS'. Den gleichen Effekt hat ein erneuter Aufruf vom CLI. Danach wird jedoch das CLI wieder freigegeben, da 'RSYS' bereits gestartet wurde.

Ab Version 1.2 sind *alle* 'RSYS'-Fenster mit diesem Patternmuster versehen.

1.6 RSys12.guide/Die ENV-Variablen

Die ENV-Variablen

=====

'RSYS' benötigt zwei Hilfsdateien. In der einen steht in Textform eine kurzer Online-Hilfe (Die Help-Datei), in der anderen sogenannte 'Aktionseinträge', die ausgeführt werden, falls ein Icon über das AppIcon (Das Icon) gezogen und der Dateityp erkannt wurde, stehen.

Dazu sollten zwei ENV:-Variablen definiert werden: 'RSYSHELP' und 'RSYSACTION'. Die Variablen enthalten gültige und vollständige Dateibezeichnungen (mit Dateipfad!) für die entsprechenden Dateien. Sollten die eine oder beide Variablen fehlen, so wird 'RSYS' bei entsprechender Aktivierung darauf hinweisen.

Doch nun zu den Formaten beider Hilfsdateien.

Die Help-Datei
Die Actions-Datei

1.7 RSys12.guide/Die Help-Datei

Die Help-Datei

Nach der Betätigung der Help-Taste im Hauptfenster von Main oder nach Anklicken des entsprechenden Menüpunktes wird diese Datei einfach in das ListView eingelesen und dargestellt.

Die Datei ist eine einfache Textdatei mit einem rechten Rand von höchstens 45 Zeichen. Die erste Zeile enthält nur die maximale Anzahl der in dieser Datei enthaltenen Zeilen minus eins.

Die Help-Datei hat den Default-Namen 'RSys.HLP' kann aber unter Verwendung der 'ENV: '-Variablen (Die ENV-Variablen) leicht geändert werden. Auch der Inhalt kann den eigenen Bedürfnissen einfach angepasst werden.

1.8 RSys12.guide/Die Actions-Datei

Die Actions-Datei

Erscheint auf der WorkBench das AppIcon, so können in bekannter Manier eine oder mehrere WorkBench-Icons (also Directories, Dateien, Disc-Icons, etc.) auf dieses Icon gezogen werden. Daraufhin erscheint ein kleines Window mit mehreren Gadgets, das Action Window.

Im Gegensatz zur Hilfedatei müssen die Einträge in der Aktionsdatei in einem bestimmten Format stehen. In der ersten Zeile steht wieder die Anzahl der Zeilen in der Datei minus 1. Die Schablone für eine solche Aktionszeile lautet folgendermaßen:

```
<lfd.Nr>#<Typ der Datei>#<Eintrag im ListView>#<Befehl>#<Wait/Close>#
```

Alle Teile werden durch ein '#' voneinander getrennt.

Die 'lfd.Nr' und der 'Typ der Datei' entsprechen der Numerierung und Deklaration in dem Programm. Sie sollten tunlichst nicht verändert werden. Der 'Typ der Datei' sollte maximal 20 Zeichen lang sein.

Der 'Eintrag im ListView' erscheint in selbigem, wenn eine Dateiart erkannt wurde und ein kleines Aktionsfenster erscheint. Es dient der Information des Benutzers und kann beliebig gewählt werden, sollte jedoch nicht länger als 30 Zeichen sein.

Der 'Befehl' kann 255 Zeichen lang werden und enthält einen gültigen

Aufruf im CLI-Format. Hierbei wird der Dateiname einfach angehängt, weswegen das letzte Zeichen vor dem trennenden '#' ein Leerzeichen sein sollte.

Der letzte Parameter 'Wait/Close' bestimmt, ob nach einer Aktion das kleine Fenster und damit die Aktionen beendet werden sollen oder nicht. Hierbei steht 'W' für 'Wait' und 'C' für 'Close' nach einer Aktion. Hier ein paar Beispiele:

```
6#Assembler file#Anzeigen#OwnC:MuchMore #W#
^ ^           ^           ^           ^- Nach dem Befehl wird das
| |           |           |           Fenster nicht geschlossen.
+-+           |           +- MuchMore im C:-Verz. wird mit dem
|             |           Dateinamen aufgerufen, die Datei also
|             |           angezeigt.
|             +- Im ListView erscheint als Eintrag
|             "Anzeigen". Wird dieser Eintrag
|             geklickt, wird dieser Befehl
|             ausgeführt
+- Die Datei ist ein Assembler-File
```

Von einem Type können auch mehrere Einträge stehen. Dann werden entsprechend mehr ListView-Einträge angeboten. Also z.B.

```
4#C Sourcefile#Bearbeiten#OwnC:ed #W#
4#C Sourcefile#Anzeigen#OwnC:MuchMore #W#
4#C Sourcefile#Compilieren#CC:cc #W#
```

Hier erscheinen drei Einträge im ListView, die dann entsprechend gewählt und ausgeführt werden können.

Die Action-Datei hat voreinstellungsmäßig den Namen 'RSys.ACT' kann aber unter Verwendung der 'ENV:'-Variablen leicht geändert werden. Auch der Inhalt kann den eigenen Bedürfnissen einfach angepasst werden.

1.9 RSys12.guide/Gadgets - Menüs - das Icon

Gadgets - Menüs - das Icon

Wie aus dem Vorangegangenen ersichtlich, kann man bei 'RSYS' zwei verschiedene Zustände unterscheiden. Zum einen erscheint das Hauptfenster mit diversen Anzeigen und einem Menü, zum anderen erscheint auf der WorkBench ein Icon, das verschiedene Aktionen ausführen kann. Beide Möglichkeiten werden hier vorgestellt.

```
Das Hauptfenster
Die Gadgets und das ListView
Das Modifizieren
Das Menü
Das Icon
```

1.10 RSys12.guide/Das Hauptfenster

Das Hauptfenster

=====

Nach einem Aufruf vom CLI oder einem Doppelklick auf das AppIcon erscheint das Hauptfenster von 'RSYS'. Es ist grob in fünf Bereiche unterteilt:



Der Bereich 'Tabellenkopf / Titelzeile' enthält entsprechend die Spaltenbezeichnung der ListView-Einträge im Anzeigebereich. Im 'Gadget'- und 'Menü'-Bereich können die gewünschten Listen angewählt werden. Allgemeine Informationen erscheinen einzellig in dem Bereich 'Informationen / Messages'. Die einzelnen Bereiche sollen nun genauer beschrieben werden.

1.11 RSys12.guide/Die Gadgets und das ListView

Die Gadgets und das ListView

=====

Im ListView werden die angeforderten Systemlisten angezeigt. Die entsprechenden Einträge im ListView können modifiziert werden, indem der Eintrag angeklickt wird. Falls eine Veränderung der Einträge erlaubt ist, erscheint dann ein Systemrequester mit weiteren Informationen, oder im Bereich 'Informationen/Messages' eine Mitteilung, dass Veränderungen nicht möglich sind.

Die einzelnen Gadgets und die damit erzeugten Listen, die dann im ListView gezeigt werden, nun im Einzelnen.

Task-Liste der ExecBase
Library-Liste der ExecBase

Memory-Liste in der ExecBase
 Port-Liste in der ExecBase
 Volume-Liste der DOSBase
 Assign-Liste der DOSBase
 Font-Liste der GfxBase
 Ressourcen-Liste der ExecBase
 Interrupt-Liste der ExecBase
 Window-Liste der IntuitionBase
 Screen-Liste der IntuitionBase
 Inpuhandler-Liste
 Allgemeine Systeminfo
 Hardware-Liste der ExpansionBase
 Abspeichern einer Liste
 Springen zwischen Public Screens

1.12 RSys12.guide/Task-Liste der ExecBase

Task-Liste der ExecBase

- Gadget: Tasks (TASTE: T KOMMANDOPARAMETER: T)

Beispielausgabe:

Taskname	Typ	Pri	Stack	Stat	CLI
ConClip Process	Proc	0	4000	Wait	2
^	^	^	^	^	^
					+ - CLI-Nummer, falls es
					sich um einen Prozeß handelt
					+ - Status des Prozesses/Tasks (P/T)
					Inv : Invalid. P/T ist "kaputt"
					Add : P/T wird gerade erzeugt
					Run : P/T wird gerade abgearbeitet,
					in jedem Falle der aktive
					P/T, also 'RSYS'!
					Rdy : Prozeß ist fertig zum Start
					Wait: P/T wartet auf Event
					Excp: Task hat eine Exception
					ausgelöst
					Rem : Task wird gerade entfernt
					+ - Stack des P/T
					+ - Priorität des Tasks
					+ - Typ des Eintrags
					Proc : Eintrag ist ein Prozeß
					Task : Eintrag ist ein Task
					+ - Name des P/T

Aktionen: Nach dem Anklicken erscheint ein Systemrequester mit vielen Daten zu dem geklickten Eintrag und sieben Gadgets.

* 'Continue':

Es wird zum Hauptfenster zurückgekehrt.

* '^C, ^D, ^E, ^F':

Die entsprechenden 'Break-Signale' werden an den P/T gesendet.

* 'Taskpri':

Über ein erscheinendes Window kann die Priorität des P/T verändert werden. Das Close- und Cancelgadget beenden die Eingabe ohne die Priorität zu ändern; 'Ok' übernimmt die neue Priorität und setzt den P/T entsprechend. Im Stringgadget kann die Eingabe direkt geschehen.

* 'Kill task!:'

Zunächst werden an den P/T alle möglichen Signale (also '^C' bis '^F') gesendet. Falls der P/T immer noch existiert, wird nach einer Sicherheitsabfrage der Task per "RemTask()" entfernt. Danach werden alle gefunden Objekte, die mit dem Task in Verbindung stehen (Screens, Windows, Ports), aus dem System entfernt.

1.13 RSys12.guide/Library-Liste der ExecBase

Library-Liste der ExecBase

- Gadget: Libraries (TASTE: L, KOMMANDOPARAMETER: L)

Beispielausgabe:

Libraryname	Version	OpenCnt	O
kd_freq.library	3.13	1	*
^	^	^	^
			+-- Falls ein * erscheint, ist
			die entsprechende Lib offen
			und wird von mindestens einem
			Programm benutzt
			+-- Anzahl, wie oft die Lib
			geöffnet wurde. Ist die Anzahl
			0, kann die Lib aus dem System
			entfernt werden
			+-- Versions- und Revisionsnummer der
			Library
			+-- Name der Library

Aktionen: Nach dem Anklicken erscheint ein Systemrequester mit drei Gadgets:

* 'Continue':

Es wird zum Hauptfenster zurückgekehrt.

* 'Flush Lib':

Es wird nach einer Sicherheitsabfrage versucht, die Library aus dem System zu entfernen.

1.14 RSys12.guide/Memory-Liste in der ExecBase

Memory-Liste in der ExecBase

- Gadget: Memory (TASTE: M, KOMMANDOPARAMETER: M)

Beispielausgabe:

```

Memory address      Size      Type
-----
Gesamt : 4717504 C: 295944 F: 2042112 <- Gesamtspeicheranzeige
Largest:           C: 270288 F: 2014344 <- Größter zusammenhg. Speicher
----- Fragmentierung -----
00200000           4194272 FAST
00212840              8 FAST-CHUNK
^                   ^ ^
|                   | +- Typ des Eintrags
|                   |   FAST           : Fast-Memory (gesamt)
|                   |   FAST-CHUNK      : Typ des Memoryeintrags
|                   |   CHIP            : Chip-Memory (gesamt)
|                   |   CHIP-CHUNK      : Typ des Memoryeintrags
|                   | +- Größe des Listeneintrags
+- Adresse des Listeneintrags

```

Aktionen: Diese Liste zeigt die Fragmentierung des Speichers im einzelnen an. Jedes einzelne Speicherfragment ist mit seiner Adresse, der Größe und dem Typ angegeben. AmigaOS verwaltet seinen Speicher in minimal acht Byte-Blöcken. Die Anzahl der acht Byte-Blöcke ist also ein Maß für die Fragmentierung des Speichers.

1.15 RSys12.guide/Port-Liste in der ExecBase

Port-Liste in der ExecBase

- Gadget: Ports (TASTE: P, KOMMANDOPARAMETER: P)

Beispielausgabe:

```

Portname           Signal-Task      Pri
-----
AREXX              REXXMaster         0
^                   ^
|                   | +- Priorität des Ports
|                   |
|                   | +- Name des Tasks, der die Signale vom
|                   |   Port verarbeitet
+- Name des Ports

```

Aktionen: Es erscheint ein Systemrequester mit zwei Gadgets:

* 'Continue':

Es wird zum Hauptfenster zurückgekehrt.

* 'Kill Port':

Nach einer Sicherheitsabfrage wird der Port aus der Systemliste entfernt.

1.16 RSys12.guide/Volume-Liste der DOSBase

Volume-Liste der DOSBase

- Gadget: Volumes (TASTE: V, KOMMANDOPARAMETER: V)

Beispielausgabe:

```

Volumename      K-Free K-Used %Used F Sta Typ
-----
AmigaDOS:        76    6084   98% * R/O FFS
^               ^      ^      ^  ^  ^
|               |      |      |  |  | +- Typ des Volumes
|               |      |      |  |  | FFS : FastFileSystem
|               |      |      |  |  | DOS : Standard DOS-Filesystem
|               |      |      |  |  | BAD : Unformatierte Disk,
|               |      |      |  |  |        unlesbar
|               |      |      |  |  | NDO : Keine DOS-Disk
|               |      |      |  |  | KIC : Kickstart-Diskette
|               |      |      |  |  | MSD : MS-DOS-Disk
|               |      |      |  |  | ??? : Sollte NIE auftreten :)
|               |      |      |  | +- Status des Volumes
|               |      |      |  | R/O : Read Only
|               |      |      |  | VAL : Volume wird gerade
|               |      |      |  |        validiert
|               |      |      |  | R/W : Read/Write
|               |      |      |  | ??? : Sonstiger Status
|               |      |      | +- Falls die Volume mehr als zu
|               |      |      |        90% voll ist, steht da ein '*'
|               |      |      +- Anzahl der benutzen KByte der Partition
|               |      |        oder Diskette in Prozent
|               |      +- Anzahl der benutzen KByte der Partition
|               |        oder Diskette in KBytes
|               +- Anzahl der freien Kapazität in KBytes
+- Name des Volumes

```

Aktionen: Nach dem Anklicken erscheint ein Systemrequester mit zwei Gadgets.

