

Miami_deutsch

COLLABORATORS

	<i>TITLE :</i> Miami_deutsch		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 31, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Miami_deutsch	1
1.1	Miami_deutsch.guide	1
1.2	Miami_deutsch.guide/NODE_DISCLAIMER	2
1.3	Miami_deutsch.guide/NODE_CONDITIONS	4
1.4	Miami_deutsch.guide/NODE_REGISTRATION	5
1.5	Miami_deutsch.guide/NODE_INTRODUCTION	6
1.6	Miami_deutsch.guide/NODE_REQUIREMENTS	7
1.7	Miami_deutsch.guide/NODE_INSTALLATION	8
1.8	Miami_deutsch.guide/NODE_MIAMIINIT	8
1.9	Miami_deutsch.guide/NODE_MIAMIINITSANAT	9
1.10	Miami_deutsch.guide/NODE_TOOLTYPES	10
1.11	Miami_deutsch.guide/NODE_MENU	10
1.12	Miami_deutsch.guide/NODE_CONFIGURATION	11
1.13	Miami_deutsch.guide/NODE_GUI_GENERAL	12
1.14	Miami_deutsch.guide/NODE_GUI_GENERAL_REGISTER	12
1.15	Miami_deutsch.guide/NODE_GUI_INTERFACE	13
1.16	Miami_deutsch.guide/NODE_GUI_INTERFACE_TYPE	14
1.17	Miami_deutsch.guide/NODE_GUI_INTERFACE_DEVICE	15
1.18	Miami_deutsch.guide/NODE_GUI_INTERFACE_SPEED	15
1.19	Miami_deutsch.guide/NODE_GUI_INTERFACE_CD	16
1.20	Miami_deutsch.guide/NODE_GUI_INTERFACE_PROTOCOL	16
1.21	Miami_deutsch.guide/NODE_GUI_INTERFACE_FLOW	16
1.22	Miami_deutsch.guide/NODE_GUI_INTERFACE_EOF	17
1.23	Miami_deutsch.guide/NODE_GUI_INTERFACE_SERIAL	17
1.24	Miami_deutsch.guide/NODE_GUI_INTERFACE_IP	18
1.25	Miami_deutsch.guide/NODE_GUI_INTERFACE_MASK	18
1.26	Miami_deutsch.guide/NODE_GUI_INTERFACE_GWAY	19
1.27	Miami_deutsch.guide/NODE_GUI_INTERFACE_MULTICASTS	19
1.28	Miami_deutsch.guide/NODE_GUI_INTERFACE_MAPPING	20
1.29	Miami_deutsch.guide/NODE_GUI_INTERFACE_MTU	21

1.30	Miami_deutsch.guide/NODE_GUI_INTERFACE_STP	21
1.31	Miami_deutsch.guide/NODE_GUI_INTERFACE_INACTIVITY	22
1.32	Miami_deutsch.guide/NODE_GUI_PPP	23
1.33	Miami_deutsch.guide/NODE_GUI_PPP_CHAP	23
1.34	Miami_deutsch.guide/NODE_GUI_PPP_CALLBACK	24
1.35	Miami_deutsch.guide/NODE_GUI_PPP_VJC	24
1.36	Miami_deutsch.guide/NODE_GUI_PPP_ACCM	25
1.37	Miami_deutsch.guide/NODE_GUI_PPP_QUICK	25
1.38	Miami_deutsch.guide/NODE_GUI_PPP_ESCAPE	26
1.39	Miami_deutsch.guide/NODE_GUI_PPP_DNSIPCP	26
1.40	Miami_deutsch.guide/NODE_GUI_PPP_TERMREQ	26
1.41	Miami_deutsch.guide/NODE_GUI_DIALER	27
1.42	Miami_deutsch.guide/NODE_GUI_DIALER_SCRIPT	27
1.43	Miami_deutsch.guide/NODE_GUI_DIALER_PHONE	28
1.44	Miami_deutsch.guide/NODE_GUI_DIALER_MAX	28
1.45	Miami_deutsch.guide/NODE_GUI_DIALER_DELAY	29
1.46	Miami_deutsch.guide/NODE_GUI_DIALER_RDELAY	29
1.47	Miami_deutsch.guide/NODE_GUI_DIALER_TEACH	29
1.48	Miami_deutsch.guide/NODE_GUI_DIALER_NAME	29
1.49	Miami_deutsch.guide/NODE_GUI_DIALER_CAPTURE	30
1.50	Miami_deutsch.guide/NODE_GUI_DATABASE	30
1.51	Miami_deutsch.guide/NODE_GUI_DATABASE_PROTOCOLS	31
1.52	Miami_deutsch.guide/NODE_GUI_DATABASE_SERVICES	32
1.53	Miami_deutsch.guide/NODE_GUI_DATABASE_HOSTS	32
1.54	Miami_deutsch.guide/NODE_GUI_DATABASE_NETWORKS	32
1.55	Miami_deutsch.guide/NODE_GUI_DATABASE_DOMAINS	33
1.56	Miami_deutsch.guide/NODE_GUI_DATABASE_DNSSERVERS	33
1.57	Miami_deutsch.guide/NODE_GUI_DATABASE_INETD	33
1.58	Miami_deutsch.guide/NODE_GUI_DATABASE_USERS	34
1.59	Miami_deutsch.guide/NODE_GUI_DATABASE_GROUPS	35
1.60	Miami_deutsch.guide/NODE_GUI_DATABASE_ARP	35
1.61	Miami_deutsch.guide/NODE_GUI_DATABASE_IPFILTER	35
1.62	Miami_deutsch.guide/NODE_GUI_TCPIP	36
1.63	Miami_deutsch.guide/NODE_GUI_TCPIP_HOSTNAME	37
1.64	Miami_deutsch.guide/NODE_GUI_TCPIP_NAME	38
1.65	Miami_deutsch.guide/NODE_GUI_TCPIP_ICMP	38
1.66	Miami_deutsch.guide/NODE_GUI_TCPIP_BOOTP	38
1.67	Miami_deutsch.guide/NODE_GUI_TCPIP_VERIFYDNS	39
1.68	Miami_deutsch.guide/NODE_GUI_TCPIP_FAKEIP	39

