

ARTM

COLLABORATORS

	<i>TITLE :</i> ARTM		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 23, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ARTM	1
1.1	ARTM Anleitung	1
1.2	Copyright und andere rechtliche Dinge	1
1.3	shareware	2
1.4	Adresse des Authors	2
1.5	installation	3
1.6	programmstart	3
1.7	tastaturbelegung	4
1.8	hilfe	5
1.9	funktionen	5
1.10	aktionen	5
1.11	tasks	6
1.12	windows	6
1.13	libraries	7
1.14	devices	7
1.15	resources	8
1.16	ports	8
1.17	residents	8
1.18	interrupts	9
1.19	vectors	9
1.20	memory	10
1.21	mount	10
1.22	assign	11
1.23	locks	11
1.24	fonts	11
1.25	hardware	12
1.26	res_cmds	13
1.27	semaphoren	13
1.28	monitor	13
1.29	last alert	14

1.30	zz	14
1.31	i	14
1.32	disksymbol	15
1.33	priority	15
1.34	remove	15
1.35	close	15
1.36	relase	15
1.37	clear	16
1.38	freeze	16
1.39	unlock	16
1.40	more	16
1.41	break	17
1.42	signal	17
1.43	cancel	17
1.44	monitor_acts	18
1.45	eh	18
1.46	history	18
1.47	zukunft	19
1.48	hinweis	19
1.49	Danksagungen	19
1.50	Index	20

Chapter 1

ARTM

1.1 ARTM Anleitung

ARTM Amiga Real Time Monitor V2.04

ARTM gibt Aufschluß über viele Aktivitäten im Amiga, und ist voll oberfächengesteuert und dadurch sehr leicht zu bedienen. Außerdem können viele systeminterne Strukturen verändert werden.

Copyright Copyright und andere rechtliche Dinge
shareware shareware Gebühr
Author Die Adressen des Autors

Installation So wird ARTM installiert
Programmstart Start ARTM
Tastaturbelegung Tastaturbelegung von ARTM
Hilfe Die Hilfs_Funktion in ARTM
Funktionen Von ARTM unterstützte Funktionen
Aktionen Von ARTM ausführbare Aktionen

History Wie alles begann:
Zukunft Weiterentwicklung von ARTM
Hinweis Wichtig bitte lesen
Danksagungen unkommentiert !?!!?
Enforcer Hits Enforcer Hits halt

```

      / ' ( ) ' ) ) )      _/_
    , - / - , - _/_ / / / _ _ / _ _ _
  ( _ / o - / / o / ' ( (<_/_ ( _ (<_/_ ) _/_ ) _
      (_/_

```

1.2 Copyright und andere rechtliche Dinge

Copyright und andere rechtliche Dinge:

Copyright:

Copyright © 1990 - 1994 by D. Jansen & F.J. Mertens

ARTM ist ein Shareware Programm und darf ohne Profit kopiert werden (max. Diskettenpreis + Porto). ARTM darf nur unverändert, komplett und mit Copyright-Notiz weitergegeben werden. Weder das Programm, noch der Quelltext (oder Teile davon) dürfen ohne schriftliche Genehmigung der Autoren in kommerziellen Programmen benutzt werden.

Keine Garantie:

Es wird keine Garantie dafür übernommen, daß das Programm, 100%ig zuverlässig arbeitet. Sie benutzen es auf eigene Gefahr. Es wird keine Haftung für eventuelle Schäden durch die Benutzung von ARTM übernommen.

1.3 shareware

shareware:

Bei regelmäßiger Benutzung bitten wir um einen Shareware-Beitrag von 20.-- DM. Dafür erhalten Sie einen Update-Service ohne die lästigen Requester.

Wer das Geld überweisen will, kann dies an folgende Bankverbindung tun. Bitte geben Sie dabei immer Ihre volle Adresse an.

Kreissparkasse Wildberg
BLZ 606 510 70
KontoNr. 219190

Jeder registrierte ARTM-User erhält alle weiteren Updates kostenlos, wenn Sie mir einen ausreichend frankierten Rückumschlag und eine formatierte, fehlerfreie Diskette zuschicken.

update

Wenn Sie PGP (Pretty Good Privacy Amiga-Umsetzung von Peter Simons) benutzen und auch noch im Usenet oder Fido vertreten seit, kann der Update auch encrypted per Mail erfolgen. Dazu benötige ich eine Mail mit eurem gültigen PGP.key.