* 'Continue':

Es wird zum Hauptfenster zurückgekehrt.

* 'Tree':

Es erscheint ein neues Fenster mit einem ListView und zwei ausgeschalteten Gadgets. Nach dem Fensteraufbau wird sofort mit dem Einlesen der Disk-Verzeichnis-Struktur begonnen. Das Einlesen kann mit einem Anklicken des Closegadgets links oben und der daraufhin erscheinenden Sicherheitsabfrage abgebrochen werden. Nach dem Einlesen werden die zwei

Gadgets aktiviert.

Das linke Gadget ist ein Cycle-Gadget, mit dem zwischen zwei Darstellungen der Verzeichnisstruktur gewechselt werden kann. Voreinstellung ist die 'Tree'-Darstellung, also die Anzeige des Verzeichnisbaumes. Nach dem Anklicken wechselt die Darstellung auf die Anzeige der vollständigen Pfadnamen. Befindet sich der Zeichensatz 'IBM.font' im 'FONTS:'-Verzeichnis, werden die Graphikzeichen dieses Fonts verwendet, um die Verzeichnisstruktur darzustellen.

Das Gadget 'Save Tree' bietet nach dem Anklicken einen Filerequester an, mit welchem Ihr die Datei auswählen könnt, in der Ihr die momentan aktive Darstellung ('Tree' oder 'Path') abspeichern könnt.

Das Fenster kann über das Closegadget geschlossen werden.

1.17 RSys12.guide/Assign-Liste der DOSBase

Assign-Liste der DOSBase

- Gadget: Assigns (TASTE: A, KOMMANDOPARAMETER: A)

Beispielausgabe:

Assign	Directory
BATCH:	AmigaDos:c
^	^
	+ - Zu dem Assign gehörender Datei/Verzeichnisname
+ - Name des Assigns	

Aktionen: Siehe Volumes-Gadget.

1.18 RSys12.guide/Font-Liste der GfxBase

Font-Liste der GfxBase

- Gadget: Fonts (TASTE: F, KOMMANDOPARAMETER: F)

Beispielausgabe:

Fontname	Typ	Y	X	Low	Upp
topaz.font	ROM	8	8	32	255
^	^	^	^	^	^
					+ - Höchster ASCII-Wert, für den die
					Zeichen dieses Fonts definiert
					worden sind
				+ -	Niedrigster ASCII-Wert, für den
					die Zeichen in diesem Font definiert

```

|                               |   |   |   worden sind
|                               |   |   +- Breite eines Zeichens in diesem
|                               |   |   Zeichensatz
|                               |   +- Höhe eines Zeichens in diesem
|                               |   Zeichensatz
|   +- Art des Zeichensatzes:
|       ROM : Der Zeichensatz wurde von einem
|             Programm geladen oder ist ein
|             Systemfont
|       Disk : Der Font wurde noch nicht geladen
|             und muß von Diskette oder Festplatte
|             geladen werden
+- Name des Fonts

```

Aktionen: Nach dem Anklicken eines Eintrages im ListView erscheint ein Requester mit drei Gadgets:

* 'Continue':

Es wird zum Hauptfenster zurückgekehrt.

* 'Show Font':

Es erscheint ein Fenster mit den vier Gadgets 'Normal', 'Italic', 'Bold' und 'Underl'. Nach dem Anklicken eines dieser Toggle-Gadgets wird der Beispieltext (im Format '(<fontsize> <fontname>') entsprechend normal, kursiv, fett oder unterstrichen dargestellt. Das Gadget 'Normal' inaktiviert alle anderen Gadgets.

* 'Remove Font':

Nach einer Sicherheitsabfrage wird der Font aus dem System entfernt.

1.19 RSys12.guide/Resourcen-Liste der ExecBase

Resourcen-Liste der ExecBase

- Gadget: Resourcen (TASTE: R, KOMMANDOPARAMETER: R)

Beispielausgabe:

Resourcenname	Version	OpenCnt
Snap.resource	32.16	0
^	^	^
		+- Anzahl, wie oft die Resource
		geöffnet wurde
	+- Versions-	und Revisionsnummer der
	Resourcen	
+- Name der Resource		

Aktionen: Alle im System verfügbaren Resourcen werden angezeigt.

1.20 Rsys12.guide/Interrupt-Liste der ExecBase

Interrupt-Liste der ExecBase

- Gadget: Interrupts (TASTE: I, KOMMANDOPARAMETER: I)

Beispielausgabe:

Interruptname	Pri	Data	Code
disk.resource	0	00203624	00F8F774
^	^	^	^
			+-- Falls einer der 16 Interrupts
			aktiviert wird, wird der Code, dessen
			Adresse angezeigt wird, ausgeführt
			+-- Der Zeiger auf den privaten Datenbereich der
			Interruptroutine
			+-- Priorität dieses Interrupts
			+-- Handler oder Resource, der zu diesem Interrupt gehört

Aktionen: Die sechzehn Interrupts des Amigas werden unter Angabe des Namens, der Priorität und der Adressen der eventuell vorhandenen Interrupthandler aufgelistet.

1.21 Rsys12.guide/Window-Liste der IntuitionBase

Window-Liste der IntuitionBase

- Gadget: Windows (TASTE: W, KOMMANDOPARAMETER: W)

Beispielausgabe:

Windowtitle	H	W	X	Y	Screen
AmigaShell	100	640	0	0	Workbench
^	^	^	^	^	^
					+-- Name des Screens, zu dem dieses Window
					gehört
					+-- Y-Koordinate der Position des Windows
					+-- X-Koordinate der Position des Windows
					+-- Breite des Windows
					+-- Höhe des Windows
					+-- Titel des Windows. Steht hier ein '<no title>', so hat das Window
					keine Titelzeile

Aktionen: Nach dem Anklicken eines Eintrages im ListView erscheint ein Requester mit zwei Gadgets:

* 'Continue':

Es wird zum Hauptfenster zurückgekehrt.

* 'Close Window':

Nach einer Sicherheitsabfrage wird versucht, das Window zu schließen.

1.22 RSys12.guide/Screen-Liste der IntuitionBase

Screen-Liste der IntuitionBase

- Gadget: Screens (TASTE: S, KOMMANDOPARAMETER: S)

Beispielausgabe:

Screen-Title	H	W	X	Y	Typ
Workbench Screen	272	684	0	0	W C
^	^	^	^	^	^
					+ - Typ des angeklickten Screens.
					Mögliche Anzeigen:
					W : Workbench-Screen
					P : Public Screen
					C : Custom-Screen
					+ - Y-Koordinate der Position des
					Screens
					+ - X-Koordinate der Position des Screens
					+ - Breite des Screens
					+ - Höhe des Screens
					+ - Titelzeile des Screens

Aktionen: Nach dem Anklicken eines Eintrages im ListView erscheint ein Requester mit zwei Gadgets:

* 'Continue':

Es wird zum Hauptfenster zurückgekehrt.

* 'Close Screen':

Nach einer Sicherheitsabfrage wird versucht, den Screen mit allen assoziierten Windows zu schließen.

1.23 RSys12.guide/Inputhandler-Liste

Inputhandler-Liste

- Gadget: HandlerInp (TASTE: H, KOMMANDOPARAMETER: H)

Beispielausgabe:

Inputhandler name	Pri	Data	Code
Snap Input Handler	52	534E4150	003A46DE

```

^
|
|
|
|
|
|
|
+- Name des Inputhandlers

^ ^
| |
| |
| |
| +- Zeiger auf den privaten Datenbereich des
| Inputhandlers
+- Priorität des Inputhandlers

```

Aktionen: Alle über das 'input.device' direkt angemeldeten Input Handler werden angezeigt. Unter AmigaOS 2.0 übernimmt die 'commodities.library' einen Großteil der Handler.

1.24 Rsys12.guide/Allgemeine Systeminfo

Allgemeine Systeminfo

- Gadget: Systeminfo (TASTE: Y, KOMMANDOPARAMETER: -)

Beispielausgabe:

```

                        Systeminfo
-----
14.07.1993, 195. day, 28. week <- Datum, Tag und Woche
Tasks      : 30                <- Anzahl der entsprechenden
Libraries   : 24                Systemobjekte
Ports       : 9
Volumes     : 11
Assigns     : 36
Fonts       : 5
Resources   : 9
Windows     : 4
Screens     : 2

```

Aktionen: Anzeige aller Systemobjekte (Tasks, Libraries, etc.) sowie Datum, Woche.

1.25 Rsys12.guide/Hardware-Liste der ExpansionBase

Hardware-Liste der ExpansionBase

- Gadget: Hardware (TASTE: D, KOMMANDOPARAMETER: -)

Beispielausgabe:

```

                        Hardware/Expansion
-----
----- Hardware
Processor      : CPU 68030          <- Prozessortyp
Coprocessor    : FPU 68882          <- FPU-Typ

```

```

Memory Unit      : MMU 68030          <- MMU-Typ
Denise           : Normal Denise (8362) <- Custom Chips
Agnus            : Normal Agnus
Chipset          : No AA-Chipset      <- Falls hier das AA-Chipset
                                         erkannt wurde, sehen hier
                                         noch mehr Infos
----- Expansion
Type             : RAM Expansion      <- Beschreibung der
Company          : C= West Chester    Hardware-Erweiterung
Name             : A2630 MC68030 (2MB Stand.)
Init Action      : Need Driver
                  > : Private processed
Diagnosis        : No action attached
Manuf.nr.        : 514               <- Hardware-Kennnummern
Prod.nr.         : 81
Serial           : 0x00000000
Boardaddr.       : 0x00200000
Boardsize        : 2.097.152         <- Hier steht die evtl.
Slotaddr.        : 0x00000020        vorhandene Speichergröße
Slotsize         : 32                der Erweiterung

Summary          : 1 expansions found

```

Aktionen: Anzeige des Prozessor- und Co-Prozessortyps, wie er in der ExecBase verzeichnet ist, und aller internen Erweiterungen, dessen Einträge aus der ExpansionBase-Struktur entnommen wurden.

1.26 RSys12.guide/Abspeichern einer Liste

Abspeichern einer Liste

- Gadget: Save List (TASTE: -, KOMMANDOPARAMETER: -)

Aktionen: Die aktuelle angezeigte Liste im ListView wird nach einer Abfrage mit einem Filerequester in die gewünschte Datei abgespeichert.

1.27 RSys12.guide/Springen zwischen Public Screens

Springen zwischen Public Screens

- Gadget: Jump (TASTE: J, KOMMANDOPARAMETER: -)

Aktionen: 'RSYS' schließt das Hauptfenster, sucht in der Liste der Public Screens nach dem nächsten geöffneten Screen, öffnet das Hauptfenster auf diesem und holt dann den Public Screen nach vorne. Existiert kein weiterer Public Screen, so erscheint eine entsprechende Meldung in einem Systemrequester.

1.28 RSys12.guide/Das Modifizieren

Das Modifizieren

=====

In den folgenden Listen ist eine Modifikation oder Anzeige von weiteren Eigenschaften der Einträge möglich: Tasks, Libraries, Ports, Volumes, Assignes, Fonts, Ressources, Windows und Screens.

Die entsprechenden Datenstrukturen werden mit maximaler Sicherheit modifiziert, d.h., es wird im Zweifelsfalle mehrmals geprüft, ob das bearbeitete Objekt vor und nach der Modifikation existiert. Da alle Systemlisten nur *snapshot-artig* ausgelesen und angezeigt werden können, ist in bestimmten Fällen eine Modifikation nur eingeschränkt oder gar nicht möglich. Das ist beispielsweise der Fall, wenn ein Window modifiziert werden soll, das von dem entsprechenden Task schon geschlossen wurde.

Als Kontrollkriterium wird primär die Adresse – diese ist als einzige im System für dieses Objekt eindeutig! – und sekundär der Name des jeweiligen Objekts herangezogen, also bei Windows die Titelzeile, bei Tasks der Taskname etc. Tritt in einem der beiden Fälle ein Fehler auf, so wird die Modifikation mit einer Fehlermeldung abgebrochen. Eine Modifikation der Objekte wird erst gar nicht begonnen, falls in dem Namensfeld eines Objektes 'RSYS' ein '-' eingetragen hat. Diese Objekte können nicht eindeutig zugeordnet werden, weshalb diese also auch nicht modifiziert werden können.

Da hierbei auf das verwendete Schutzprotokoll geachtet werden muß, hier ein paar Hinweise dazu.

Intuition-Objekte werden beim Auslesen der Daten mit dem Protokoll "LockIBase() / UnlockIBase()" geschützt. Damit werden die jeweiligen Listen vor der Veränderung durch Intuition-Routinen während des Auslesens ausreichend geschützt. Bei der Veränderung der Objekte durch 'RSYS' ist selten ein Schutzprotokoll erforderlich, da die Routinen von Intuition dieses meistens selbst erledigen.

Alle Exec-Objekte, wie Tasks, Ports, Libraries, etc., werden während des Auslesens durch ein "Forbid()/Permit()", im Falle von Tasks, sogar durch ein "Disable()/Enable()" vor Veränderung durch Systemroutinen geschützt. Im Falle der Tasks ist zum Auslesen der Taskzeiger das Protokoll "Disable()/Enable()" zu verwenden, zum Auslesen der Taskstruktur reicht jedoch ein "Forbid()/Permit()".