1.69	Miami_deutsch.guide/NODE_GUI_TCPIP_TTCP	39
1.70	Miami_deutsch.guide/NODE_GUI_TCPIP_ADDDOMAIN	40
1.71	Miami_deutsch.guide/NODE_GUI_TCPIP_DOWN	40
1.72	Miami_deutsch.guide/NODE_GUI_TCPIP_PING	41
1.73	Miami_deutsch.guide/NODE_GUI_TCPIP_GETTIME	41
1.74	Miami_deutsch.guide/NODE_GUI_EVENTS	42
1.75	Miami_deutsch.guide/NODE_GUI_MODEM	43
1.76	Miami_deutsch.guide/NODE_GUI_MODEM_INIT	43
1.77	Miami_deutsch.guide/NODE_GUI_MODEM_EXIT	43
1.78	Miami_deutsch.guide/NODE_GUI_MODEM_PREFIX	44
1.79	Miami_deutsch.guide/NODE_GUI_MODEM_SUFFIX	44
1.80	Miami_deutsch.guide/NODE_GUI_MODEM_NULLMODEM	44
1.81	Miami_deutsch.guide/NODE_GUI_LOGGING	44
1.82	Miami_deutsch.guide/NODE_GUI_LOGGING_CONSOLE	45
1.83	Miami_deutsch.guide/NODE_GUI_LOGGING_FILE	45
1.84	Miami_deutsch.guide/NODE_GUI_LOGGING_SYSLOG	45
1.85	Miami_deutsch.guide/NODE_GUI_LOGGING_PHONE	46
1.86	Miami_deutsch.guide/NODE_GUI_GUI	46
1.87	Miami_deutsch.guide/NODE_GUI_GUI_REQQUIT	47
1.88	Miami_deutsch.guide/NODE_GUI_GUI_REQOFFLINE	47
1.89	Miami_deutsch.guide/NODE_GUI_GUI_REQERRORS	47
1.90	Miami_deutsch.guide/NODE_GUI_GUI_DIALER	47
1.91	Miami_deutsch.guide/NODE_GUI_MISC	48
1.92	Miami_deutsch.guide/NODE_DIALERLANG	48
1.93	Miami_deutsch.guide/NODE_AREXX	50
1.94	Miami_deutsch.guide/NODE_EXCONFIG	50
1.95	Miami_deutsch.guide/NODE_EXCONFIG_DIST	51
1.96	Miami_deutsch.guide/NODE_EXCONFIG_PASSWORDS	54
1.97	Miami_deutsch.guide/NODE_EXCONFIG_CLIENTS	55
1.98	Miami_deutsch.guide/NODE_UTILITY	56
1.99	Miami_deutsch.guide/NODE_UTILITY_ARP	57
1.100	Miami_deutsch.guide/NODE_UTILITY_FINGER	58
1.101	Miami_deutsch.guide/NODE_UTILITY_IFCONFIG	58
1.102	Miami_deutsch.guide/NODE_UTILITY_MAPMBONE	59
1.103	Miami_deutsch.guide/NODE_UTILITY_MRINFO	60
1.104	Miami_deutsch.guide/NODE_UTILITY_MROUTED	61
1.105	Miami_deutsch.guide/NODE_UTILITY_MTRACE	62
1.106	Miami_deutsch.guide/NODE_UTILITY_NETSTAT	64
1.107	Miami_deutsch.guide/NODE_UTILITY_PING	68

1.108Miami_deutsch.guide/NODE_UTILITY_RESOLVE	69
1.109Miami_deutsch.guide/NODE_UTILITY_ROUTE	69
1.110Miami_deutsch.guide/NODE_UTILITY_SYSCTL	71
1.111Miami_deutsch.guide/NODE_UTILITY_TCPDUMP	73
1.112Miami_deutsch.guide/NODE_UTILITY_TRACEROUTE	75
1.113Miami_deutsch.guide/NODE_COMPATIBILITY	76
1.114Miami_deutsch.guide/NODE_RESTRICTIONS	77
1.115Miami_deutsch.guide/NODE_HISTORY	78
1.116Miami_deutsch.guide/NODE_FUTURE	90
1.117Miami_deutsch.guide/NODE_SUPPORT	90
1.118Miami_deutsch.guide/NODE_ACKNOWLEDGEMENTS	91

Chapter 1

Miami_deutsch

1.1 Miami_deutsch.guide

Miami

Dies ist die Dokumentation für Miami V2.1, ein integriertes TCP/IP-System für AmigaDOS. Copyright (C) 1996,1997 Nordic Global Inc. All rights reserved. Dokumentation von Holger Kruse.