Wenn Sie mir schreiben, um Bugs, Anregungen oder Fragen loszuwerden, dann fügen Sie bitte einen ausreichend frankierten Rückumschlag bei. Wenn die Antwort dann mal etwas länger dauert, verzweifeln Sie nicht, denn ich muß nebenbei auch noch meiner Arbeit nachgehen.

1.4 Adresse des Authors

Adresse des Authors:

Franz-Josef Mertens
Sonnenrain 17
72218 Wildberg
Germany

E-Mail:

UUCP: fjm@merlin.tynet.sub.org
FIDO: (2:246/2013.9) Mentopolis Wildberg

1.5 installation

Installation automatisch:

Zur automatischen Installation existiert im Archiv ein Installer_script. Hierbei wird der C= Installer benutzt.

Installation manuell:

Die Installation läuft ganz einfach ab. Da ARTM nur aus einem Programm besteht, kopieren Sie ARTM einfach mittels copy ARTM zielpath/ in das von Ihnen gewünschte Directory. Oder wählen Sie das ARTM ICON mit der Maus an und schieben es in die gewünschte Schublade auf der Workbench. Die .doc files kopieren Sie auch mit dem "copy" Befehl in das von Ihnen ausgewählte Verzeichnis. Das ARTM.guide File soll nach LOCALE:HELP/ kopiert werden.

```
copy ARTM.guide SYS:LOCALE/HELP/ARTM.guide
```

```
assign HELP: SYS:LOCAL/HELP
```

Folgende files sind im Archiv ARTM:

```
ARTM
ARTM.info
Install_ARTM
Install_ARTM.info
german.doc/artm.guide
german.doc/artm.guide.info
german.doc/artm.dok
german.doc/artm.dok.info
german.doc/artm.history
german.doc/artm.history.info
english.doc/artm.guide
english.doc/artm.guide.info
english.doc/artm.doc
english.doc/artm.doc.info
english.doc/artm.history
english.doc/artm.history.info
```

1.6 programmstart

Programmstart von der Shell:

```
Type ARTM [opt]
```

und ARTM öffnet sein Fenster auf dem WB Screen

folgende Optionen sind möglich:

- q Startet ARTM als Icon am rechten oberen Rand des WB Screens. Ein Doppelklick darauf und ARTM erscheint in voller Größe.
- poutputpath teilt ARTM mit, wo es sein output hinschreiben soll. Möglich sind z.B. alle devices prt: ram:file.
- ix/y Startet ARTM als WB Fenster. Wenn die Funktion iconify benutzt wird, erscheint das Icon an der vorgegebenen x/y position.
- wx/y Startet ARTM als WB Fenster, x/y ist die linke obere Ecke des ARTM-Window. Sollten von x/y bis zum rechten Rand weniger als 640 Pixel sein, wird für x/y = 0/0 gewählt.
- hx Startet ARTM als WB Fenster, x=window.height. Ist Ihr Screen kleiner x, wird eine Standardhöhe von 168 Pixeln genommen.

Programmstart von der WorkBench:

Ein einfacher Doppelklick auf das ARTM.icon und ARTM öffnet sein Fenster auf dem WB Screen.

Die Optionen von oben können auch als ToolType verwendet werden.

```
START=WINDOW|ICON
ICON_POS=x/y
OUTPUT_PATH=RAM:ARTM.output
WINDOWHEIGHT=280
WINDOW_POS=70/16
```

1.7 tastaturbelegung

Die Tastaturbelegung von ARTM:

Momentan werden nur einige wenige Funktionen durch die Tastatur unterstützt. Dies wird sich aber in der nächsten Version ändern. Dann sollen alle Funktionen über Tastatur bedienbar sein.

ARTM kennt folgende Tastaturkommandos:

- ESC beendet ARTM
- HELP zeigt zu jeder Funktion die betreffende Node aus diesem Guide

(siehe dazu auch Funktion Hilfe)

ARTM Requester kennen folgende Tastaturkommandos:

ESC	bricht die Funktion ab gleich CANCEL
RETURN	führt die Funktion aus gleich OK

1.8 hilfe

Die Hilfefunktion in ARTM:

ARTM bietet eine umfassende Hilfefunktion. Es kann zu jeder Zeit die Help Taste gedrückt werden. Nun wird das ARTM-Help-File (artm.guide) geöffnet und die ausgewählte Funktion angezeigt. Voraussetzung hierfür ist allerdings, daß Ihr Amiga über die amigaguide.library in libs: verfügt, und das Help-File korrekt installiert ist. (siehe Installation)