"Disable()/Enable()" sollte deswegen verwendet werden, da die Systemliste in der ExecBase durch den interruptgesteuerten Task-Scheduler laufend in ihrer Anordnung geändert wird (man denke nur an die Aktivierung eines Tasks, also die Umsetzung des entsprechenden Taskknotenzeigers aus der Wait-Liste in die Ready-Liste und dann in den Running-Zustand [ExecBase->ThisTask-Eintrag]).

Die Task-Struktur selber kann jedoch nur von einem Task oder Prozeß aus verändert werden. Daher reicht zum Auslesen der Taskstruktur das Protokoll "Forbid()/Permit()" aus. Daraus ergibt sich also folgendes Schema:

Reservieren der eigenen Struktur-Speicherbereiche

```
"Forbid()";
```

```
"Disable()";
```

Auslesen und Merken der Tasknodes

```
"Enable()";
```

Auslesen der Taskstrukturen in eigene Strukturen (*ohne* Verwendung von DOS-Routinen, also KEIN FGets(), Open() etc.)

```
"Permit()";
```

Auswerten der eigenen Strukturen

Freigeben der eigenen Struktur-Speicherbereiche

Für die Implementation dieser Routinen könnt Ihr den dokumentierten Quelltext einsehen.

1.29 RSys12.guide/Das Menü

Das Menü

=====

In dem Menü befinden sich neben dem Window-Handling noch die Einträge für weitere Systemlisten – die aber seltener gebraucht werden – und das Handling einer internen Liste. Weiterhin können vom Menü aus die Utilities gestartet und ein paar Änderungen an der Konfiguration von 'RSYS' vorgenommen werden. Hier die Menüpunkte im einzelnen:

```
Project
Other Lists
Actions
Internal List
Flushing
Clip
Flags
```

1.30 RSys12.guide/Project

Project

- Menüpunkt: Iconify (TASTENSEQUENZ: 'RAMIGA-I')
Aktion: Nach dem Aktivieren dieses Menüpunktes (direkt, ESC-Taste oder 'RAMIGA-I') wird das Hauptfenster geschlossen, und es erscheint auf der WorkBench ein AppIcon. Wurde 'RSYS' vom CLI aus gestartet, wird durch ESC das Programm beendet.

- Menüpunkt: Help (TASTENSEQUENZ: 'RAMIGA-H')
Aktion: Die Environment-Variable 'RSYSHELP' wird ausgelesen und der Eintrag als vollständige Dateibezeichnung der Help-Datei von 'RSYS' verwendet. Diese Datei wird ausgelesen und im ListView dargestellt.
- Menüpunkt: Reload actionfile (TASTENSEQUENZ: -)
Aktion: Wurde die Aktionsdatei verändert, so kann sie mit diesem Menüpunkt erneut gelesen werden.
- Menüpunkt: About... (TASTENSEQUENZ: -)
Aktion: In einem Systemrequester erscheint eine kurze Information zu diesem Programm und die Adresse des Autors.
- Menüpunkt: Quit 'RSYS'... (TASTENSEQUENZ: 'RAMIGA-Q')
Aktion: Das Programm wird mit einer Sicherheitsabfrage beendet. Bei dem erscheinenden Systemrequester sind drei Möglichkeiten wählbar:
 - * 'Ja': Das Programm wird beendet.
 - * 'Iconify': Das Hauptfenster verschwindet und ein AppIcon erscheint auf der WorkBench.
 - * 'Nein': Rückkehr zum Hauptwindow.

1.31 RSys12.guide/Other Lists

Other Lists

Ab 'RSYS' 1.2 wurde das Menü mit den weiteren verfügbaren Listen umgestellt. Die Listen sind jetzt nach der entsprechenden Library geordnet.

```

Exec
DOS
Graphics
Diskfont
Intuition
Preferences
Other Libraries

```

1.32 RSys12.guide/Exec

Exec

....

- Menüpunkt: Software Devices (TASTENSEQUENZ: 'RAMIGA-1')
Beispielausgabe:

Devicename	Version	OpenCnt

audio.device	37.10	0	
^	^	^	
			--- Anzahl der Tasks oder
			Prozesse, die das
			Device öffneten
			--- Versions und Revisionsnummer
--- Name des Devices			

Aktionen: Alle Handler und Treiber werden angezeigt, also die Software Devices, die Handler oder Treiber für die Hardware darstellen.

- Menüpunkt: Vectors (TASTENSEQUENZ: 'RAMIGA-2')

Beispielausgabe:

```

                        System Vectors
-----
----- Capture vectors
ColdCapture      : 0x00000000
CoolCapture      : 0x00000000
WarmCapture      : 0x00000000

----- Kick pointer
KickMemPtr       : 0x00000000
KickTagPtr       : 0x00000000
KickChecksum     : 0x00000000

----- Exec special pointer
Debug entry      : 0x00F82D58
ResModules       : 0x002003E0
MMU-Lock         : 0x00000000

```

Aktion: Die Capture-Vektoren, Kick-Pointer und die speziellen Vektoren der ExecBase werden angezeigt. Die Capture-Vektoren können einen Hinweis auf einen Virus sein, falls diese nicht 'NULL' sind.

- Menüpunkt: CLI processes (TASTENSEQUENZ: -)

Beispielausgabe:

CLI nr.	Name	Pri	Stack	
CLI 2	ConClip Process	0	4000	
^	^	^	^	
				--- Stack-Größe des CLI's
				--- Priorität des Prozesses
	--- Name des CLI-Prozesses			
--- Nummer des Prozesses in der Liste der CLI's				

Aktion: AmigaOS 2.0 erlaubt ein einfaches Auslesen der geöffneten CLI's. Alle geöffneten CLI's werden im ListView angezeigt. Die Numerierung ist systembedingt nicht fortlaufend.

- Menüpunkt: Resident programs (TASTENSEQUENZ: -)

Beispielausgabe:

Resident name	UseCount	Type	
Alias	-2	INTERNAL	
^	^	^	
			--- Typ des residenten Kommandos
			INTERNAL : Ist in einer Shell

```

|                                     | eingebaut
|                                     |     EXTERNAL : Mit 'Resident'
|                                     | eingebunden
|                                     +-- Anzahl der Benutzungen dieses Befehls.
|                                     Ist er -1, handelt es sich um einen
|                                     Handler oder CLI. -2 steht für einen
|                                     eingebauten Befehl.
|                                     Alle Zahlen > 0 geben die
|                                     Benutzungsanzahl an
+-- Name des Handlers oder Befehls

```

Aktion: Die gezeigte Liste entspricht fast der Ausgabe des Befehls 'resident' des AmigaOS 2.0. Jedoch werden hier *alle* residenten Programme aufgelistet, auch die, die systemintern vorhanden sind.

- Menüpunkt: Semaphores (TASTENSEQUENZ: -)

Beispielausgabe:

```

Semaphore name          Pri Ownertask
-----
« SetPatch »           0 <no task>
^                      ^ ^
|                      | +-- Name des Besitzer-Tasks
|                      |     der Semaphore
|                      +-- Priorität der Semaphore
+-- Name der Semaphore

```

Aktion: Die Semaphore-Liste der ExecBase wird ausgelesen und dargestellt.

1.33 RSys12.guide/DOS

DOS

...

- Menüpunkt: Volume Locks (TASTENSEQUENZ: 'RAMIGA-3')

Beispielliste:

```

File                      Access
-----
AmigaDOS:                 READ
^                          ^
|                          | +-- Art des Locks:
|                          |     READ : Zum Lesen geöffnet,
|                          |           auch mehrere Zugriffe
|                          |           möglich (shared)
|                          |     WRITE: Exklusiv-Zugriff
|                          |     -----: Unbekannter Typ
+-- Pfadbezeichnung, auf dem der
    Lock existiert

```

Aktion: Alle vom System reservierten Dateien werden angezeigt. Das sind alle Verzeichnisse, die durch ein "Open()" oder ein "Lock()" gehalten werden.

- Menüpunkt: Physical Devices (TASTENSEQUENZ: 'RAMIGA-2')

Beispielausgabe:

```

Device name   Driver/Handler
-----
PIPE:         L:Queue-Handler
^             ^
|             |
|             |--- Name des Handlers oder Treibers
+--- Name des Devices

```

Aktion: Es werden alle Devices angezeigt, die hardware-bezogen vorkommen, also Schnittstellen, Festplatten, Floppies, etc.

1.34 RSys12.guide/Graphics

Graphics

.....

- Menüpunkt: Display Modes (TASTENSEQUENZ: 'RAMIGA-4')

Beispielliste:

Name	Depth	MinR	MaxR
PAL:High Res Lace	4	32,1	1008,1024
^	^	^	^
			--- Maximale Größe des Screenrasters
			Screenrasters
		--- Minimale Größe des Screenrasters	
	--- Tiefe des Screens, der Screen hat dann 2^n		
	Farben zur Verfügung, in diesem Fall also		
	2^4 = 16 Farben		
--- Name des Monitortreibers			

Aktion: Alle im System installierten Monitortreiber werden angezeigt.

- Menüpunkt: Blitter Wait Queue (TASTENSEQUENZ: -)

Aktion: Die Tasks, die auf den Blitter warten, werden angezeigt.

- Menüpunkt: TOF Wait Queue (TASTENSEQUENZ: -)

Aktion: Die Tasks, die auf das Top-Of-Frame warten, werden angezeigt.

1.35 RSys12.guide/Diskfont

Diskfont

.....

- Menüpunkt: Fonts in FONTS: (TASTENSEQUENZ: 'RAMIGA-5')

Beispielausgabe:

Font name	Type	Y	Style
KaraGRANITE.font	DISK	55	N C
^	^	^	^
			--- Style des Fonts:
			N : Normal

```

|           |           |           U      : Unterstrichen
|           |           |           B      : Bold
|           |           |           I      : Kursiv (italic)
|           |           |           E      : Extended
|           |           |           C      : Colour-Font
|           |           |           T: Tagged
|           |           |           +-- Größe des Fonts in Pixel
|           |           |           +-- Typ des Fonts, ROM- oder DISK-Font
+-- Name des Fonts

```

Aktion: Das logische Verzeichnis 'FONTS:' wird ausgelesen und die ermittelten Fonts angezeigt. Dazu gehören auch die Fonts, die bereits in das System eingebunden sind. Nach einem Anklicken eines ListView-Eintrages wird der Font in einem eigenen Fenster angezeigt.

1.36 RSys12.guide/Intuition

Intuition

.....

- Menüpunkt: Public Screens (TASTENSEQUENZ: 'RAMIGA-6')

Aktion: Hier werden alle im System geöffneten Public Screens, also die Screens, die öffentlich und für jedes Programm zugänglich sind, angezeigt. Die Anzeige besteht aus der Adresse, dem Namen, der Visitor-Anzahl (die Anzahl der geöffneten Windows auf diesem Screen) und dem zugehörigen Signal-Task, die aus der 'PublicScreenNode'-Struktur ausgelesen werden können.

- Menüpunkt: Gadgets (TASTENSEQUENZ: -)

Beispielliste:

Gadget text	Typ	PosX	PosY	W	H

* SCR: Workbench Screen					
<has image>	SYSWUP	-22	0	23	11
** WIN: RU 1.2 FreeWare!					
Ja	REQBOO	14	26	35	9
^ ^	^ ^	^	^	^	^
					+-- Höhe des Gadgets
					+-- Breite des Gadgets
					+-- Y-Position der linken oberen
					Ecke des Gadgets
					+-- X-Position der linken oberen
					Ecke des Gadgets
		+-- Aktivierungstyp des Gadgets:			
		WDR: Window dragging			
		SDR: Screen dragging			
		WUP: Window upfront			
		SUP: Screen upfront			
		WDO: Window downback			
		SDO: Screen downback			
		CLO: Close Gadget			
		BOO: Boolean Gadget			
		G02: Dummy typ			
		PRO: Proportional Gadget			

```

| |          | STR: String Gadget
| |          | CUS: Custom Gadget
| |          | SIZ: Sizing Gadget
| |          | UNK: Unbekannter Typ
| |          +-- Objekttyp des Gadgets:
| |          | SYS: System Gadget
| |          | SCR: Screen Gadget
| |          | GZZ: Gimmezerozero Gadget
| |          | REQ: Requester Gadget
| |          | CUS: Andere Gadgets
| +-- Window oder Screen-Titelzeile
+-- Gadgettext, falls einer existiert, sonst '<has image>' oder
    ein '-', falls ein Image existiert oder kein Text vorhanden
    ist.

```

Aktion: Nach dem Anklicken werden alle Gadgets auf allen Screens und in allen Windows ermittelt und die wichtigsten Daten angezeigt. Während dieser Aktion kann es passieren, daß man nicht weiterarbeiten kann. Das ist normal! Damit die Intuition-Listen systemkonform durchsucht werden können, muß während dieser Zeit die IntuitionBase mit dem Paar "LockIBase()/UnlockIBase()" gesperrt werden. Der Zeitraum der Sperrung ist aber i.a. so kurz, daß er vom Benutzer kaum wahrgenommen wird.

1.37 RSys12.guide/Preferences

Preferences

.....