Haftung

Rechtliche Informationen

Gebrauch / Verbreitung

Bedingungen für Gebrauch und Verbreitung

Registrierung

Shareware-Registrierung

Einführung

Einführung in Miami

Voraussetzungen

Benötigte Hardware und Software

Installation

Installation von Miami

MiamiInit

Einfacher Start mit MiamiInit

MiamiInitSANA2

Einfacher Start mit MiamiInitSANA2

ToolTypes

ToolTypes für Miami

Menüs

Programmmenüs

Konfiguration	Manuelle Konfiguration
Wählmodul-Befehlssprache	Beschreibung des Wählmoduls
ARexx-Schnittstelle	Unterstützte ARexx-Befehle
Austauschen von Einstellungen	Im-/Exportieren von Einstellungen
Hilfsprogramme	Hilfsprogramme für Miami
Kompatibilität	Kompatibilität von Miami
Einschränkungen	Einschränkungen der aktuellen Version ↔
Entwicklung	Entwicklung von Miami
Die Zukunft	Die Zukunft von Miami
Unterstützung	Hilfe und neue Programmversionen
Danksagungen	Danksagungen

1.2 Miami_deutsch.guide/NODE_DISCLAIMER

Haftung

Miami WURDE ALS TCP/IP-PAKET FÜR AmigaOS ENTWICKELT, UM DEN AMIGA ÜBER MODEM ODER MITTELS EINER SCHNITTSTELLENKARTE MIT DEM INTERNET ZU VERBINDEN. OBWOHL ANSTRENGUNGEN UNTERNOMMEN WORDEN SIND, Miami SO KOMPATIBEL ZUM TCP/IP-STANDARD WIE MÖGLICH ZU MACHEN, KANN ICH TROTZDEM NICHT AUSSCHLIESSEN, DASS Miami FEHLER ENTHÄLT, DIE SCHÄDLICHE NEBENWIRKUNGEN AUF IHR SYSTEM ODER ANDERE ANGESCHLOSSENE RECHNER HABEN.

ICH LEHNE HIERMIT AUSDRÜCKLICH JEGLICHE RECHTLICHE ODER FINANZIELLE VERANTWORTUNG FÜR DIESE ODER ANDERE KONSEQUENZEN AUS DEM GEBRAUCH VON Miami AB. DAS BEINHÄLTET, BESCHRÄNKT SICH ABER NICHT AUF SCHÄDEN AN IHREM GERÄT, IHREN DATEN, ANDEREN MASCHINEN, DIE MIT IHREM AMIGA VERBUNDEN SIND, PERSONENSCHÄDEN, FINANZIELLE SCHÄDEN, ODER

IRGENDWELCHE ANDEREN NEBENWIRKUNGEN.

Miami WIRD IHNEN VON MIR SO ZUR VERFÜGUNG GESTELLT, WIE SIE ES VORGEFUNDEN HABEN, DAS HEISST ICH GARANTIERE NICHT, DASS Miami FÜR IRGEND EINEN BESONDEREN ZWECK GEEIGNET IST, ODER DASS ES NEUERE PROGRAMMVERSIONEN, FEHLERKORREKTUREN, ODER HILFE BEI DER FEHLERSUCHE GEBEN WIRD.

Miami basiert auf dem 4.4BSD V.2 TCP/IP Netzwerk-Code, in der Version, die von Walnut Creek auf CD-ROM vertrieben wird.

Der gesamte 4.4BSD-Code darf frei vertrieben werden, und steht unter Copyright von verschiedenen Autoren. Für Details zu individuellen Copyright- und Vertriebs-Bedingungen sehen Sie bitte in den Quelltexten nach, die von verschiedenen Quellen erhältlich sind, z.B. von dem 4.4BSD Lite-CD-ROM von Walnut Creek.

Der folgende Copyright-Hinweis bezieht sich auf das original-4.4BSD-Software-Paket:

Zitat Anfang

All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by The Regents of the University of California.

Copyright 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors. 4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Zitat Ende

Bitte beachten Sie, daß sich dieser Copyright-Hinweis NICHT auf das Miami-Programmpaket bezieht. Miami ist NICHT frei verbreitbar, sofern das nicht an anderer Stelle ausdrücklich erlaubt ist. Siehe

Gebrauch / Verbreitung
für Details.

Miami benötigt Magic User Interface (MUI). MUI steht unter Copyright von Stefan Stuntz.

Miami benötigt die MUI-Custom-Class "Busy.mcc" von Klaus Melchior. Hier ist der dazugehörige Copyright-Hinweis:

Zitat Anfang

Busy.mcc is (c) 1994-1996 by Klaus 'kmel' Melchior

Zitat Ende

1.3 Miami_deutsch.guide/NODE_CONDITIONS

Gebrauch / Verbreitung

Miami ist Shareware. In diesem Fall bedeutet das, daß Sie ein persönliches Keyfile benötigen, um die volle Funktionalität von Miami zu nutzen.

Anwender bekommen ein persönliches Keyfile von mir, nachdem sie sich registriert haben. Das Keyfile darf anderen Anwendern nicht zur Verfügung gestellt werden. Die Weitergabe des Keyfiles an andere Anwender und die Benutzung von Keyfiles, die Sie nicht direkt von mir zu Ihrem persönlichen Gebrauch bekommen haben, gelten als Software-Piraterie.

Keyfiles sind nicht übertragbar, und dürfen nicht an eine andere Person oder Organisation verkauft oder getauscht werden. Sie sind ausschließlich für die Person bestimmt, die sich registriert hat.