1.9 funktionen

Folgende Funktionen sind implementiert:

Tasks	listet alle gerade laufenden Tasks
Windows	listet alle Screens und Windows
Libraries	listet alle geöffneten Libraries
Devices	listet alle geöffneten Devices
Resources	listet alle Resources
Ports	listet alle geöffneten Ports
Residents	listet alle resident eingebundenen Programme
Interrupts	listet alle Interrupts
Vectors	listet den Inhalt der wichtigsten Vektoren
Memory	listet alle eingebundenen Speicherbänke
Mount	listet alle mounted Devices
Assign	listet alle assigned Devices
Locks	listet alle Locks
Fonts	listet alle eingebundenen Fonts sowie RomFonts
Hardware	zeigt die wichtigsten Hardware-Merkmale auf
Resident Cmds	listet alle residenten Programme
Semaphoren	listet alle eingerichteten Semaphoren
Monitor	ein kleiner Speicher-Monitor readonly
Last Alert	zeigt die letzte Alert Nr. und Adresse an
zz	Das ARTM-Window wird iconified
I	zeigt den AboutRequester
Disksymbol	Inhalt der Funktion wird gespooled

1.10 aktionen

Folgende Aktionen sind möglich:

Remove	kann Einträge wie Tasks usw. entfernen
Close	kann Windows etc. schließen
Release	zeigt Semaphoren an
Clear	cleared Vektoren
Freeze	legt laufende Tasks auf Eis
Activate	erweckt eingefrorene Tasks zum Leben
Priority	kann Prioritäten von Tasks etc. verändern
Unlock	gibt gesperrte Files Icons etc. frei
More	zeigt die Struktur zu angewählten Funktionen
Break	sendet CTRL C/D/E/F an einen Task
Signal	sendet eine einzugebene Sequenze an einen Task
Address	sucht Adresse im Monitor
Forward	blättert den Monitor eine Seite vor
Backward	blättert den Monitor eine Seite zurück
Search	sucht nach ASCII String im Monitor
Cancel	schaltet die untere Gadget-Reihe um

1.11 tasks

Tasks:

ist eingeschaltet wenn ARTM gestartet wird. Es werden alle Task's aufgelistet, die sich im Amiga befinden. Durch Mausklick auf einen beliebigen Task wird diese selektiert.

Folgende Aktionen sind möglich:

Remove, Priority, Freeze, Activate, More, Break und Signal

Folgende Informationen werden angezeigt:

Address, Priority, Signals, State, Code, Type, Name.

zum Beispiel:

Address	Pri	Stack	Signals	State	Code	Type	Name
003f96f0	0	4094	00000010	run	ET--	process	ARTM
002f1cc0	0	2998	80000000	wait	ET--	process	« ConClip »

1.12 windows

Windows:

listet alle geöffneten Screens und Windows. Auch hier kann durch Mausklick ein Screen oder ein Window selektiert und mit Close aus dem Amiga entfernt werden. Mit More werden Zusatzinformationen angezeigt, wie min, max Width und Hight. Achtung Windows sollte man nur schließen, wenn auch der zugehörige Task tot ist. Sonst ist der ALERT gewiß. Das gleiche gilt für Windows mit geshartem IDCMP-Port.

Folgende Informationen werden angezeigt:

Address, Position, Size, Flags, IDCMP (nur bei Windows), Title.

zum Beispiel:

Address	Pos	Size	Flags	IDCMP	Title
002c1428	0,0	708,512	00004611		Workbench Screen
003da88c	68,16	640,283	2800300f	00400670	AMIGA RealTime Monitor V2.0 Beta

1.13 libraries

Libraries:

zeigt alle eingebundenen Libraries im Amiga. Wurde eine Library angewählt, kann mit der Funktion Priority die Priorität und mit Close die Library geschlossen werden (der OpenCount wird auf 0 gesetzt). Libraries die selbst eingebunden wurden, können auch mit Remove entfernt werden, wenn der OpenCount = 0 ist.

Folgende Informationen werden angezeigt:

Address, Type, Priority, OpenCount, Version (setzt sich zusammen aus Version & Revision), Negsize, Possize, Name.

zum Beispiel:

Address	Type	Pri	OpenC	Version	Neg	Pos	Name
002801b0	library	0	25	40.1	296	48	utility.library
00283d74	library	0	1	40.24	1076	544	graphics.library

1.14 devices

Devices:

listet alle Devices auf. Mit Remove können Devices entfernt werden, sofern diese geschlossen sind. Priority kann die Priorität wie bei Libraries verändern.