- Menüpunkt: Preferences (TASTENSEQUENZ: 'RAMIGA-7')
- Beispielausgabe:

```

                                Preferences
-----
----- Input/Keyboard:
Keyboard repeat speed   : 0 sec 6000    $\mu$ sec
Keyboard repeat delay   : 0 sec 400000  $\mu$ sec
Double click time       : 0 sec 800000  $\mu$ sec

----- Pointer
"Hot spot" X            : -1
"Hot spot" Y            : 0
Pointer Sprite Col 0    : 0x0E44
Pointer Sprite Col 1    : 0x0000
Pointer Sprite Col 2    : 0x0EEC
Pointer Sensitivity     : 1

----- Workbench
Height default font     : 8
Standard Color 0        : 0x0AAA
Standard Color 1        : 0x0000
Standard Color 2        : 0x0FFF
Standard Color 3        : 0x009A
View X-Offset           : -26
View Y-Offset           : -7
View init offset X      : 129

```

```
View init offset Y      : 44
Height default font     : 8
CLI availability         : CLI enabled
Workbench interlaced    : No
Override WB width       : 0
Override WB height      : 0
Override WB depth       : 0
Extension Info size     : 0
```

----- Printer

```
Printer port           : Parallel port
Printer type           : Custom
Printer Driver          : HP_LaserJet
Print pitch            : 0
Print quality          : Letter
Lines per inch         : 6 LPI
Left margin            : 0
Right margin           : 75 Chars
Print image            : Positive
Print aspect           : Horizontal
Print shade            : Color
Print Threshold        : 7
Paper length           : 66 Rows
Paper type             : Single paper
Temp file for printer  :
Affect Rows/Columns    : 10
User preference flags  : 0x0000080F
Max width of picture   : 0 inch
Max height of picture  : 0 inch
Offset of picture      : 0 inch
Print density          : 4
```

----- Serial configuration

```
Baud rate              : MIDI (31.250 baud)
Read bits              : 8
Write bits             : 8
Stop bits              : 1
Parity                 : None
Handshake              : RTS/CTS
```

Aktion: Die Preferences-Struktur wird ausgelesen und mit Klartext versehen angezeigt.

- Menüpunkt: Printer Data (TASTENSEQUENZ: -)
Beispielausgabe:

Printer data

```
-----
Printer name          : HP_LaserJet
Version              : 35
Revision             : 59
Printer class         : 1
Color class          : 1
Print columns        : 0
Char sets            : 0
Print head pins       : 1
Max X-dots in dump    : 2325 dots
Max Y-dots in dump    : 3198 dots
```



```
Density X      : 300 dots/inch
Density Y      : 300 dots/inch
Timeout        : 60 sec
```

Aktion: Alle Daten in der Preferences-Datei, speziell die des Druckertreibers, werden angezeigt.

- Menüpunkt: ENV-Variables (TASTENSEQUENZ: -)

Beispielausgabe:

```
ENV-Variable  Content
-----
FREE_DRIVES   RAM:,df0:,df1:,vd0:,dh0:,dh1
^             ^
|             +-- Inhalt der Environmentvariablen
|             <no text> : Die Variable ist keine Textdatei
|             <dir>      : Ein weiteres 'ENV:'-Verzeichnis
+-- Name der Variablen
```

Aktion: Das logische Verzeichnis 'ENV:' wird ausgelesen und die Inhalte der Environmentvariablen im ListView dargestellt. Hierbei wird davon ausgegangen, daß sich in einer 'ENV'-Variablen *eine* Zeile befindet. Sollten sich mehr als eine Zeile in der entsprechenden Datei befinden, werden die nachfolgenden Zeilen nicht ausgelesen.

Ab Version 1.2 werden die Dateien überprüft, ob eine Textdatei vorliegt. Es kommt immer mal vor, daß auch Binärdateien im 'ENV:'-Verzeichnis zu finden sind. Die Ausgabe dieser binären Daten ist etwas unsauber.

1.38 RSys12.guide/Other Libraries

Other Libraries

.....

- Menüpunkt: Commodities (TASTENSEQUENZ: 'RAMIGA-8')

Beispielausgabe:

```
Commodity name      Prio
-----
RSys                0
^                   ^
|                   +-- Priorität des Brokereintrags
+-- Name des Brokers
```

Aktion: Alle Commodities, die auch im Fenster des Programmes 'Exchange' erscheinen, werden angezeigt. Dazu wird eine undokumentierte Funktion verwendet, die aber in der 'commodities.library' enthalten ist.

Anm.: Unter OS 3.0 wird ein Zeiger auf die Liste der Libraries zurückgegeben. Diese Funktion hat wurde also irgendwohin verlagert. Ein gutes Beispiel dafür, daß man nicht ungestraft undokumentierte Funktionen verwendet!

Anm. 2: Ich habe einen legalen Weg herausgefunden, wie man sich eine Liste aller Commodity-Einträge besorgen kann. Dazu verwende ich einfach den Broker von RSys. Beim Einrichten des Brokers wird ein Zeiger auf ein

CxObj zurückgegeben. Diesen *caste* (konvertiere) ich auf einen 'struct Node'-Zeiger um und erhalte damit, analog zu den Input-Handlern, einen Einsprung in die Liste der angemeldeten Broker-Objekte! Jetzt brauche ich nur noch zu dem Anfang der Liste zu hangeln (Commodore erhalte uns die doppelt verketteten Listen! :-)) und von dort aus die Liste erneut durchzuscannen. Schon habe ich meine Liste! Und alles durch eine 'undokumentierte Vermutung', ich liebe 'C'!

- Menüpunkt: Amigaguide XRef (TASTENSEQUENZ: 'RAMIGA-9')

Beispielliste:

XRef keyword	Type	File	Line
Gadget	STRU	+intuition.h	216
^	^	^^	^
			++- Zeile des Includefiles,
			in der die Struktur
			vorkommt
		++-	Dateibezeichnung Referenzfiles
			Falls die Bezeichnung länger ist, wird
			ein '+' anstelle des Pfades gesetzt
		++-	Der Typ des Eintrags. Hier können
			folgende Werte stehen:
		'LINK'	: Amigaguide Link
		'FUNC'	: Funktionsbeschreibung
			(z.B. C-Routine)
		'COMM'	: Befehlsbeschreibung
		'INCL'	: Include-Datei eines Compilers
		'MACR'	: Macro-Beschreibung
		'STRU'	: Struktur-Beschreibung
			(z.B struct Gadget)
		'FIEL'	: Ein Feld einer Struktur
		'TYPD'	: Datentyp-Beschreibung
			(z.B. typedef)
		'DEFI'	: #define-Macro-Beschreibung
		++-	Schlüsselwort des Objektes

Aktion: Mit dem Befehl LoadXRef des Amigaguide-Entwicklersystems können Kreuzreferenzen in den Speicher des Amigas geladen werden. Diese stehen dann bei jeder Amigaguide Session dem Benutzer zur Verfügung. Welche Informationen geladen sind, kann man mit diesem Menüpunkt erfahren.

1.39 RSys12.guide/Actions

Actions

- Menüpunkt: CRC Check

Aktion: Es erscheint ein kleines Window mit mehreren Gadgets. Mit dem 'Select'-Gadget kann man in dem erscheinenden Filerequester eine Datei auswählen, von der die CRC-Checksumme berechnet werden soll. In dem Cycle-Gadget kann man den Modus, CRC-16 oder CRC-32, einstellen. Hierbei ist zu beachten, daß die CRC-16-Methode nur für Dateien mit einer Größe kleiner als 65536 Bytes verwendet werden kann.

Für größere Dateien ist die 'CRC32'-Methode zu wählen. Ist die Datei gewählt (der Dateiname erscheint in dem 'File'-Gadget), wird mit dem 'Compute'-Gadget die Berechnung gestartet. Die berechnete CRC-Summe erscheint dann in hexadezimaler Form im 'CRC'-Gadget.

- Menüpunkt: Crypt

Aktion: Es erscheint wiederum ein kleines Window mit acht Gadgets. Im 'Source'-Gadget kann die Quelldatei, die kodiert werden soll, direkt eingegeben, oder mit dem danebenliegenden Gadget per Filerequester ausgewählt werden. Ebenso bei dem 'Dest'-Gadget, in dem der Name der Datei eingetragen wird, in der die kodierten Daten gespeichert werden. Es werden zwei Modi zur Verschlüsselung von Text- oder Binärdateien angeboten.

Die Modi können in dem Cyclegadget ausgewählt werden. 'Crypt' verwendet die Verschlüsselung von Dateien mittels einem *Zellular-Automaten*. Diese Routine habe ich aus Olaf 'Olsen' Barthels 'term'-Source geratzt, weswegen er u.a. auch ganz unten bei den Danksagungen erwähnt wird. Während 'Crypt' auf jede Art von Dateien angewendet werden kann, wurde die 'ROT 13'-Methode nur für Textdateien implementiert. Dabei werden einfach die Buchstaben eines Textes um 13 Zeichen "verschoben". Hierbei ist auch nicht möglich, wie bei 'Crypt' ein Paßwort zu verwenden.

- Menüpunkt: List Hunks

Aktion: Es erscheint ein Window mit drei Gadgets, einem ListView und zwei Booleangadgets, 'Load' und 'Save List'. Zunächst wird mit 'Load' und dem dann erscheinenden Filerequester eine Datei ausgewählt. Danach wird diese Datei (i.a. eine ausführbare oder ladbare Datei) bezüglich ihrer Hunk-Struktur analysiert. Die ermittelten Daten erscheinen dann in dem ListView. Wenn es sich um eine Library oder eine Objektbibliothek handelt, kann die Ausgabe etwas dauern. Deshalb kann man die Aktion jederzeit durch das Anklicken des Closegadgets und nach einer Sicherheitsabfrage abbrechen. Mit dem 'Save List'-Gadget kann dann die Liste in eine auszuwählende Datei abgespeichert werden.

- Menüpunkt: Find File

Aktion: Nach der Anwahl dieses Menüpunktes erscheint ein großes Window mit zehn Gadgets. Diese werden jetzt der Reihe nach beschrieben:

- ListView: Devices

In diesem ListView stehen alle angemeldeten Devices und Assigns. Wenn ein Eintrag angeklickt wird, erscheint der Eintrag in dem danebenliegenden ListView.

- ListView: Selected Devices

Damit werden alle die Verzeichnisse angewählt auf denen Dateien gesucht werden sollen. Jeder Eintrag, der in den beiden ListViews angeklickt wird, erscheint automatisch in dem anderen ListView. Bei einem Eintrag mit einem '-' davor handelt es sich um ein Assign. Ich habe diese Kennzeichnung deshalb so gewählt, weil ich dann entsprechend die Einträge sortieren kann. Erst die *echten*, dann die *logischen* Devices.

- Cycle-Gadget: Case sense/No case sense

Hier kann gewählt werden, ob die Dateien *case sensitive* (Groß- und Kleinschreibung wird berücksichtigt) oder *case insensitive* (Groß- und Kleinschreibung werden nicht unterschieden) gesucht

werden. Lautet z.B. das Suchmuster 'Test.exe', so wird mit 'No case sense' sowohl die Datei 'Test.exe', als auch die Datei 'TeSt.ExE' gefunden.

- Boolean Gadget: Sel all
 - Boolean Gadget: Sel none
Damit können alle Einträge vom linken ListView in das rechte gebracht werden ('Sel all': alle Verzeichnisse werden durchsucht) und umgekehrt ('Sel none': Kein Verzeichnis soll durchsucht werden).
 - Boolean Gadget: Start/Stop Search
Wurden Verzeichnisse ausgewählt und im Pattern-Gadget ein Suchmuster eingegeben, so ist dieses Gadget anwählbar und startet die Suche in den Verzeichnissen des 'Selected Devices'-ListViews. Wird das Gadget während eine Suche angeklickt, wird nach einer Sicherheitabfrage der Suchvorgang abgebrochen.
 - Boolean Gadget: Save found list
Nach der Anwahl erscheint ein Filerequester, in dem die Namen der gefundenen Dateien in eine auszuwählende Datei abgespeichert werden können.
 - String Gadget: Pattern
Hier kann ein Suchmuster (Dateiname oder Pattern) eingegeben werden. Hierbei gelten die üblichen Wildcards und Suchmuster des AmigaOS.
 - Cycle Gadget: Dirs + Files/Files/Directories
Hier kann gewählt werden, was für ein Typ von Datei gefunden werden soll.
 - ListView: Found files/dirs
In diesem ListView erscheinen alle gefunden Einträge mit vollständiger Pfadbezeichnung. Diese Liste kann nach dem Anklicken des 'Save found list'-Gadgets in eine Datei abgespeichert werden.
 - Menüpunkt: Format Disk
Aktion: Mit 'Format Disk' wird eine kleine Schnittstelle zum 'Format'-Befehl des AmigaDOS angeboten. Die einzelnen Gadgets werden im folgenden näher beschrieben.
 - Stringgadget: Format
In diesem Gadget steht der vollständige Pfadprogrammname des zu verwendenden 'Format'-Befehls. Dieser lautet üblicherweise 'SYS:System/Format'.
Mit dem nebenstehenden 'Select'-Gadget kann das 'Format'-Programm mit einem Filerequester ausgewählt werden.
 - ListView: Devices
In diesem ListView erscheinen alle formatierbaren Devices alphabetisch geordnet. Das ausgewählte Device wird dann im zugehörigen Textgadget darunter angezeigt.
 - Stringgadget: Disk label
Hier wird der Name des Mediums (voreingestellt und systemkonform 'Empty') eingetragen.
-

- Cycle: Icons/No Icons
- Cycle: OFS/FFS
- Cycle: International/No international
Mit diesen Cyclegadgets können weitere Formatierungsvorgaben ausgewählt werden, also ob ein 'Trashcan' auf dem formatierten Medium erzeugt, das Medium mit dem alten Filesystem oder dem Fastfilesystem formatiert und ob es im 'internationalen Modus' formatiert werden soll.
- Stringgadget: <ohne Namen>
In diesem Stringgadget ohne Namen steht der vollständige Befehl, der nach Festlegung aller Kriterien als eigenständiger Prozeß ausgeführt wird.
- Booleangadgets: Format
- Booleangadget: Quick
Mit diesen Gadgets wird nach einer Sicherheitsabfrage das 'Format'-Programm gestartet. Mit 'Format' wird das Medium auch physikalisch formatiert, mit 'Quick' werden nur die Root-Blöcke überschrieben und so die eventuell vorhandenen Daten überschrieben.
- Menüpunkt: Save windows
Aktion: Nach der Auswahl dieses Menüpunktes erscheint ein kleines Fenster mit drei Gadgets.
 - ListView: Windows
In diesem ListView erscheinen alle geöffneten Windows, die im System bekannt sind. Dabei werden auch die aufgeführt, die auf anderen Screens als dem WorkBench Screen geöffnet wurden.
 - Boolean: IFF File
Wurde ein Fenster angewählt, wird dieses Gadget aktiviert. Nach dem Anklicken erscheint ein Filerequester mit der Aufforderung, eine Datei auszuwählen, in der das angewählte Window als IFF-Datei abgespeichert werden soll.
 - Boolean: Rescan windows
Damit kann die Window-Liste aktualisiert werden. Dies geschieht nach dem Abspeichern eines Windows automatisch.
- Menüpunkt: Check Libraries
Aktion: Alle im ROM angelegten Libraries werden gecheckt, ob sie von anderen Programmen gepatcht wurden. Sehr nützlich, um eventuell installierte Viren aufzuspüren.