Die ausführbare Programmdatei "Miami" und die Programmdateien von Hilfsprogrammen dürfen in keiner Weise modifiziert oder gepatcht werden (auch nicht zum persönlichen Gebrauch), außer auf eine Art und Weise, die von mir ausdrücklich erlaubt wurde, um auf neue Versionen upzudaten. Der Gebrauch von gepatchten oder modifizierten Programmdateien gilt als Software-Piraterie.

Miami-Programmdateien dürfen nur in der vom Autor beabsichtigten Weise genutzt werden, d.h. sie dürfen nur von Amiga-Computern mit AmigaOS ausgeführt werden. Reassembeln, Reverse-Engineering oder das Übersetzen der Programmdateien sind ausdrücklich verboten.

Die Dokumentation und Programmtexte von Miami unterliegen den gleichen Copyright-Bestimmungen wie das Programm selbst. Das heisst,

daß weder die Dokumentation noch die Programmtexte in irgendeiner Weise verändert oder übersetzt werden dürfen.

Um jegliches Mißverständnis zu vermeiden: SIE DÜRFEN die Miami-Programmtexte und -Dokumentation NICHT übersetzen, und die Übersetzungen dann vertreiben, es sei denn, Sie werden von mir ausdrücklich als offizieller Miami-Übersetzer benannt. Unauthorisierte Übersetzungen der Programmtexte oder Dokumentation sind illegal, verletzen mein Copyright, und werden von öffentlichen Software-Sites gelöscht.

Wenn Sie das Miami-Archiv vertreiben möchten, gelten die folgenden Bedingungen:

- * Der Verkaufspreis darf nicht höher sein als die Kosten für die für die Miami-Dateien benötigten Leerdisketten plus einer nominellen Kopiergebühr plus Kosten für Versand. Der Gesamtpreis darf nicht höher sein als 10 US\$ oder 15 DM oder dem entsprechenden Betrag in anderen Währungen.
- * Wenn das Miami-Archiv als Teil einer CD-ROM-Sammlung von Public Domain- und/oder Shareware-Software vertrieben werden soll, dann darf der Verkaufspreis des CD-ROM 20 US\$, 30 DM oder den entsprechenden Betrag in anderen Währungen nicht überschreiten.
- * Alle Teile des Programms und der Dokumentation müssen vollständig sein. Der Vertrieb von Teilen oder unvollständigen Archiven ist nicht gestattet. Der Vertrieb von Keyfiles ist nicht gestattet.
- * Miami oder Teile davon dürfen normalerweise nicht in Verbindung mit oder als Teil von kommerzieller Software vertrieben werden. Auf Anfrage sind Lizenzbedingungen für den kommerziellen Verkauf von Miami von kruse@nordicglobal.com erhältlich. Allerdings sollten Sie nicht annehmen, daß Sie Miami kommerziell vertreiben dürfen, bis Sie von mir eine ausdrückliche, schriftliche Zusage erhalten.
- * Programm und Dokumentation dürfen nicht verändert werden. Ausnahme (d.h. akzeptabel ist): die Verwendung von Archivierprogrammen wie LHA, wenn dabei garantiert ist, daß es möglich ist, die Originaldateien zu extrahieren.

1.4 Miami_deutsch.guide/NODE_REGISTRATION

Registrierung

Wenn Sie Miami häufig nutzen, Funktionen, die in der Demoversion nicht verfügbar sind, benötigen, oder Ihre Verbindung länger als eine Stunde aufrecht erhalten möchten, sollten Sie sich für Miami registrieren.

Um sich zu registrieren, starten Sie bitte das Programm MiamiRegister. Es erklärt die Registrierprozedur im Detail, und

erlaubt Ihnen, sich im Dialog zu registrieren.

Wenn Sie aus irgend einem Grund MiamiRegister nicht starten können, kontaktieren Sie mich bitte bei kruse@nordicglobal.com.

Die Registrierg Gebühr ist US\$ 35 für eine normale 'Volllizenz'.

Für Registrierungen innerhalb Deutschland wird derzeit der folgende Händler empfohlen:

ADX Datentechnik GmbH
Hallesdorferstr. 119
22179 Hamburg

Tel. 040 - 642 02656

Internet: <http://www.adx.de>
Infos zu Miami: miami@adx.de

Registrierte Anwender von ppp.device erhalten einen Rabatt beim Umstieg auf Miami, wenn Sie sich direkt beim Autor registrieren. Die Details werden in MiamiRegister erklärt.

Sonderangebote für Gruppenregistrierung (10 oder mehr Anwender auf einmal), Vorausbezahlung von Lizenzen und kommerziellen Wiederverkauf sind ebenfalls möglich. Kontaktieren Sie bitte kruse@nordicglobal.com für mehr Details.

1.5 Miami_deutsch.guide/NODE_INTRODUCTION

Einführung

Miami ist ein integriertes TCP/IP-System für AmigaDOS, mit dem Sie auf sehr einfache Weise auf das Internet oder ein lokales Netzwerk zugreifen können, sowohl per Modem als auch per Schnittstellenkarte (z.B. Ethernet).

Miami basiert auf der aktuellen Version (4.4BSD V2) des offiziellen BSD-Netzwerk-Codes, plus einiger Erweiterungen Dritter (wie z.B. FreeBSD-T/TCP- und Path-MTU-Discover-Code). Das heisst Miami ist ein "echter" und vollständiger TCP/IP-Stack, nicht eine Emulation, die nur Teile des TCP/IP-Standards unterstützt.