Folgende Informationen werden angezeigt:

Address, Type, Priority, OpenCount, Version (setzt sich zusammen aus Version & Revision), Negsize, Possize, Name.

zum Beispiel:

Address	Type	Pri	OpenC	Version	Neg	Pos	Name
00286e74	device	0	1	40.1	36	338	gameport.device
0028700c	device	0	19	39.4	68	380	timer.device

1.15 resources

Resources:

listet alle Resources. Mit der Funktion Priority kann die Priorität geändert werden und mit Remove eine vorher selektierte Resource entfernt werden. Eine kleine Unstimmigkeit tritt in Zusammenhang mit den ALF 2.0 Ressourcen auf. Bei einem Opencount von 21000 und der Version 14021.0, schiebe ich diesen Bug der Programmierung von ALF2.0 zu.

Folgende Informationen werden angezeigt:

Address, Type, Priority, OpenCount, Version (setzt sich zusammen aus Version & Revision), Negsize, Possize, Name.

zum Beispiel:

Address	Type	Pri	OpenC	Version	Neg	Pos	Name
00280254	resource	0	0	37.0	20	42	potgo.resource
002802b8	resource	0	0	0.0	24	136	ciaa.resource

1.16 ports

Ports:

zeigt alle Ports mit den Namen der zugehörigen Tasks. Ports können mit Remove entfernt werden, aber Vorsicht: ist angebracht. Sollte ein Port, einer noch laufenden Task, geschlossen werden, bekommen Sie einen ALERT.

Folgende Informationen werden angezeigt:

Address, PortName, Type, SigBit, Flags, TaskName.

zum Beispiel:

Address	Pri	Portname	Type	SigBit	Flags	TaskName
00300d50	0	fBlanker	msgport	30	signal	fBlanker
0032b864	0	REXX	msgport	31	signal	RexxMaster

1.17 residents

Residents:

zeigt alle Residents im Amiga. Zusätzlich werden auch die Residents gezeigt, die durch das Verbiegen des Pointers KickTagPtr zusätzlich zur Verfügung stehen. (z.b. ein Virus oder die RAD:) Dabei werden Residents, die nur in der KickTagPtr-Tabelle stehen color 3 und solche, die nach einem Reboot auch in der ResmodulesTabelle stehen schwarz angezeigt.

Folgende Informationen werden angezeigt:

Address, Type, Priority, Version, Flags, Name.

zum Beispiel:

Address	Type	Pri	Vers	Flags	Name
002036d0	library	110	40	00000002	expansion.library
002000b6	library	105	40	00000002	exec.library

1.18 interrupts

Interrupts:

listet alle Interrupts im Amiga auf. Interrupts die durch Interrupt-Server verwaltet werden, können mit Remove entfernt werden.

Folgende Informationen werden angezeigt:

Address, Data, Code, Priority, Type, Nr., InterruptName, Name.

zum Beispiel:

Address	Data	Code	Pri	Type	Ctrl	Nr	IntrName	Name
00283882	00283834	0021507c	0	unknown	hand	1	Disk Block	disk.resource
002802e2	002802b8	0020e6ba	120	interrupt	serv	3	CIA-A	ciaa.resource

1.19 vectors

Vectors:

zeigt die wichtigsten SystemVektoren des Amigas. ColdCapture, CoolCapture, WarmCapture, KickMenPtr, KickTagPtr, KickChecksum. Mit dem Gadget Clear können ColdCapture, CoolCapture und WarmCapture auf NULL gesetzt werden. Wird KickMemPtr, KickTagPtr oder KickChecksum vorher selektiert, werden alle drei auf einmal auf NULL gesetzt. Der KickTagPtr bietet noch eine Besonderheit. Wird in der Residentliste, auf die er zeigt, ein Eintrag gefunden, wird dessen Node (Address) und Name hinter dem KickTagPtr gezeigt. Sind mehrere Einträge vorhanden, wird momentan nur der erste angezeigt. Will man alle sehen, so ist die Funktion Residents aufzurufen. Die Einträge der KickTagPtr-Tabelle erscheinen farbig (siehe oben).