1.40 RSys12.guide/Internal List

Internal List

- Menüpunkt: Save current list (TASTENSEQUENZ: 'RAMIGA-S')
Aktion: Die angezeigte Liste wird in eine 'RSYS'-interne Liste kopiert und gespeichert. Der Inhalt bleibt erhalten, bis eine neue Liste abgespeichert wird, der Menüpunkt 'Kill saved list'

gewählt wurde oder das Programm beendet wird.

- Menüpunkt: Restore saved list (TASTENSEQUENZ: 'RAMIGA-R')
Aktion: Die 'RSYS'-interne Liste wird in das ListView eingetragen. Die ursprüngliche Liste kann durch ein nochmaliges Anklicken des entsprechenden Gadgets wieder erzeugt werden. Die 'RSYS'-interne Liste bleibt dabei erhalten.
- Menüpunkt: Kill saved list (TASTENSEQUENZ: -)
Aktion: Eine gespeicherte 'RSYS'-interne Liste wird gelöscht und die damit verbundenen Speicherbereiche freigegeben.

1.41 RSys12.guide/Flushing

Flushing

- Menüpunkt: Flush Libraries
Aktion: Entfernt alle Libraries aus dem System, die einen Open-Count von 0 haben. Diese sind geschlossen und entfernen sich meistens nicht selbst aus dem System.
- Menüpunkt: Flush Fonts
Aktion: Entfernt alle Fonts aus dem System, die nicht mehr verwendet werden.
- Menüpunkt: Flush all libraries and Fonts
Aktion: Entspricht beiden Menüpunkten 'Flush Libraries' und 'Flush Fonts'.

1.42 RSys12.guide/Clip

Clip

- Menüpunkt: Clip entry... (TASTENSEQUENZ: 'RAMIGA-C')
Aktion: Nach Auswahl dieses Menüpunktes kann ein Eintrag im Hauptfenster von 'RSYS' angeklickt werden. Dieser Eintrag wird dann vom Programm ins Clipboard übernommen und kann von anderen Programmen gelesen und weiterverwendet werden.
- Menüpunkt: Clip list (TASTENSEQUENZ: -)
Aktion: Mit diesem Menüpunkt kann die ganze angezeigte Liste ins Clipboard kopiert werden.

1.43 RSys12.guide/Flags

Flags

- Menüpunkt: Fast Mode (TASTENSEQUENZ: 'RAMIGA-F')
Aktion: Ist dieser Modus aktiviert, wird das ListView beim Erzeugen der Liste nicht aktualisiert, sondern erst, nachdem die Liste erstellt worden ist. Ist dieser Modus nicht aktiviert, wird nach jedem Erstellen eines Eintrags das ListView aktualisiert. Default ist 'EIN'.
- Menüpunkt: Sorting (TASTENSEQUENZ: 'RAMIGA-O')
Aktion: Ist das Sorting aktiviert, sortiert 'RSYS' automatisch die Einträge des ListViews. Auf manche Listen hat dieses Flag keinen Einfluß (z.B. Memory), da es dort auf die systemgegebene Reihenfolge der Daten ankommt. Default ist 'EIN'.
- Menüpunkt: Working bar (TASTENSEQUENZ: 'RAMIGA-B')
Aktion: Mit diesem Flag kann die Anzeige eines Statusbalkens umgeschaltet werden. Während eine Systemliste im ListView erzeugt wird, zeigt der Balken immer den entsprechenden Balkenabschnitt an. Als Balkenbereich wird das 'Informationen / Messages'-Gadget (Die Gadgets und das ListView) verwendet. Default ist 'AUS'.
- Menüpunkt: Auto front window (TASTENSEQUENZ: 'RAMIGA-Z')
Aktion: Ist dieses Flag aktiviert, werden das Hauptfenster und alle anderen Fenster von 'RSYS' automatisch nach vorne geholt, falls sie geöffnet oder *gezipt* wurden.
- Menüpunkt: Mouse window (TASTENSEQUENZ: 'RAMIGA-M')
Aktion: Wenn diese Option angewählt wurde, wird jedes Fenster, das von 'RSYS' geöffnet wird, zentriert unter dem Mauszeiger erscheinen.
- Menüpunkt: Topaz font (TASTENSEQUENZ: 'RAMIGA-T')
Aktion: Wurde als System-Screen-Font ein Proportionalfont gewählt, sieht die Ausgabe im ListView etwas zerrissen aus, da die Anzeige nur mit Leerzeichen formatiert werden kann. Schaltet man diese Option ein, wird das Hauptfenster geschlossen und 'RSYS' erscheint mit dem 'Topaz-8'-Systemfont.
- Menüpunkt: Speaking (TASTENSEQUENZ: 'RAMIGA-N')
Aktion: Ist 'Speaking' eingeschaltet, werden die Ausgaben im 'Informationen / Message'-Bereich auch über das 'Narrator'-Device, also die Sprachausgabe, ausgegeben.
- Menüpunkt: Save asking (TASTENSEQUENZ: -)
Aktion: Ist dieses Menüflag gesetzt, werden alle Sicherheitsabfragen von 'RSYS' unterdrückt. Genauer: es werden alle Ja-Nein-Abfragen unterdrückt.

VORSICHT

Auch Sicherheitsabfragen, wie *Wollen Sie diesen Task killen?*

werden anstandslos mit *Ja* beantwortet!!!

1.44 RSys12.guide/Das Icon

Das Icon
=====

Das Actioning
Die Hotkey-Aktivierung

Wurde der Menüpunkt 'Iconify' gewählt oder das Programm von der WorkBench gestartet, erscheint auf dem WorkBench-Screen ein sogenanntes AppIcon. Darauf können andere WorkBench-Objekte, wie Schubladen, Textdateien, Devices etc., deren Icon sichtbar ist, gezogen werden.

Ab 'RSYS' 1.2 gibt es die Möglichkeit, durch Angabe von 'NOAPPICON' in den ToolTypes kein Icon erscheinen zu lassen. Damit ist dann auch das nachfolgend beschriebene 'Actioning' nicht durchführbar.

1.45 RSys12.guide/Das Actioning

Das Actioning

Nach dem Aktivieren des Icons durch Anklicken oder Draufziehen eines WorkBench-Objektes öffnet 'RSYS' ein kleines Fenster mit drei Gadgets. Die Titelzeile des Fensters zeigt den Namen des Objektes an, das auf das 'RSYS'-Icon gezogen wurde. Darunter wird der Typ des Objektes angezeigt. In dem ListView-Gadget werden alle Aktionen, die dem Typ dieses Objektes in der Actions-Datei zugeordnet wurden, aufgelistet. Das 'Info'-Gadget gibt weitere Informationen über das Objekt in Form eines Systemrequesters aus.

Nach der Anwahl eines ListView-Eintrages wird die entsprechende Aktion mit diesem Objekt ausgeführt. Dabei wird der Name des Objekts an den entsprechenden String in der Actions-Datei angehängt und als CLI-Befehl ausgeführt.

Hierzu ein Beispiel: Für LHA-Archive existieren folgende Einträge:

```
39#LHArc archive#Archiv listen#OwnC:LZ l #W#
39#LHArc archive#Archiv testen#OwnC:LZ t #W#
```

In der Titelzeile erscheint der Eintrag 'LHArc archive' und in dem ListView die Einträge 'Archiv listen' und 'Archiv testen'. Wird der Eintrag 'Archiv listen' angeklickt, wird der Befehl 'OwnC:LZ l <objektname>' ausgeführt.

1.46 RSys12.guide/Die Hotkey-Aktivierung

Die Hotkey-Aktivierung

In dem Icon-Zustand von 'RSYS' kann das Hauptfenster mit dem Hotkey

'lshift rshift y' wieder geöffnet werden. Desweiteren kann 'RSYS' auch mit dem Programm 'Exchange' angesteuert und auch beendet werden.

Alle Hotkeys werden voreingestellt mit den beiden Shift-Tasten zusammen definiert. Hier eine Zusammenfassung aller Hotkey-Voreinstellungen, wie sie auch den ToolTypes des 'RSYS'-Icons entnommen werden können, versehen mit ein paar Erklärungen:

- ToolType: OUTWINDOW DEFAULT: CON:/10//150/'RSYS' WINDOW/AUTO/CLOSE
Manche Aktionen von 'RSYS' erfordern einen Ein-/Ausgabe-Kanal. Diese Ein- und Ausgaben werden dann auf das angegebene CLI-Window umgeleitet.

Anm.: Dieser Eintrag ist *ZWINGEND*!!! NIL: oder das NULL:-Device sollten *NICHT* verwendet werden, wenn man einen Guru vermeiden will!

- ToolType: CX_HOTKEY DEFAULT: 'LSHIFT RSHIFT Y'
Mit diesem Hotkey kann 'RSYS' aktiviert werden, wenn das AppIcon existiert. Falls das Hauptfenster offen ist, guckt 'RSYS' nach, ob der vorderste Screen ein PublicScreen ist und öffnet das Hauptfenster auf diesem.
- ToolType: CX_CRYPT DEFAULT: 'LSHIFT RSHIFT K'
Damit wird das Verschlüsselungs-Tool von 'RSYS' aufgerufen (Menüpunkt Crypt).
- ToolType: CX_FINDFILE DEFAULT: 'LSHIFT RSHIFT S'
Damit wird das Dateisuchfenster von 'RSYS' aufgerufen (Menüpunkt Find File).
- ToolType: CX_FORMAT DEFAULT: 'LSHIFT RSHIFT F'
Damit wird das 'Format-Disk'-Fenster von 'RSYS' aufgerufen (Menüpunkt Format).

1.47 RSys12.guide/Die Utilities von RSys

Die Utilities von RSys

An dieser Stelle noch ein paar zusätzliche Informationen zu den Menüpunkten.

Verschlüsseln von Daten und Dateien
Dateien suchen
CRC-Checksummen berechnen
Fenster in IFF-Dateien speichern

1.48 RSys12.guide/Verschlüsseln von Daten und Dateien

Verschlüsseln von Daten und Dateien

=====

Bei der Verschlüsselung der Daten werden zwei Methoden verwendet. Die 'ROT 13'-Methode ist ein einfaches Verschlüsselungsverfahren, das die Buchstaben eines Textes einfach um 13 Zeichen nach vorn oder zurück gezählte Buchstaben macht. Dazu ein Beispiel: aus der Zeile

"Das ist ein Satz, der 1 mal kodiert werden soll!"

```
34 68 97 115 32 105 115 116 32 101 105 110 32 83 97 116 122 44 32 100 101
114 32 49 32 109 97 108 32 107 111 100 105 101 114 116 32 119 101 114 100
101 110 32 115 111 108 108 33 34
```

macht das 'ROT 13'-Verfahren

"Qnf vfg rva Fngm, qre 1 zny xbvreg jreqra fbyy!"

```
34 81 110 102 32 118 102 103 32 114 118 97 32 70 110 103 109 44 32 113 114
101 32 49 32 122 110 121 32 120 98 113 118 114 101 103 32 106 114 101 113
114 97 32 102 98 121 121 33 34
```

Darunter habe ich mal die entsprechenden ASCII-Kodierungen geschrieben. Es ist deutlich zu erkennen, daß nur Buchstaben kodiert (verschoben) werden. Interpunktionen und Zahlen werden nicht berücksichtigt.

Das zweite Verfahren basiert auf einem zellularen Automaten ähnlich dem der Spielsimulation 'Life'. Dabei wird ein Ring von Zellen angelegt, der dann durch ein vorgegebenes Paßwort mit Nummern aufgefüllt wird. Mit diesen Nummern schließlich, wird eine Datei kodiert. Diese Methode stammt von der Verschlüsselungsmethode, die in Olaf "Olsen" Barthels 'term' zur Kodierung des Telefonbuches verwendet wird.

'Anm.:' Das Paßwort ist überaus *WICHTIG*! Es sollte erstens nie vergessen und zweitens ein echtes Paßwort verwendet werden. Das Paßwort-Gadget sollte also nie leer sein, da sonst die Methode zufällige Verschlüsselungen erzeugt und diese Datei nie wieder restauriert werden kann!

1.49 RSys12.guide/Dateien suchen

Dateien suchen

=====

Diese Utility arbeitet intensiv mit der Funktion "ExAll()" und den Funktionen "MatchPattern()/MatchPatternNoCase()" der DOS-Library. Dabei werden zunächst alle Dateien bestimmt und währenddessen untersucht, ob es sich jeweils um ein Verzeichnis oder eine Datei handelt. Trifft dann auch das Suchmuster zu, wird es in das ListView eingetragen.