Die Programmierschnittstelle von Miami ist kompatibel mit der von AmiTCP 4.x verwendeten (`bsdsocket.library`), d.h. die meisten für AmiTCP 4.x geschriebenen und kompilierten Programme funktionieren ohne Änderung und ohne neu-compilieren mit Miami.

Ausserdem enthält Miami ein eingebautes Wählmodul, das sowohl per Skript gesteuert, als auch interaktiv genutzt werden kann, eine Implementierung der (C)SLIP- und PPP-Protokolle und eine graphische Benutzeroberfläche für Programmsteuerung und -konfiguration.

Miami hat zusätzlich eine eingebaute Implementierung von `inetd`, dem

"Internet-Super-Server", mit diversen eingebauten Diensten einschließlich "fingerd" und "identd", eine eingebaute Implementierung von TCP:, dem AmigaDOS-Stream-Handler für TCP/IP, und eine eingebaute Implementierung der usergroup.library, dem Interface zur Verwaltung von Anwendern und Anwendergruppen.

Im Gegensatz zu anderen Mehrzweckstacks enthält Miami eine umfangreiche Unterstützung für Modem-basierte Internet-Anwahl. Die Konfiguration dafür ist so einfach wie möglich gehalten: die meisten Konfigurationsparameter werden von Miami automatisch ermittelt. Ausserdem unterstützt Miami die Verwendung von vorkonfigurierten Einstellungen für bestimmte Internet-Provider. Miami kann auch mit anderen Verbindungen außer Modem-Verbindungen genutzt werden, z.B. mit einer Ethernet-Karte, einer Arcnet-Karte oder einem Kabelmodem.

Miami verwendet MUI 3.3 oder höher für die Benutzeroberfläche, d.h. Sie müssen MUI installiert haben, bevor Sie Miami nutzen können.

Vor dem Start sollten Sie sich zuerst
MiamiInit
(für PPP- oder
SLIP-Verbindungen) oder
MiamiInitSANA2
(für SANA-II-Verbindungen)
ansehen.

MiamiInit und MiamiInitSANA2 sind Programme, die Miami für die meisten Anwender automatisch konfigurieren, einschließlich Wählskript, Passwort-Abfrage, IP-Adresse, DNS-Server, Netzmaske und allen anderen Konfigurationsparametern.

Nachdem Sie MiamiInit oder MiamiInitSANA2 ausgeführt haben, sollten Sie Miami starten, die importieren, die Einstellungen speichern, und die Verbindung zu Ihrem Provider aufbauen.

1.6 Miami_deutsch.guide/NODE_REQUIREMENTS

Voraussetzungen

Um Miami zu benutzen, brauchen Sie:

- * einen Amiga mit OS 2.04 oder höher
- * MUI 3.3 oder höher

Sie brauchen auch irgendwelche Hardware für die Netzwerkanbindung, und einen Rechner, zu dem Sie die Verbindung aufbauen möchten. Das können z.B. sein:

- * ein mit dem Amiga verbundenes Modem und eine Telefonleitung. Das Modem sollte einigermaßen Hayes-kompatibel sein. Das trifft auf die meisten modernen Modems zu. Ausserdem brauchen Sie dann einen

SLIP- oder PPP-Zugang bei einem Internet-Provider. Wenn Sie nur einen Shell-Zugang haben, können Sie Miami damit auch verwenden, aber Sie müssen dann zunächst Slirp oder TIA bei Ihrem Provider installieren. In diesem Fall sollten Sie Ihren Provider zunächst um Erlaubnis fragen, und dann auch, wie und woher Sie Slirp oder TIA am besten bekommen können.

- * eine Ethernet-Karte, ein Kabelmodem, und einen SLIP/PPP-Zugang, wie oben beschrieben.
- * eine Ethernet-Karte, die Ihren Amiga mit einem lokalen Netz verbindet.

Beachten Sie, daß Miami ppp.device, appp.device, amipp.device oder (rh)(c)slip.device nicht benötigt. Die Protokolle PPP und (C)SLIP sind in Miami eingebaut, und zwar in Versionen, die effizienter und fortgeschrittener sind als die derzeit erhältlichen SANA-II-Treiber.

1.7 Miami_deutsch.guide/NODE_INSTALLATION

Installation

Bitte starten Sie das beiliegende Installer-Skript, um Miami auf Ihrer Festplatte zu installieren. Das Installer-Skript kann für eine Neu-Installation und für Updates benutzt werden, und sowohl zum Kopieren der Dateien während der Installation, als auch zum Installieren im aktuellen Verzeichnis.

Alle Dateien werden in einem einzigen Verzeichnis installiert, und keine Systemdateien oder Systemverzeichnisse werden von Miami verändert, mit einer Ausnahme:

Das Installer-Skript fragt Sie, ob Sie den "Miami:"-Assign erzeugen möchten, und fügt dann den benötigten Befehl zu Ihrer user-startup-Datei hinzu. Das wird für maximale Kompatibilität empfohlen, ist aber nicht absolut notwendig, d.h. Sie können diesen Schritt während der Installation überspringen.

Miami liegen drei verschiedene Icons bei: ein Standardicon, ein Icon für "Magic Workbench" und ein Icon im "NewIcons"-Stil. Während der Installation können Sie das Icon auswählen, das Sie bevorzugen.