Folgende Informationen werden angezeigt:

Address, VectorName und Content, sollte KickTagPtr auf eine eigene Residenttabelle zeigen, wird Address & Name dieser Residents mitangezeigt.

zum Beispiel:

Address	Content	Name	Address	Name
---------	---------	------	---------	------

```
-----
0028083a  00000000  ColdCapture
0028083e  00000000  CoolCapture
00280842  00000000  WarmCapture
00280a32  00000000  KickMemPtr
00280a36  00000000  KickTagPtr
00280a3a  00000000  KickChecksum
```

1.20 memory

Memory:

zeigt die Speicheraufteilung in chip, fast und expansion Ram an. Mit Funktion Priority kann die Priorität geändert werden und More zeigt die Größe und Lage der einzelnen Speicherfragmente.

Folgende Informationen werden angezeigt:

Address, Priority, Attribute, Lower, Upper, Maximum, in Use, Free, Largest, Name und ganz unten die Totalwerte

zum Beispiel:

```
-----
Address  Pri Lower      Upper    Maximum In-Use   Free Largest Name
-----
00280000    0 00280020 00600000  3583k  1692k  1891k   1853k expansion mem
00000400  -10 00000420 00100000  1022k   237k   785k    763k chip memory
```

1.21 mount

Mount:

listet alle mit mount angemeldeten Devices im Amiga auf.

Folgende Informationen werden angezeigt:

Address, Device, Heads, Sectors, Buffer, Cylinder, Interleavefaktor, Handlername. Bei Speichereinheiten zeigt ARTM zusätzlich Size, Full, rw (read/write), Volume (name) an.

zum Beispiel:

```
-----
Address  Device Hd Sec Buff  Cyl I HandlerName      Size Full RW Volume
-----
002d080c  PIP:
002c03d4  CNC:
002bb40c  PIPE:                      L:Queue-Handle
00282c5c  RAM:                      304  100% rw Ram Disk
00282c1c  CON:
00282bdc  RAW:
00282b9c  SER:                      L:port-handler
00282b5c  PAR:                      L:port-handler
```

```

00282b2c PRT:                L:port-handler
002915b0 DH0:      4  42   30  586 0 gvpscsi.device  48   98% rw Amiga
00291438 DF0:      2  22   20   80 0 trackdisk.devi  1   98% rw PSFonts_abd
00291688 DH1:      4  42   30  366 0 gvpscsi.device  30   94% rw Platte
00291748 DH2:      4  42   32   74 0 gvpscsi.device   6   98% rw Mail
002917f8 DH3:      4  42   32  196 0 gvpscsi.device  16   87% rw dh3
002918b8 DH5:      2  42   32 1215 0 gvpscsi.device  49   94% rw dh5
00291378 DF1:      2  11    5   80 0 trackdisk.devi      no disk present

```

1.22 assign

Assign:

zeigt alle assignten Pfade und Devices an.

Folgende Informationen werden angezeigt:

Address, Directory, Path.

zum Beispiel:

Address	Directory	Path
002bebc4	plz:	Amiga:plz
002be9ec	Documentum:	Amiga:Arbeit

1.23 locks

Locks:

listet alle Locks (read und write). Wird ein Lock angewählt, kann er mit Unlock entfernt werden.

Folgende Informationen werden angezeigt:

Address, Access, Path.

zum Beispiel:

Address	Access	Path
003eec1c	read	Ram Disk:
002ba6bc	read	Ram Disk:ENV
00365758	read	dh5:

1.24 fonts

Fonts:

zeigt alle geladenen Fonts, auch die im KickRom. Mit Remove wird der

selektierte Font aus der Liste entfernt. Gibt das benutzende Programm den Font frei, (Count auf NULL) wird auch der dafür belegte Speicher freigegeben.

Folgende Informationen werden angezeigt:

Address, Count, YSize, XSize, Style, Flgs, LoChar, HiChar und Name.

zum Beispiel:

Address	Cnt	Y	X	Style	Flags	LoChar	HiChar	Type	Name
00286cf8	25	8	8	00	41	32	255	ROM	topaz.font
00286ca8	3	9	10	08	49	32	255	ROM	topaz.font
002df3ea	9	8	8	00	42	32	255	disk	pearl.font

1.25 hardware

Hardware:

listet alle installierten "autoconfig-boards" (ExpansionBase) auf. Außerdem wird angegeben welcher Prozessor und Mathprozessor eingebaut sind. Ferner werden auch die CustomChips gelistet und zwar wird unterschieden ob Fat Agnus oder Fat Denise (PAL oder NTSC) oder die normalen Agnus und Denise oder das neue AA Chipset eingebaut sind.