1.50 RSys12.guide/CRC-Checksummen berechnen

CRC-Checksummen berechnen

=====

Ich habe mehrere Programme, die CRC-Summen berechnen, zum Vergleich mit meinen Routinen gestartet. Alle Programme gaben für ein und dieselbe Datei andere Werte aus, als meine Routine. Ich suche den Fehler heute noch. Wenn sich also eine hilfreiche Seele finden sollte, die den Fehler findet, dieser ist eine Tüte Gummibärchen sicher `:-)`

1.51 RSys12.guide/Fenster in IFF-Dateien speichern

Fenster in IFF-Dateien speichern

=====

Hierbei kam auch eine Routine zum Einsatz, die ich von Olsens 'term' geratzt habe. Leider funktioniert Olsens Routine nicht mit allen Fenstern, was an der Verwendung der Funktion "ClipBlit()" liegt, die zwar die ausgewählte BitMap eines Windows berücksichtigt, aber die überlappenden Bereiche anderer Fenster mit Datenmüll füllt. Außerdem verwendet Olsen die BitMap des zu einem Fenster gehörenden Screen-RastPorts anstelle des Window-RastPorts.

Dieses Problem der Überlappung habe ich durch die Verwendung der Routinen "BltBitMap()" und "WindowToFront()" behoben. Damit wird das zu speichernde Fenster nach vorne geholt, weswegen auch keine Überlappungen mit anderen Fenstern auftreten. Dieses Verfahren funktioniert natürlich nicht mit Backdrop-Fenstern.

Es muß aber eine Möglichkeit geben, *nur* den Fensterinhalt abspeichern zu können. Immerhin kann es die 'DumpRastPort'-Routine des 'printer.device' auch. Wer also sachdienliche Hinweise zu diesem Thema hat, möge sich bitte mit mir in Verbindung setzen.

1.52 RSys12.guide/Der Text-Formatter

Der Text-Formatter

Ab 'RSYS' 1.2 ist es möglich, beim Abspeichern einer Liste die Ausgabe noch etwas zu formatieren. Dazu erscheint nach jedem Filerequester ein Fenster mit acht Gadgets.

- Gadget: Prefix
Hier wird eingetragen, was *vor* dem zu speichernden Listeneintrag stehen soll.
- Gadget: Text to save
In diesem Gadget wird die Formatierung des zu speichernden Listeneintrags angezeigt. Diese kann sich ändern, wenn die Gadgets 'Blank

between texts' und 'Quote-in text' gesetzt sind.

- Gadget: Postfix
Hier steht der Text, der *nach* dem Listeneintrag kommen soll.
- Gadget: Blank between texts
Ist dieser Schalter angeklickt, wird zwischen dem Prefix und dem Listeneintrag sowie zwischen dem Listeneintrag und dem Postfix ein Leerzeichen ('Blank') eingefügt.
- Gadget: Quote-in text
Ist dieser Schalter angeklickt, wird der Listeneintrag in Anführungszeichen gesetzt.
- Gadget: No header in file
Bei formatierten Ausgaben ist es manchmal nicht wünschenswert, daß eine Titelzeile in die Datei geschrieben wird. Das kann mit diesem Gadget abgeschaltet werden.

Eine Anwendung wäre zum Beispiel, wenn Ihr nach Dateien suchen laßt, die Ihr dann löschen wollt. Ihr öffnet das 'Find File'-Fenster und tragt das entsprechende Pattern ein, z.B. '#?.bak'. Die ausgewählten Devices werden dann nach solchen Dateien durchsucht. Habt Ihr alle zusammen, speichert Ihr die Datei ab. Nach dem Filerequester erscheint der 'Text-Formatter'. Dort schreibt Ihr als Prefix 'Delete' hinein. Dann schaltet Ihr noch alle Schalter ein und klickt dann auf 'Use format'. Die Einträge werden dann in dem Format

```
Delete "datei.bak"
```

abgespeichert. Diese Datei könnt Ihr dann mit 'Execute' ausführen und alle gefundenen Dateien werden gelöscht.

1.53 RSys12.guide/RSys und Hintergrundinfos

RSys und Hintergrundinfos

'RSYS' wurde unter Berücksichtigung aller mir bekannten Richtlinien der Programmierung unter AmigaOS 2.0 geschrieben. Alle kritischen Routinen wurden bis zu dreimal abgesichert. Das bläht zwar etwas den Code, gewährleistet aber die Lauffähigkeit auch unter Betriebssystemen >= 2.04.

Besonderer Wert wurde auf die Vermeidung von Speicherfehlern und 'Enforcer'-Hits gelegt. Bei einem auftretenden Speicherfehler wird in den meisten Fällen das Programm unter Angabe von Quelldatei- und Funktionsname, sowie der Zeilennummer im Quelltext abgebrochen und beendet.

Einige Listen lassen es zu, daß man die Einträge *systemkonform* modifizieren kann. Dazu gehören zum Beispiel die Prioritäten von Tasks. Andere Listenelemente können unter Beachtung wichtiger Regeln entfernt werden. Hierbei sind die sogenannten Schutzprotokolle zu berücksichtigen.

Bei der Modifizierung von Intuition-Objekten (Windows, Screens) ist das Protokollpaar "LockIBase()/UnlockIBase()" zu verwenden. Geht es nur um Veränderung der Lage von solchen Objekten, z.B. mit "WindowToFront()", so ist diese Operation schon von Intuition ausreichend geschützt. Bei allen Listen der ExecBase ist das Paar "Forbid()/Permit()" zu verwenden, das das Multitasking für die Dauer der Operationen zwischen den beiden Routinen ausschaltet.

Bei manchen Listen reicht jedoch dieses Protokoll nicht aus. Dazu gehören alle interruptgesteuerten Listen, wie die Task-, Interrupt-, Speicher- und Blitter-Queue-Listen. Diese müssen zusätzlich mit dem Protokoll "Disable()/Enable()" geschützt werden. Das Protokollpaar unterbricht die Abarbeitung von Interrupts. Da fast jedes Multitasking stark interruptabhängig ist, kann es zu einem Absturz kommen, falls dieser Zustand zu lange anhält. Schätzungen über die Zeitdauer lauten bis etwa 250 Mikrosekunden. Obwohl diese Zeit ausreicht, um auch eine große Liste auszulesen, sollte dieser Zustand nicht zu lange gehalten werden.

"Disable()/Enable()" beinhaltet damit auch logischerweise implizit ein "Forbid()/Permit()". Beide Protokolle sind schachtelbar, womit man ein 'totsicheres' Verfahren an die Hand bekommt, um Systemlisten auszulesen.

Unter diesen Vorsichtsmaßnahmen sollte es keine Probleme im Zusammenspiel mit anderen Programmen geben, bis auf eine Einschränkung: 'RSYS' kann nicht mit Programmen zusammenarbeiten, die nicht systemkonform programmiert wurden. Beispiele dafür sind Programme, die beispielsweise den Namen eines öffentlichen Ports nicht korrekt initialisieren. So kommt es beispielsweise vor, daß ein Programm zwar einen Zeiger auf einen Portnamen ungleich Null hat, diesen Zeiger aber uninitialisiert läßt und dieser dann folglich irgendwohin zeigt. Die Folge ist im harmlosesten Fall ein 'Enforcer'-Hit des Typs 'READ-BYTE' (beim Auslesen des vermeintlichen Strings), im extremsten Fall ein Guru!

Ich habe das Problem jetzt so gelöst, daß ich bei den auszulesenen Node-Namen das Typen-Flag untersuche. Steht dort der Eintrag 'NT_UNKNOWN' oder nicht das erwartete Flag, lese ich den String einfach nicht aus, sondern trage in das ListView

```
<wrong type:0>
```

ein. Hierbei steht die '0' für den ermittelten Knotentypen. Die möglichen Knotentypen sind:

Wert	Typ

0	Unbekannter Node-Typ
1	Task
2	Interrupt
3	Device
4	Message Port
5	Message
6	'freie' Message
7	Message wurde beantwortet
8	Resource
9	Library
10	Memory-Node
11	Softinterrupt

12	Font
13	Prozeß
14	Semaphor
15	Signalsemaphor
16	Boo-Node
17	Kick-Memory-Node
18	Graphics-Node (Monitor-Node z.B.)
19	Death Message (eine 'tote' Nachricht)
254	Benutzerdefinierter Node
255	Erweiterung (auch benutzerdefiniert)

Manche Systemutilities (z.B. ARTM) achten darauf nicht und produzieren 'Enforcer'-Hits en mas. Diese Fehler lassen sich auch nicht vermeiden. Der Aufwand dafür wäre einfach zu groß, da man ja praktisch Teile des Programms 'Enforcer' in das eigene Programm implementieren müßte. Solange man nicht davon ausgehen kann, daß *JEDER* Programmierer systemkonform programmiert, wird es diese Lücke auch weiterhin geben.

Ein weiterer typischer Fehler ist die Verwendung der Compiler-Funktion "strcpy()" auf Quellstrings vorher unbekannter Länge. Viele Programmierer verwenden diese Routine, um schneller Strings zu kopieren. Dabei wird nicht beachtet, daß man eventuell gar nicht soviel Speicherplatz reserviert hat, um den Quellstring aufzunehmen. Ein signifikantes Beispiel dafür ist das o.g. Port-Namen-Problem. Ist der String uninitialisiert und nicht mit 'ASCII 0' terminiert, kopiert "strcpy()" einen solchen Portnamen bis in alle Ewigkeit, bis zum Ende des Speichers, bis zur nächsten Einsprungadresse eines anderen Tasks oder bis zur nächsten Reise von Indian tours ':-)' Manche Programmierer sagen sich dann, überprüfen wir doch einfach mit "strlen()" den Quellstring. Nun, da "strlen()" auch solange zählt, bis 'ASCII 0' erkannt wurde, ist diese Methode auch für eine Auslandsreise nach Indien durchaus geeignet.

Das einfachste und probateste Mittel in diesem Fall, ist die Compiler-Funktion "strncpy()". Damit kann man einfach festlegen, wieviel Zeichen denn nun kopiert werden sollen. Das einzige, was jetzt noch stört, ist der 'READ-BYTE'-Hit den man bekommt, wenn man *merkwürdige* Adressen an "strncpy()" übergibt.

Ein weitere Fehlerquelle ist die Verwendung von "printf()" in allen seinen Erscheinungsformen ("sprintf()", "vsprintf()" etc.) im Zusammenhang mit Systemlisten. Da die "printf()" -Routinen auch auf DOS-Routinen zugreifen, sind sie zum zügigen Kopieren von mehreren Systemeinträgen ungeeignet. Die Routine "RawDoFmt()" der Exec-Library ist jedoch *sicher*. Mit ihr kann man sich selbst ein "sprintf()" zusammenbauen, was ich auch getan habe. Normalerweise befindet sich aber diese Routine in der 'amiga.lib'.

Sicher ist weiterhin die Verwendung der "str...()" -Routinen zwischen Schutzprotokollen, da diese nur Speicherbereiche kopieren oder verschieben. Möchte man es trotzdem noch schneller haben, gibt es noch die Funktionen "CopyMem()" und "CopyMemQuick()". Bei letzterer ist zu beachten, daß die Daten auf longwords ausgerichtet sein müssen. Beide Routinen gehören zur 'exec.library', können also bei Systemlistenuntersuchungen verwendet werden.

1.54 RSys12.guide/Tips und Tricks

Tips und Tricks

Diese Sparte ist zunächst leer. Sollten sich Anfragen und Tips aus dem hoffentlich großen Kreis von 'RSYS'-Benutzern, wird sich dieser Bereich füllen.

Stop! Einen Tip habe ich schon :-)

Der folgende Batch (Name 'tracer') war mir beim Testen des Programmes sehr hilfreich:

```
.KEY onoff,program/A
.bra {
.KET }

if { program } eq ""
    echo "Aufruf: tracer [ON | OFF] [program]"
    quit
endif

if { onoff } eq ON
    assign t: doks:t
    cpu nofastrom
    run >NIL: <NIL: MungWall TASK {program}
    run >NIL: <NIL: IO_Torture
    run >NIL: <NIL: enforcer ON
    run sushi <>"CON:0/190//80/Sushi CTRL-E:Empty CTRL-F:File/AUTO/CLOSE"
        ON BUFK=1024
else
    if { onoff } eq OFF
        Utilities:ProgUtils/Sushi/sushi OFF
        run >NIL: <NIL: enforcer OFF
        tbreak IO_Torture
        tbreak MungWall
        cpu fastrom
        assign t: ram:t
    else
        echo "Aufruf: tracer [ON | OFF] <[program]>"
    endif
endif
endif
```

Das Programm 'tbreak' sendet einen Break-Signal an den Task, dessen Name als Parameter beim Aufruf mit angegeben wurde. Es ist ein kleines Programm, daß ich mal quick and dirty gehackt habe. Hier kommt der kleine Quelltext dazu:

```
#include <exec/types.h>
#include <exec/exebase.h>

#include <dos/dos.h>
#include <dos/dosextens.h>

#include <clib/dos_protos.h>
```

```
#include <clib/exec_protos.h>

#include <stdio.h>
#include <string.h>

#define STRINGEND '\0'

char *
B2CStr(char *ret, BSTR inp)
{
    register int i;
    char *help = (char *)BADDR(inp);

    for (i = 0; i < *help; i++)
        *(ret + i) = *(help + i + 1);

    *(ret + i) = STRINGEND;

    return (ret);
}

int
main(int argc, char **argv)
{
    struct DOSBase *DOSBase = NULL;
    extern struct ExecBase *SysBase;
    int ret = RETURN_FAIL;

    if(DOSBase = (struct DOSBase *)OpenLibrary((STRPTR)"dos.library", 33L))
    {
        struct Node *node, *membnode[200];
        struct CommandLineInterface *CLI;
        char modname[200];
        int i, j, out;

        if(argc != 2)
            Printf((STRPTR)"Aufruf: %s <taskname>\n", argv[0]);
        else
        {
            Forbid();
            Disable();

            for (node = SysBase->TaskWait.lh_Head;
                 node->ln_Succ && i < 200; node = node->ln_Succ)
                membnode[i++] = node;

            for (node = SysBase->TaskReady.lh_Head;
                 node->ln_Succ && i < 200; node = node->ln_Succ)
                membnode[i++] = node;

            if (i < 200)
                membnode[i++] = &SysBase->ThisTask->tc_Node;

            Enable();

            out = FALSE;
```

```

    for (j = 0; j < i && !out && i < 200; j++)
    {
        struct Process *proc = (struct Process *)membernode[j];
        if(CLI = (struct CommandLineInterface *)BADDR(proc->pr_CLI))
        {
            B2CStr(modname,CLI->cli_CommandName);
            if (!strcmp(argv[1],modname))
            {
                Signal((struct Task *)proc,
                        SIGBREAKF_CTRL_C | SIGBREAKF_CTRL_D |
                        SIGBREAKF_CTRL_E | SIGBREAKF_CTRL_F);
                out = TRUE;
            }
        }
    }

    Permit();

    if(out == TRUE)
        Printf((STRPTR)"Task with command \'%s\' signaled...\n",argv[1]);
    if(i >= 200)
        PutStr((UBYTE *)"More than 200 tasks read!\n");

    ret = RETURN_OK;
}
CloseLibrary((struct Library *)DOSBase);
}

return(ret);
}