1.8 Miami_deutsch.guide/NODE_MIAMIINIT

MiamiInit

MiamiInit ist ein Hilfsprogramm, das versucht, alle Konfigurationsparameter, die Miami für eine serielle Verbindung (SLIP

oder PPP) braucht, automatisch zu ermitteln, und das dann eine Konfigurationsdatei speichert, die später von Miami genutzt werden kann.

Nach dem Installieren von Miami sollten Sie zur Konfiguration einer seriellen Verbindung zuerst MiamiInit starten und dem Dialog folgen. MiamiInit wählt dann Ihren Internet-Provider an, ermittelt alle notwendigen Parameter, und speichert diese zum Schluß.

MiamiInit unterstützt derzeit nur die häufig verwendeten Rechner-Modem-Konfigurationen. Sehr ungewöhnliche Fälle, wie andere Datenformate als 8N1, nicht-Hayes-kompatible Modems oder 3-Draht-Modemkabel werden nicht unterstützt. Wenn Sie eine derartige, ungewöhnliche Konfiguration haben, sollten Sie Miami manuell konfigurieren, statt MiamiInit zu benutzen.

1.9 Miami_deutsch.guide/NODE_MIAMIINITSANAT

MiamiInitSANA2

MiamiInitSANA2 ist ein Hilfsprogramm, das versucht, alle Konfigurationsparameter, die Miami für eine SANA-II-Verbindung (Ethernet, Arcnet usw.) braucht, automatisch zu ermitteln, und das dann eine Konfigurationsdatei speichert, die später von Miami genutzt werden kann.

Nach dem Installieren von Miami sollten Sie zur Konfiguration einer SANA-II-Verbindung zuerst MiamiInitSANA2 starten und dem Dialog folgen. MiamiInitSANA2 greift dann auf Ihre Netzwerkkarte und Ihr lokales Netz zu, und versucht alle notwendigen Parameter zu ermitteln, um Ihre Verbindung zu konfigurieren.

Beachten Sie, daß je nach der Konfiguration Ihres lokalen Netzes MiamiInitSANA2 unter Umständen nicht alle Informationen selbständig finden kann. Es ist kein Fehler, wenn MiamiInitSANA2 Sie nach Dingen wie der IP-Adresse oder der Netzmaske während der Konfiguration fragt. Das bedeutet nur, daß es in Ihrem lokalen Netz keinen Server gibt, der MiamiInitSANA2 diese Art von Information übermittelt. In diesem Fall sollten Sie Ihren Internet-Provider oder Netz-Administrator nach den fehlenden Informationen fragen.

Im Allgemeinen gilt: wenn Sie ein sehr kleines Netz einrichten, das nur aus Amigas und PCs besteht, werden Sie üblicherweise die meisten Informationen selbst eingeben müssen. Wenn Sie dagegen eine Verbindung zu einem grösseren, existierenden Firmennetz aufbauen, daß für die automatische Konfiguration neuer Rechner eingerichtet ist, dann kann MiamiInitSANA2 oft die meisten oder alle Informationen von einem Server ermitteln.

1.10 Miami_deutsch.guide/NODE_TOOLTYPES

ToolTypes

Miami unterstützt die folgenden ToolTypes (beim Start von der Workbench) bzw. Parameter (beim Start von der Shell):

DONTCONNECT

Wenn Sie Miami so konfiguriert haben, daß es beim Start automatisch eine Verbindung zu Ihrem Internet-Provider aufbaut, können Sie das mit dieser Option unterbinden. Das gibt Ihnen die Möglichkeit, vor dem Aufbau der Verbindung noch die Einstellungen zu verändern.

SETTINGS

Ein Projekt-Icon muss ein "SETTINGS"-ToolType haben, damit Miami es als eine Einstellungen-Datei erkennt. Von der Shell können Sie den Parameter "SETTINGS Dateiname" verwenden, um anzugeben, welche Einstellungen-Datei geladen werden soll.

IMPORTMIAMIINIT

"IMPORTMIAMIINIT=Dateiname" veranlaßt Miami, eine Einstellungen-Datei von MiamiInit zu importieren.

IMPORTASCII

"IMPORTASCII=Dateiname" veranlaßt Miami, eine Einstellungen-Datei vom ASCII-Format (Format für Vertrieb) zu importieren.

SAVESETTINGS

"SAVESETTINGS" veranlaßt Miami, die aktuellen Einstellungen als Standardeinstellungen zu speichern. Das ist besonders nützlich in Verbindung mit den ToolTypes "IMPORTMIAMIINIT" und "IMPORTASCII", um Einstellungen zu importieren, und in das Miami-Einstellungen-Format zu konvertieren.

1.11 Miami_deutsch.guide/NODE_MENUS

Menüs

Beschreibung aller Menüpunkte:

Projekt/Über...

Informationen über Miami anzeigen.

Projekt/Über MUI...

Informationen über MUI (Magic User Interface) anzeigen.

Projekt/Iconifizieren

Alle Fenster von Miami iconifizieren.

Projekt/Ende ohne aufzulegen...

Miami beenden, ohne vorher die Modem-Leitung aufzulegen.

Projekt/Ende...

Miami beenden.

Einstellungen/Laden...

Eine Einstellungen-Datei laden.

Einstellungen/Speichern

Einstellungen in der aktuellen Datei speichern.

Einstellungen/Speichern als...

Einstellungen in einer neuen Datei speichern.

Einstellungen/Speichern als Standard

Einstellungen in der Standarddatei speichern.

Einstellungen/Piktogramm erzeugen

Automatisch Piktogramm (Icon) erzeugen, wenn die Einstellungen gespeichert werden.