Folgende Informationen werden angezeigt:

Address, Boardadr, Boardsize, Slotadr, Slotsize, Driver, Product, Type, Manufacturer und Board.

zum Beispiel meine Hardware:

Address	Boardadr	Bo.Size	Slotadr	Ssize	Driver	Prod	Type	Manuf	Boardname
00000420	00200000	2048kB	00000020	32	no	81	230	514	A2630
00000468	00e90000	64kB	000000e9	1	yes	11	209	2017	GVP-II HD
000004b0	00400000	2048kB	00000040	32	no	10	230	2017	GVPMemory

```

Processor ..... 68030
Math CoProcessor..... 68882
MMU ..... 68030 (OFF)
DMA CustomChip ..... ECS Agnus (PAL)
Graphic CustomChip ..... Normal Denise
CPU Instruction Cache ..... ON
CPU Instruction Burst ..... ON
CPU Data Cache ..... ON
CPU Data Burst ..... ON
Kickstart Rom ..... 2.04
Rom Kickstart Version ..... 37.175
Used Kickstart Version ..... 37.175
Workbench Version ..... 38.12

```


1.26 res_cmds

Resident Cnds:

zeigt alle residenten Befehle im Rom intern und auch die, die mit resident Programm pure resident gemacht werden. Bei den letzteren werden alle Datensegmente mitangezeigt.

Folgende Informationen werden angezeigt:

Name, ROM INTERN. Bei selbst resident gemachten Programmen werden Lower, Upper und Size der DatenSegmente mitangezeigt.

zum Beispiel:

Name

ls

Hunks:

Lower	Upper	Size
004351c0	00438324	12652
004046b0	0040530c	3172
total:		15824

Alias	ROM INTERN
Ask	ROM INTERN
CD	ROM INTERN

1.27 semaphoren

Semaphoren:

zeigt alle Semaphoren vom AmigaDos an. Mit Remove wird die vorher selektierte Semaphore erst angezeigt und dann entfernt. Mit Release wird die vorher selektierte Semaphore angezeigt.

Folgende Informationen werden angezeigt:

Address, Priority, NestCount, WaitQueue, Name, OwnerTask.

zum Beispiel:

Address	Pri	Nest	Queue	Name	OwnerTask
002d775c	0	0	-1	« IPrefs »	NO_OWNER_TASK
0038ae74	-100	0	-1	Alert Patch	NO_OWNER_TASK

1.28 monitor

Monitor:

ist ein kleiner Speichermonitor, der es erlaubt alle Speicherstellen zu begutachten. Es werden immer 100 Zeilen angezeigt, die gescrollt werden können. Mit Forward und Backward können die nächsten oder die vorherigen 100 records angezeigt werden. Bei Address erscheint ein kleiner Requester, in der eine Adresse angegeben werden kann. Mit Search kann auch einem ASCII String gesucht werden.

Folgende Informationen werden angezeigt:

Address, Bytes (Hex), Bytes (ASCII)

zum Beispiel:

Address	Bytes (Hex)	Bytes (ASCII)
00200000	11 16 4e f9 00 20 00 d2	. . N ù . . Ò
00200008	00 00 ff ff 00 28 00 3e	. . ÿ ÿ . (. >
00200010	00 28 00 0a ff ff ff ff	. (. . ÿ ÿ ÿ ÿ
00200018	00 41 4d 49 47 41 20 52	. A M I G A R
00200020	4f 4d 20 4f 70 65 72 61	O M O p e r a
00200028	74 69 6e 67 20 53 79 73	t i n g S y s
00200030	74 65 6d 20 61 6e 64 20	t e m a n d

Enforcer Hits

Wird per absoluter Adresse auf einen Speicherbereich gesprungen der nicht existiert treten Enforcer Hits auf. Dies ließe sich zwar abfangen, würde aber einen sehr hohen Aufwand erfordern.

1.29 last alert

Last Alert:

zeigt die letzte Alertnummer mit Adresse an.

Enforcer Hits

Hier treten leider 2 Enforcer Hits auf. Diese lassen sich aber nicht vermeiden, da ich sonst nicht an die Daten komme.

1.30 zz

zz:

ist eine iconify Funktion. Wird dieses Gadget angewählt, verschwindet das Window und wird zum Icon. Ein doppelter Mausklick auf dieses Icon öffnet wieder das ARTM-Window.