```

1.55 RSys12.guide/Weitere Pläne

Weitere Pläne

OS 2.0 gibt mir immer neue Ideen. Das merke ich jedesmal, schon während ich eine Routine programmiere. Folgendes habe ich für die nächsten Versionen geplant:

- * Ein Autolister für die Diskettenlaufwerke (Benutzeroberfläche ist schon da ;-), also Diskette rein, Inhalt automatisch lesen, Diskette raus, nächste Diskette rein usw.
- * Die Utilities werden als eigene Prozesse gestartet
- * Bei 'RSYS' verschwinden evtl. die Gadgets und der Menüpunkt "Special lists", dafür kommt ein zweites ListView.
- * Ein erweiterte Dateitypen-Erkennung, evtl. eine FileType.library
- * Evtl. eine eigene Utility.library, die auch von anderen Programmen genutzt werden kann

* Bugs beseitigen `:-)`

* AmigaGuide-Unterstützung statt Hilfsfile

Wer einen oder mehrere Fehler findet, der sollte mich schnellstmöglichst darüber informieren, wo und unter welchen Umständen es zu Problemen kam. Zu einer solchen Fehlermeldung gehören folgende Informationen über die Rechnerumgebung:

* Verwendetes Rechnermodell (A500, A500+, A1000, A2000, A2000+, A2500, A3000, A3000T, A4000, etc.)

* Speicherausbaustufe

* Prozessor-Hardware (68000, 68010, 68020, 68030, 68040, etc.)

* Graphik-Hardware (ECS, AGA, ImpactVision, Domino, etc.)

* Verwendete Betriebssystemversion und Art (A3000 ROM, A3000 Kickfile, A500+ ROM, A600 ROM, Kickfile, etc.)

* Verwendete WorkBench-Version

* Festplatten-Controller (sofern vorhanden)

* Modemtyp und zur Ansteuerung des Modems verwendete Hardware (Multi-IO-Karten, etc.)

Alle diese Listen können übrigens einfach mit 'RSYS' erzeugt und in eine Datei geschrieben werden.

Wenn der auftretende Fehler beschrieben wird, bitte soviel wie möglich über den Arbeitsvorgang, der dem Fehler vorausging, mit aufschreiben! Zuletzt bitte ich darum, den Text so sorgfältig zu schreiben, daß ich auch wirklich etwas damit anfangen kann; vage Beschreibungen helfen mir nicht besonders und wenn ich aus einer Fehlermeldung nicht schlau werden kann, bleibt ein möglicher Fehler vielleicht so lange im Programm, bis ich ihn zufällig selbst entdecke.

Fehlermeldungen können mir auf jedem sinnvoll erscheinende Wege zugeleitet werden, es hilft allerdings manchmal, dies über Z-Netz zu bewerkstelligen, was mir die Möglichkeit gibt, bei Bedarf noch einmal nachzufragen.

1.56 RSys12.guide/Copyrights und Verwendung

Copyrights und Verwendung

Es wird keine Garantie gegeben, daß das Programm 100%ig zuverlässig ist. Ihr benutzt dieses Programm auf eigene Gefahr. Der Autor kann auf keinen Fall für irgendwelche Schäden verantwortlich gemacht werden, die durch die Anwendung dieses Programmes entstehen.

Weiterhin darf das Programm und der Quelltext zu 'RSYS' 1.2 nicht zu kommerziellen Zwecken jeder Art verwendet werden. Desweiteren dürfen das Programm und der Quelltext nicht direkt und nicht indirekt zu militärischen Zwecken jeder Art verwendet werden! Jede Veröffentlichung in Diskmagazinen und Zeitschriften *muß* mit der Zustimmung des Autors stattfinden.

Der Quelltext ist völlig Public Domain und Giftware, aber es wäre nett, wenn mich der jeweilige Programmierer in einer kleinen Danksagung erwähnen würde. Auch wenn andere dieses Programm als Grundlage nehmen, um ein neues 'RSYS' zu programmieren, sollten diese sich mit mir in Verbindung setzen, damit wir unsere Aktivitäten koordinieren können.

Der Autor ist weiterhin sehr empfänglich für kleine Geschenke, also Geld :-), Gummibärchen, Briefe, Postkarten, Disketten, Musikkassetten u.v.a.m. Auch Geldspenden, Schecks oder Überweisungen werden nicht abgelehnt :-))
Hier meine Adresse:

Rolf Böhme
Stammestr. 48
30459 Hannover
Germany
EMAIL: R.BOEHME@COPS.ZER
FIDO: 2:245/44.2 (Rolf Boehme)

Ich behalte mir weiterhin vor, den Typ des Programmes (PD, Freeware, Shareware, Payware) zu ändern. Daher gelten diese Festlegungen nur für 'RSYS' 1.2. Alle weiteren oder vorhergehenden Releases oder Versionen unterliegen dann anderen Bestimmungen.

1.57 RSys12.guide/Danksagungen

Danksagungen

* Klaus Spark (Klaspa)

Mein erster und kritischster Betatester. Ohne ihn sähe 'RSYS' ganz anders aus! Danke Klaus!

* Klaus Nolte (Viking)

Der zweite Betatester. Seine Langzeitversuche sind die besten! Danke Klaus!

* Lars Monecke (Mave)

Er betatestete und steuerte die Icons zu 'RSYS' bei, die ich ein klein wenig modifizierte. Danke Mave!

* Michael (Pokey, Sysop der CASABLANCA)

Seinen kritischen Anmerkungen verdankt das Installer-Script sein heutiges Aussehen. Danke Michael!

* Olaf 'Olsen' Barthel

Seine Quelltexte zu 'term' und anderen sind ein wahre Fundgrube für jeden Programmierer!

* Thomas Breit, Sysop der Box ACBW in Wolfsburg

Er betatestete und fand einige Fehler.

* Florian Faber

Er betatestete und trieb mich mit seinem 4000er fast an den Rand der Aufgabe. Danke Florian!

* Franz-Josef Reichert (FJR)

Ihm verdanke ich einige Hinweise zur Systemprogrammierung. Danke, FJ!

* Michael Sinz

Sein Enforcer ist der Beste. Viele Fehlerbeseitigungen gehen auf sein Konto ':-)'

* Jan van den Baard

Seine GadToolsBox erlaubte das Design von 'RSYS'. Danke Jan!

* Carolyn Scheppner

Dein Sushi zeigte mir ohne viel Papierverschwendung, was über die Serielle an Hits kamen ':-)'

* Thomas Carstens

Sein Programm ShowGuru sagte mir, was ich schon wieder falsch gemacht habe ':-)'

Viele, viele andere Leute im Z-Netz und sonstwo,
von denen ich gelernt habe, zu programmieren.

1.58 RSys12.guide/Verwendete Software

Verwendete Software

Dieses Programm entstand mit Hilfe folgender Programme :

Aztec C68k/Amiga 5.2a Sep 25 1991 17:04:29
Copyright 1991 by Manx Software Systems, Inc.

GadToolsBox 37.176 (01.9.92)
(c) Copyright 1991,92 Jaba Development

CEdPro2 Version 2.11

(c) Copyright by CygnusSoft Software

mungwall 36.45 (9.5.91)
Copyright (c) 1990, 1991 Commodore-Amiga, Inc.

Enforcer 37.25
Copyright (c) 1992 Michael Sinz

io_torture 37.6 (16.1.91)

texinfo 2.109
Copyright (C) 1985, 86, 88, 90, 91, 92, 1993
Free Software Foundation, Inc.

MakeInfo 1.55
Copyright (C) 1993 Reinhard Spisser and Sebastiano Vigna
Under GNU License

DVI-Print - Version 1.07L for AMIGA (c) 1990/91
(c)Copyright 1990-91, (hes/rbs) , All rights reserved. Feb 9 1991

ShowDVI - Version 1.17L for AMIGA (c) 1990/91
(c)Copyright 1990-91, (hes). All rights reserved. Sep 28 1991

Da das Programm Funktionen der Kickstart 2.0 verwendet, läuft es
folglich nur auf Amigas mit Kickstart >= 2.0.

1.59 RSys12.guide/History

History

=====
Version 1.01 (unveröffentlicht)
=====

10-Apr-93

- BACKPATTERNCOL = 0, dann wurde das Fenster nach einem ZipWindow nicht korrekt aufgebaut. Bug fixed.
- Iconname (Programmname) global gemacht mit Notify auf die Icon-Datei.

18-Apr-93

- CenterWindow() erheblich vereinfacht
- Aus "Available fonts" wurde "Fonts in FONTS:", da die ROM-Fonts schon mit dem Button "Fonts" angezeigt werden können.
- Aus "Quit Sys" wurde "Quit RSys"

22-Apr-93

- Library offsets eingeführt, checkt die gepatchten Libraries und Devices

=====

Version 1.02 (unveröffentlicht)

=====

01-May-93

- Das sprintf() bricht kein Forbid()! Daher schreibe ich alle Funktionen um.

02-May-93

- Die ASL-Routinen wurden gekürzt
- Das Einsortieren der Listeneinträge verursachte unter bestimmten Umständen Enforcer-Hits. Diese wurden beseitigt.
- Die Anzeige des DOS-Typs bei Volumes sah etwas wild aus. Beseitigt.
- Die Fontanpassung wurde etwas verbessert. Bei kleinen System-Screen-Font wurde bei einigen Anzeigen im ListView das letzte Zeichen verschluckt.

06-May-93

- Abspeichern ins Clipboard ist jetzt möglich. Eine ganze Liste kann im Clipboard abgelegt werden

07-May-93

- Ein neuer Menüpunkt ist dazugekommen: Flushing. Mit ihm können entweder alle unbenutzten Libraries oder Fonts aus dem System entfernt werden

08-May-93

- Die Source wurde ausführlicher dokumentiert. Jede Quelldatei hat nun ihren Header.
- Die Port-Liste wurde nicht sortiert. Bug fixed.
- Der eingebaute Font ibm.font wurde entfernt und durch ein OpenDiskFont()-Call ersetzt. Wird der ibm.font nicht gefunden, wird Standard-Topaz für die Baumstruktur verwendet.
- Das ListView wird Read Only, wenn ein Hunk-Listing erstellt wird.

10-May-93

- Es gibt ein neues Menü "Clipboard". Es kann dort die ganze Liste, oder ein Eintrag aus dieser Liste ins Clipboard kopiert werden.
- Ein Enforcer-Hit wurde beseitigt. Die Routine GetNode() funktioniert nun korrekt.

13-May-93

- Einige Requester wurden geändert
- Der Menüpunkt "Check Libraries" wurde ins Menü "Special Actions" verlagert.

16-May-93

- Das globale Steuerfeld bekommt ein Bit mehr. Darin wird verzeichnet, ob eine Zählstatistik im unteren Textgadget angezeigt werden soll.

20-May-93

- Bug im Menüpunkt "Speak" beseitigt. Nach einem "Topaz font" wurde dieser Menüpunkt nicht korrekt gesetzt.
- Es gibt jetzt einen Modus, in dem alle Ja/Nein-Abfragen von RSys default gesetzt werden. Damit wird die Bedienung etwas schneller, falls es gewünscht wird. Deshalb gibts dann auch das neue Tooltype NOSAVEASKING.
- Der Quelltext wurde etwas überarbeitet.
- Die Menüs wurden überarbeitet.

20-May-93

- Der Menüpunkt "Volume locks" ist dazugekommen.
- Bug in der Memory-Anzeige gekillt.
- Schweren Bug in der Jump()-Routine beseitigt!

07-Jun-93

- Jedes Fenster hat jetzt ein Pattern-Muster, wie das Hauptfenster.
- Die "Check Libraries-Routine funktionierte nicht mit 4000er-ROMs. Jetzt wird der korrekte Offset des KickROMs berechnet.

12-Jun-93

- Das APP-Icon kann jetzt abgestellt werden.
- Das Centering funktionierte mal wieder nicht. Behoben.
- Das Patternmuster der Fenster kann jetzt zweifarbig dargestellt werden.

21-Jun-93

- Es ist ein Textformatter für die Ausgaben auf Datei hinzugekommen.
- Die Sourcedateien wurden überarbeitet und weiter aufgeteilt
- Der MakeFile sieht etwas anders aus
- Das Installations-Script wurde vollständig neu geschrieben

24-Jun-93

- Riesenbug in der "Memory"-Anzeige beseitigt. Führte auf dem 2000er zu keinerlei Abstürzen, aber unter OS3 zu Chaos!
 - Das Fehlerbehandlungssystem wurde total überarbeitet. Und schon wieder
-

1.5 KB kürzer ;-).

- Alle Libraries werden mit der Versionsnummer 36 geöffnet. Damit werden "falsche" Library-Aufrufe vermieden.