Importieren von Vertrieb...

Einstellungen-Datei von Miami's Vertriebs-(ASCII-)Format importieren.

Exportieren für Vertrieb...

Einstellungen-Datei in Miami's Vertriebs-(ASCII-)Format exportieren.

Importieren von MiamiInit...

Einstellungen-Datei von MiamiInit importieren.

Importieren von MiamiInitsANA2...

Einstellungen-Datei von MiamiInitsANA2 importieren.

MUI Einstellungen...

MUI-Konfigurationsfenster öffnen.

1.12 Miami_deutsch.guide/NODE_CONFIGURATION

Konfiguration

Die Konfiguration von Miami erfolgt vollständig durch die graphische Benutzeroberfläche. Es müssen keine Konfigurationsdateien oder Environment-Variablen modifiziert werden.

Beschreibung der graphischen Benutzeroberfläche:

Allgemeines

Die 'Allgemeines'-Seite

Schnittstelle	Die 'Schnittstelle'-Seite
PPP	Die 'PPP'-Seite
Wählmodul	Die 'Wählmodul'-Seite
Datenbank	Die 'Datenbank'-Seite
TCP/IP	Die 'TCP/IP'-Seite
Ereignisse	Die 'Ereignisse'-Seite
Modem	Die 'Modem'-Seite
Protokollierung	Die 'Protokollierung'-Seite
GUI	Die 'GUI (Benutzeroberfläche) '- ↔ Seite
Sonstiges	Sonstige Schalter

1.13 Miami_deutsch.guide/NODE_GUI_GENERAL

Allgemeines

=====

Hier gibt es nicht viel, ausser dem offiziellen Miami-Logo und einem Schalter, um das Miami-Registrierprogramm zu starten.

Registrieren

Der 'Registrieren'-Schalter

1.14 Miami_deutsch.guide/NODE_GUI_GENERAL_REGISTER

Registrieren

Dieser Schalter startet das Programm MiamiRegister, mit dem Sie einen Miami-Lizenzcode bestellen, Miami registrieren oder Ihre Registrierung upgraden können. MiamiRegister muß in dem gleichen Verzeichnis wie Miami sein, oder im Standard-Suchpfad der Shell.

1.15 Miami_deutsch.guide/NODE_GUI_INTERFACE

Schnittstelle

=====

Schnittstellen-Typ

Der 'Schnittstellen-Typ'-Schalter

Gerät / Nummer

Die 'Gerät'- und 'Nummer'-Schalter

Geschwindigkeit

Der 'Geschwindigkeit'-Schalter

Verwende CD

Der 'CD'-Schalter

Protokoll

Der 'Protokoll'-Schalter

Handshaking

Der 'Handshaking'-Schalter

EOF-Modus

Der 'EOF-Modus'-Schalter

Betriebsart

Der 'Betriebsart'-Schalter

IP-Typ / Adresse

Die 'IP'-Schalter

Netzmasken / Typ

Die 'Netzmasken'-Schalter

Gateway / Typ

Die 'Gateway'-Schalter

Multicasts

Der 'Multicasts'-Schalter

Umwandlung

Der 'Umwandlung'-Schalter

MTU

Der 'MTU'-Schalter

SANA-II-Parameter

Der 'SANA-II-Parameter'-Schalter

Inaktivität

Die 'Inaktivität'-Schalter

1.16 Miami_deutsch.guide/NODE_GUI_INTERFACE_TYPE

Schnittstellen-Typ

Dieser Schalter wählt den Typ der zu verwendenden Schnittstelle aus. Das genaue Aussehen der 'Schnittstelle'-Seite hängt davon ab, welcher Schnittstellen-Typ gewählt wird, d.h. es werden nur solche Schalter angezeigt, die für den ausgewählten Schnittstellentyp von Bedeutung sind.

Mögliche Typen:

seriell (PPP/SLIP)

Die eingebaute Standard-Implementierung von PPP oder (C)SLIP, in Verbindung mit einem beliebigen serial.device-kompatiblen Treiber. Dieser Schnittstellen-Typ benötigt keinen SANA-II-Treiber.

SANA-II Punkt-zu-Punkt

Ein SANA-II-Treiber für eine Hardware, die genau zwei Rechner verbindet, z.B. SLIP, PPP oder PLIP.

SANA-II Ethernet

Ein SANA-II-Treiber für eine Ethernet-Karte, z.B. die A2065-Karte, die Hydra-Karte, oder die Ariadne-Karte.

SANA-II "altes" Standard-Arcnet

Ein SANA-II-Treiber für eine Arcnet-Karte, z.B. die A2060-Karte. Diese Einstellung verwendet die normale "alte" RFC1051-Arcnet-Verkapselung, die in Amiga-Netzen häufiger verwendet wird als die "neue" RFC1201-Verkapselung. Sie sollten die "alte" Verkapselung verwenden, wenn Sie Ihren Amiga mit Rechnern vernetzen wollen, die AmiTCP/IP, I-Net 225 oder NetBSD 1.1 fahren.

SANA-II "neues" Arcnet

Ein SANA-II-Treiber für eine Arcnet-Karte, z.B. die A2060-Karte. Diese Einstellung verwendet die "neue" RFC1201-Arcnet-Verkapselung. Sie ist nicht kompatibel zu AmiTCP/IP oder NetBSD 1.1, aber Sie brauchen diese Einstellung möglicherweise, wenn Sie Ihren Amiga mit anderen Plattformen wie Windows 95 vernetzen wollen.