1.31 i

I:

zeigt den AboutRequester

1.32 disksymbol

Disksymbol:

speichert alle Punkte der gerade ausgewählten Funktion in einen file oder gibt die Informationen auf ein Device aus z.B. PRT: siehe hierzu Option -p

1.33 priority

Priority:

verändert die Priorität der vorher angewählten Task, Library, Device, Ports, Interrupts, Memory oder Resource über einen Requester. Der Wert kann über Tastatur verändert werden. Die Priority ist nur im Rahmen von -128 bis 127 veränderbar. Werte von -10 bis 10 sollten nicht überbeziehungsweise unterschritten werden, da sonst bei Werten über 10, wichtige Prozesse ausgebremst werden. Bei Werten unter -10 zeigt es keine Wirkung da in der Regel kein Programm ein Priorität von weniger als -10 hat.

1.34 remove

Remove:

entfernt vorher selektierte Task, Library, Device, Ports, Fonts, Resources, Semaphoren und Interrupts aus dem System. Selektiert wird mit einfachem Mausklick auf den entsprechenden Eintrag.

1.35 close

Close:

schließt Windows, Screens und Libraries und listet anschließend alle restlichen Screens, Windows und Libraries. Vorgehensweise wie bei Remove.

1.36 relase

Relase:

relased Semaphoren aus der Liste SignalSemaphore.

1.37 clear

Clear:

setzt einen vorher selektierten SystemVektore aus Funktion Vectors auf NULL. Sollte einer dieser Vektoren nicht gleich NULL sein, könnte sich ein ↩
residenter Virus eingeschlichen haben. Mit Clear wird ihm die Fähigkeit genommen einen Reset zu überleben. Ein eventueller Virus wird damit nicht entfernt. Es können aber auch nützliche Programme sein, die sich resident-fähig einbinden z.B. die Rad:

1.38 freeze

Freeze:

hält einen laufenden Task an. Entfernt den Task aus der Nodestruktur und speichert die Taskstruktur zwischen. In ARTM ist das in der Taskliste am STATE "Frozen" zu erkennen. Wird ein frozen Task ausgewählt, kann er mit Activate wieder aktiviert werden.

Activate:

ist die Umkehrfunktion von Freeze und aktiviert einen frozen Task.

1.39 unlock

Unlock:

entfernt einen Lock (read oder write) aus der Liste. Hier ist Vorsicht angebracht. Wenn das System den Lock noch kennt, geht die Reise nach Indien.

1.40 more

More:

gibt zusätzliche Structure-Informationen. Dies gilt in dieser Version nur für Tasks, Windows und Screens.
Beispiel: Ein gewünschter Task wird mittels Mausklick selektiert, danach einmal ↩
More anwählen. Das Ergebnis sieht dann so aus.

Task- Process- und ComandLineInterface Structure plus Hunks des ToolManger:

Task Structure:

```
-----
IDNestCnt:          -1                TDNestCnt:          0
SigAlloc:           c000ffff          SigWait:           c000f000
SigRecvd:           00000100          SigExcept:         00000000
```

```

TrapAlloc:      00008000          TrapAble:      00000000
ExceptData:     00000000          ExceptCode:   002033f0
TrapData:       00000000          TrapCode:    0021678e
SPLower:        0037ca48          SPUpper:     0037da48
SPReg:          0037d946
Switch():       00000000          Launch():    00000000
MemEntry:       00389b16          UserData:    00000000
Process Structure:
WindowPtr:      00000000          Title:
CurrentDir:     00371fa8          Path:  Amiga:
MsgPort:        00000000
StackBase:      00389bac          StackSize:    3200
CIS:            0036577c          COS:          003654a4
ConsoleTask:    00000000          FileSystemTask: 00294ca4
SegList:        00367328          SegPointer:   00000000
PktWait:        00000000          CLI:          000d98ff
CommandLineInterface Structure:
CommandDir:     002c94a4
StandardInput:  0036577c          StandardOutput: 003654a4
CurrentInput:   0036577c          CurrentOutput: 003654a4
Module:         0038c9dc
Hunks:
Lower      Upper      Size
-----
0038c9e0   00396520   39752
0037b7b8   0037ca40    4752
00023378   00023408     152
total:                                44656

```

Bei Memory wird die Speicherfragmention in Adresse und Size aufgelistet.

1.41 break

Break:

bricht einen laufenden Prozess ab, wenn CTRL Break nicht ausgeschaltet wird. Es werden die Signale Control D/E/F/C gesetzt.