25-Jun-93

- Die Speicherverwaltung wurde etwas optimiert
- Ich habe endlich einen einigermaßen sicheren Weg gefunden, uninitialisierte Pointer auf Node-Namen korrekt abzufangen. Glücklicherweise trägt das System (manchmal) ein NT_UNKNOWN als Type des Nodes ein. Geschieht das, ist dieser Node sowieso Müll.
- Die Labels der Gadgets sind bei einer unglücklichen Wahl des Patternmusters und der Farben etwas "zerlegt". Bug fixed.
- Die Funktion, die den Namen eines Nodes in einen String kopiert, wurde jetzt vom Node-Typen abhängig gestaltet. Damit hoffe ich einige Enforcerhits, die sich bei unsauber eingerichteten Knotennamen ergaben den Garaus gemacht zu haben.
- Das Fenster zum Einstellen einer Taskpriorität bekam kein Pattern. Bug fixed.
- Gewaltiger Bug in der "Save Windows"-Routine entfernt!

=====

Version 1.2

=====

30-Jun-93

- Die Dokument-Dateien wurden mit TexInfo bearbeitet und neu gesetzt.

03-Jul-93

- Die Menü-Leiste wurde gekürzt und das Menü etwas anders aufgeteilt

07-Jul-93

- Ein neuer Menüpunkt - Gadgets - ist hinzugekommen.
- Die Quelltexte wurden überarbeitet
- Das Icon wird jetzt vom Programmicon übernommen. Damit ist das eingebaute Icon überflüssig und RSys wieder kleiner geworden

10-Jul-93

- Noch einen Bug in der Menü-Iconify-Routine ausgebügelt
- Die Gadget-Texte haben einen Rahmen bekommen, eine Bevelbox
- Das Layout der einzelnen Fenster wurde etwas "verschönert"
- Ein neues Font-Anzeigefenster wurde programmiert

15-Jul-93

- Ein Statusbalken ist dazugekommen und ein neues Flag, "Working bar"
 - Fehler in der Preference-Liste entdeckt! Harter Bug fixed.
-

- Dokumentation angepaßt

1.60 RSys12.guide/Anhang

Anhang

Funktionen

=====

<ohne Namen>
 About...
 Amigaguide XRef
 Assigns
 Auto front window
 BACKGROUNDCOL
 BACKPATTERN
 BACKPATTERNCOL
 Blank between texts
 Blitter Wait Queue
 BLOCKPEN
 Case sense/No case sense
 Check Libraries
 CLI processes
 Clip entry...
 Clip list
 Commodities
 CopyMem()
 CopyMemQuick()
 CRC Check
 Crypt
 CX_CRC
 CX_CRYPT
 CX_CRYPT
 CX_FINDFILE
 CX_FINDFILE
 CX_FORMAT
 CX_FORMAT
 CX_HOTKEY
 CX_HOTKEY
 CX_HUNKLIST
 CX_SAVEWINDOW
 DETAILPEN
 Devices
 Devices
 Dirs + Files/Files/Directories
 Disable()/Enable()
 Disk label
 Display Modes
 ENV-Variables
 Fast Mode
 Find File

Actions
 Project
 Other Libraries
 Assign-Liste der DOSBase
 Flags
 Aufruf von der WorkBench
 Aufruf von der WorkBench
 Aufruf von der WorkBench
 Der Text-Formatter
 Graphics
 Aufruf von der WorkBench
 Actions
 Actions
 Exec
 Clip
 Clip
 Other Libraries
 RSys und Hintergrundinfos
 RSys und Hintergrundinfos
 Actions
 Actions
 Aufruf von der WorkBench
 Die Hotkey-Aktivierung
 Aufruf von der WorkBench
 Die Hotkey-Aktivierung
 Aufruf von der WorkBench
 Die Hotkey-Aktivierung
 Aufruf von der WorkBench
 Die Hotkey-Aktivierung
 Aufruf von der WorkBench
 Aufruf von der WorkBench
 Aufruf von der WorkBench
 Actions
 Actions
 Actions
 RSys und Hintergrundinfos
 Actions
 Graphics
 Preferences
 Flags
 Actions

Flush all libraries and Fonts
 Flush Fonts
 Flush Libraries
 Fonts
 Fonts in FONTS:
 Forbid() / Permit()
 Format
 Format
 Format Disk
 Found files/dirs
 Gadgets
 HandlerInp
 Hardware
 Help
 Iconify
 Icons/No Icons
 IFF File
 International/No international
 Interrupts
 Jump
 Kill saved list
 Libraries
 List Hunks
 LockIBase() / UnlockIBase()
 Memory
 MENUPEN
 Mouse window
 MOUSEWINDOW
 No header in file
 NOAPPICON
 NOFASTMODE
 NOSORT
 OFS/FFS
 OUTWINDOW
 OUTWINDOW
 Pattern
 Physical Devices
 Ports
 Postfix
 Preferences
 Prefix
 Printer Data
 Public Screens
 Quick
 Quit RSYS...
 Quote-in text
 RawDoFmt()
 Reload actionfile
 Rescan windows
 Resident programs
 Ressourcen
 Restore saved list
 ROT 13-Verfahren
 Save asking
 Save current list
 Save found list
 Save List

Flushing
 Flushing
 Flushing
 Font-Liste der GfxBase
 Diskfont
 RSys und Hintergrundinfos
 Actions
 Actions
 Actions
 Actions
 Intuition
 Inputhandler-Liste
 Hardware-Liste der ExpansionBase
 Project
 Project
 Actions
 Actions
 Actions
 Interrupt-Liste der ExecBase
 Springen zwischen Public Screens
 Internal List
 Library-Liste der ExecBase
 Actions
 RSys und Hintergrundinfos
 Memory-Liste in der ExecBase
 Aufruf von der WorkBench
 Flags
 Aufruf von der WorkBench
 Der Text-Formatter
 Aufruf von der WorkBench
 Aufruf von der WorkBench
 Aufruf von der WorkBench
 Actions
 Die Hotkey-Aktivierung
 Aufruf von der WorkBench
 Actions
 DOS
 Port-Liste in der ExecBase
 Der Text-Formatter
 Preferences
 Der Text-Formatter
 Preferences
 Intuition
 Actions
 Project
 Der Text-Formatter
 RSys und Hintergrundinfos
 Project
 Actions
 Exec
 Ressourcen-Liste der ExecBase
 Internal List
 Verschlüsseln von Daten und Dateien
 Flags
 Internal List
 Actions
 Abspeichern einer Liste

Save windows
 Screens
 Sel all
 Sel none
 Selected Devices
 Semaphores
 Software Devices
 Sorting
 Speaking
 sprintf()
 Start/Stop Search
 strcpy()
 strlen()
 strncpy()
 Systeminfo
 Tasks
 Text to save
 TOF Wait Queue
 Topaz font
 TOPAZFONT
 Vectors
 Volume Locks
 Volumes
 Windows
 Windows
 Working bar
 WORKINGBAR
 zellularen Automaten

Actions
 Screen-Liste der IntuitionBase
 Actions
 Actions
 Actions
 Exec
 Exec
 Flags
 Flags
 RSys und Hintergrundinfos
 Actions
 RSys und Hintergrundinfos
 RSys und Hintergrundinfos
 RSys und Hintergrundinfos
 Allgemeine Systeminfo
 Task-Liste der ExecBase
 Der Text-Formatter
 Graphics
 Flags
 Aufruf von der WorkBench
 Exec
 DOS
 Volume-Liste der DOSBase
 Actions
 Window-Liste der IntuitionBase
 Flags
 Aufruf von der WorkBench
 Verschlüsseln von Daten und Dateien

Programm Index

=====

Check Libraries
 Gadget: Assigns
 Gadget: Fonts
 Gadget: HandlerInp
 Gadget: Hardware
 Gadget: Interrupts
 Gadget: Jump
 Gadget: Libraries
 Gadget: Memory
 Gadget: Ports
 Gadget: Ressourcen
 Gadget: Save List
 Gadget: Screens
 Gadget: Systeminfo
 Gadget: Tasks
 Gadget: Volumes
 Gadget: Windows
 Menüpunkt: About...
 Menüpunkt: Amigaguide XRef
 Menüpunkt: Auto front window
 Menüpunkt: Blitter Wait Queue
 Menüpunkt: CLI processes
 Menüpunkt: Clip entry...

Actions
 Assign-Liste der DOSBase
 Font-Liste der GfxBase
 Inputhandler-Liste
 Hardware-Liste der ExpansionBase
 Interrupt-Liste der ExecBase
 Springen zwischen Public Screens
 Library-Liste der ExecBase
 Memory-Liste in der ExecBase
 Port-Liste in der ExecBase
 Ressourcen-Liste der ExecBase
 Abspeichern einer Liste
 Screen-Liste der IntuitionBase
 Allgemeine Systeminfo
 Task-Liste der ExecBase
 Volume-Liste der DOSBase
 Window-Liste der IntuitionBase
 Project
 Other Libraries
 Flags
 Graphics
 Exec
 Clip

Menüpunkt: Clip list	Clip
Menüpunkt: Commodities	Other Libraries
Menüpunkt: CRC Check	Actions
Menüpunkt: Crypt	Actions
Menüpunkt: Display Modes	Graphics
Menüpunkt: ENV-Variables	Preferences
Menüpunkt: Fast Mode	Flags
Menüpunkt: Find File	Actions
Menüpunkt: Flush all libraries and Fonts	Flushing
Menüpunkt: Flush Fonts	Flushing
Menüpunkt: Flush Libraries	Flushing
Menüpunkt: Fonts in FONTS:	Diskfont
Menüpunkt: Format Disk	Actions
Menüpunkt: Gadgets	Intuition
Menüpunkt: Help	Project
Menüpunkt: Iconify	Project
Menüpunkt: Kill saved list	Internal List
Menüpunkt: List Hunks	Actions
Menüpunkt: Mouse window	Flags
Menüpunkt: Physical Devices	DOS
Menüpunkt: Preferences	Preferences
Menüpunkt: Printer Data	Preferences
Menüpunkt: Public Screens	Intuition
Menüpunkt: Quit RSys...	Project
Menüpunkt: Reload actionfile	Project
Menüpunkt: Resident programs	Exec
Menüpunkt: Restore saved list	Internal List
Menüpunkt: Save asking	Flags
Menüpunkt: Save current list	Internal List
Menüpunkt: Save windows	Actions
Menüpunkt: Semaphores	Exec
Menüpunkt: Software Devices	Exec
Menüpunkt: Sorting	Flags
Menüpunkt: Speaking	Flags
Menüpunkt: TOF Wait Queue	Graphics
Menüpunkt: Topaz font	Flags
Menüpunkt: Vectors	Exec
Menüpunkt: Volume Locks	DOS
Menüpunkt: Working bar	Flags
tbreak	Tips und Tricks
tracer	Tips und Tricks
Xoper, ARTM, TaskX und andere	Was ist RSys?

Konzept

=====

Abspeichern einer Liste	Abspeichern einer Liste
Adresse	Copyrights und Verwendung
Allgemeine Systeminfo	Allgemeine Systeminfo
Anhang	Anhang
Assign-Liste der DOSBase	Assign-Liste der DOSBase
Aufruf vom CLI	Aufruf vom CLI
Aufruf von der WorkBench	Aufruf von der WorkBench
Copyrights und Verwendung	Copyrights und Verwendung
CRC-Checksummen berechnen	CRC-Checksummen berechnen
Danksagungen	Danksagungen
Das Actioning	Das Actioning

Das Hauptfenster
Das Icon
Das Menü
Das Modifizieren
Dateien suchen
Der Text-Formatter
Die Actions-Datei
Die ENV-Variablen
Die Gadgets und das ListView
Die Help-Datei
Die Hotkey-Aktivierung
Die Utilities von RSys
Diskfont
DOS
Exec
Fehlermeldungen
Fenster in IFF-Dateien speichern
Font-Liste der GfxBase
Gadgets - Menüs - das Icon
Graphics
Hardware-Liste der ExpansionBase
History
Inputhandler-Liste
Installation von RSys
Interrupt-Liste der ExecBase
Intuition
Library-Liste der ExecBase
Memory-Liste in der ExecBase
Menü Other Lists
Menü: Actions
Menü: Clip
Menü: Flags
Menü: Flushing
Menü: Internal List
Menü: Project
normaler CLI-Befehl
Other Libraries
Paßwort
Port-Liste in der ExecBase
Preferences
Ressourcen-Liste der ExecBase
RSys und Hintergrundinfos
Schutzprotokolle
Screen-Liste der IntuitionBase
Springen zwischen Public Screens
systemkonform
Task-Liste der ExecBase
Tips und Tricks
Verschlüsseln von Daten und Dateien
Verwendete Software
Volume-Liste der DOSBase
Was ist RSys?
wbstartup-Verzeichnis
Weitere Pläne
Window-Liste der IntuitionBase

Das Hauptfenster
Das Icon
Das Menü
Das Modifizieren
Dateien suchen
Der Text-Formatter
Die Actions-Datei
Die ENV-Variablen
Die Gadgets und das ListView
Die Help-Datei
Die Hotkey-Aktivierung
Die Utilities von RSys
Diskfont
DOS
Exec
Weitere Pläne
Fenster in IFF-Dateien speichern
Font-Liste der GfxBase
Gadgets - Menüs - das Icon
Graphics
Hardware-Liste der ExpansionBase
History
Inputhandler-Liste
Installation von RSys
Interrupt-Liste der ExecBase
Intuition
Library-Liste der ExecBase
Memory-Liste in der ExecBase
Other Lists
Actions
Clip
Flags
Flushing
Internal List
Project
Aufruf vom CLI
Other Libraries
Verschlüsseln von Daten und Dateien
Port-Liste in der ExecBase
Preferences
Ressourcen-Liste der ExecBase
RSys und Hintergrundinfos
RSys und Hintergrundinfos
Screen-Liste der IntuitionBase
Springen zwischen Public Screens
RSys und Hintergrundinfos
Task-Liste der ExecBase
Tips und Tricks
Verschlüsseln von Daten und Dateien
Verwendete Software
Volume-Liste der DOSBase
Was ist RSys?
Installation von RSys
Weitere Pläne
Window-Liste der IntuitionBase