1.42 signal

Signal:

öffnet einen kleinen Requester. Hier können die Werte von tc_SigAlloc in Hex verändert werden. Anschließend werden sie mittels DosFunktion Signal zurückgegeben. Dies tut Break auch.

1.43 cancel

Cancel:

bietet die Möglichkeit die untere Gadgetreihe von Aktion (color 4) in Funktion (color 2) zurückzuschalten, damit wieder alle Funktionen wie z.B Monitor oder Hardware wieder zugänglich sind.

1.44 monitor_acts

Address:

Öffnet einen kleinen Requester. Hier ist die zu suchende Adresse einzugeben. Der Monitor sucht die eingebene Adresse und stellt sie oben links da und gibt anschließend die nächsten 100 Zeilen aus.

Forward:

Blättert eine Seite vor und zeigt die nächsten 100 Zeilen an.

Backward:

Blättert eine Seite zurück und zeigt die vorherigen 100 Zeilen an.

Search:

Öffnet einen kleinen Requester. Hier ist der zu suchende ASCII String einzugeben. Der Monitor sucht den eingegebenen String und stellt ihn oben links da. Die nächsten 100 Zeilen werden mit ausgegeben.

1.45 eh

Enforcer Hits:

Trotz aller Bemühungen sämtlich Enforcer Hits zu vermeiden, ging das nicht immer. Bei Funktion Last Alert z.B. komme ich ohne Hits an die Daten. Und bei Funktion Monitor treten auch Hits auf und zwar dann wenn per absoluter Adresse auf einen Bereich zugegriffen wird wo kein Memory existiert.

Sollten sonst noch Enforcer Hits auftreten bitte ich euch dies mit einer detaillierten Beschreibung mir zu melden.

1.46 history

History:

Die Programm-History von ARTM liegt in einem extra File bei. Dies ist wegen der Übersichtlichkeit geschehen. Außerdem würde das History file das ARTM.guide nur aufblähen.

1.47 zukunft

Zukunft:

Einige neue Features habe ich noch vor in ARTM einzubauen. Das ein oder andere läßt aber noch einige Zeit auf sich warten, da ich momentan durch meinen Job voll ausgelastet bin. Ich stelle mir vor folgende Funktionen zu integrieren:

Darstellung der CPU-Auslastung

An- und ausschalten der CPU-Cache

Ein "Preference-Programm" statt Tooltypes u. Shell-Optionen

Einen TrapHandler zum Abfangen von Alerts

Detaillierte Ausgabe zu Funktion Last Alert

Ein AREXX Port

Eine shared Library board.library für Zusatz-Hardware

1.48 hinweis

Hinweis:

Eine Bitte vorweg an alle, die die Funktion Hardware nutzen und Erweiterungskarten installiert haben. Teilen Sie mir doch bitte in einem kurzen Brief, oder an untenstehende Node mit, was bei PROD TYPE und MANUFACTRE rauskommt. Dazu dann auch den Namen, Typenbezeichnung und Hersteller der Karte.

Alle Funktionen sollten nur mit Vorsicht benutzt werden. Wünsche für die nächste Version und Bugreports (hoffentlich nicht) bitte an die Autoren. Und vergeßen Sie nicht den ausreichend frankierten Rückumschlag. Am einfachsten erreichen Sie mich über UUCP oder Fidonet siehe Autor.

1.49 Danksagungen

Danksagungen:

Vielen Dank an alle, die mir beim Testen mit Tips oder kleinen Routinen oder bei der Dokumentation geholfen haben. Insbesondere an:

Matthias Zepf, Markus Stoll, Matthias Scheler, Kai Bolay, Marc Heuler
Gregor Hutla, Paul Browne

und an alle anderen die mir durch fundierte Bugreports geholfen haben. Aber auch an alle die durch ihren Sharware-Beitrag an der Weiterentwicklung von ARTM mitgeholfen haben.

1.50 Index

Index:

Activate	Address
Aktionen	Assign
Author	Backward
Break	Cancel
Clear	Close
Copyright	Danksagungen
Devices	Disksymbol
Enforcer Hits	Fonts
Forward	Freeze
Funktionen	Hardware
History	Hinweis
I	Installation
Interrupts	Last Alert
Libraries	Locks
Memory	Monitor
More	Mount
Ports	Priority
Programmstart	Relase
Remove	Resident Cmds
Residents	Resources
Search	Semaphoren
Shareware	Signal
Tasks	Unlock
Vectors	Windows
Zukunft	zz