SANA-II anderer Bus/Ring

Ein SANA-II-Treiber für eine Bus- oder Ring-Hardware ausser Ethernet oder Arcnet. Wählen Sie diese Einstellung, wenn Ihre Hardware mehr als zwei Rechner verbinden kann, aber nicht

kompatibel zu Ethernet oder Arcnet ist.

1.17 Miami_deutsch.guide/NODE_GUI_INTERFACE_DEVICE

Gerät / Nummer

Für serielle Schnittstellen:

Geben Sie den Gerätenamen und die Unit-Nummer des seriellen Ports ein, an dem Ihr Modem angeschlossen ist. Für den im Amiga eingebauten seriellen Port verwenden Sie 'serial.device' '0'.

Für serielle Schnittstellenkarten verwenden Sie den Treiber, der mit der Karte mit geliefert wird, z.B. 'gvpser.device', mit der korrekten Unit-Nummer.

Andere Treiber für den eingebauten seriellen Port wie 'v34serial.device' werden auch unterstützt. Allerdings sollten Sie derzeit nicht '8nl.device' benutzen wegen Fehlern in diesem Treiber. Einige Anwender haben berichtet, daß es auch mit 'BaudBandit.device' Probleme gibt.

Für SANA-II-Schnittstellen:

Geben Sie den Gerätenamen und die Unit-Nummer Ihrer SANA-II-Hardware an. Die Unit-Nummer ist in den meisten Fällen 0.

1.18 Miami_deutsch.guide/NODE_GUI_INTERFACE_SPEED

Geschwindigkeit

(Diese Option erscheint nur für serielle Schnittstellen.)

Die Geschwindigkeit Ihres seriellen Ports. Für den internen seriellen Port sollten Sie 19200, 38400 oder (wenn Sie eine schnelle CPU und eine Grafikkarte haben) 57600 benutzen. Für serielle Schnittstellenkarten können Sie möglicherweise auch 115200 oder 230400 verwenden.

Verwenden Sie nicht 31250. Diese Geschwindigkeit ist ausschliesslich für MIDI reserviert und funktioniert mit den meisten Modems nicht.

Verwenden Sie auch nicht 14400, 28800 oder 33600. Ihr Modem mag zwar mit diesen Geschwindigkeiten Verbindungen zur Gegenseite aufbauen, aber unterstützt diese Geschwindigkeiten wahrscheinlich nicht für die serielle Schnittstelle.

1.19 Miami_deutsch.guide/NODE_GUI_INTERFACE_CD

Verwende CD

(Diese Option erscheint nur für serielle Schnittstellen.)

Wenn "Verwende CD" aktiviert ist, dann verwendet Miami die "Carrier Detect"-Leitung Ihres Modems um festzustellen, ob Ihr Modem bereits mit der Gegenseite verbunden ist oder nicht.

Das kann nützlich sein, wenn Sie Ihren Amiga resetten ohne daß das Modem auflegt, damit Sie Miami neu starten können, ohne neu zu wählen.

Diese Option kann nur verwendet werden, wenn Ihr Modem so konfiguriert ist, daß die 'Carrier Detect'-Leitung nur aktiv ist, wenn eine Verbindung besteht.

Einige Modems haben Fabrikeinstellungen, die die "Carrier Detect"-Leitung immer auf "high" setzen, auch wenn das Modem nicht verbunden ist. Wenn das auf Ihr Modem zutrifft, müssen Sie entweder die Modem-Einstellungen ändern (normalerweise "AT&C1") und die Einstellungen dann mit einem Terminalprogramm im NV-RAM Ihres Modems speichern (normalerweise "AT&W"), oder "Verwende CD" abschalten.

Wenn Sie die Nullmodem-Einstellungen (konfiguriert auf der "Modem"-Seite) verwenden, dann bekommt dieser Schalter eine andere Bedeutung:

- * Wenn der Schalter aktiviert ist, wird das Wählskript überhaupt nicht ausgeführt.
- * Wenn der Schalter deaktiviert ist, wird das Wählskript ausgeführt, aber Miami wählt keine Telefonnummer an, d.h. der "ATDT..."-Befehl wird übersprungen, und die Telefonnummern-Liste wird bedeutungslos.

1.20 Miami_deutsch.guide/NODE_GUI_INTERFACE_PROTOCOL

Protokoll

(Diese Option erscheint nur für serielle Schnittstellen.)

Das Protokoll, das Ihr Internet-Provider verwendet. Derzeit unterstützt werden SLIP/CSLIP und PPP.

1.21 Miami_deutsch.guide/NODE_GUI_INTERFACE_FLOW

Handshaking.

(Diese Option erscheint nur für serielle Schnittstellen.)

Miami unterstützt derzeit zwei Handshaking-Methoden: Hardware-Handshaking (RTS/CTS) und Software-Handshaking (Xon/Xoff). Die Standardeinstellung ist Hardware-Handshaking, und es wird dringend empfohlen, das auch nicht zu ändern.

Wenn Sie Hardware-Handshaking nicht verwenden können (normalerweise wegen eines defekten Modems, Kabels, oder seriellen Ports), sollten Sie auf Software-Handshaking umschalten. Achten Sie dann aber darauf, dass der Initialisierungsstring Ihres Modems (im Wählmodul-Fenster) entsprechend angepasst wird. Ausserdem ist Software-Handshaking nur mit PPP möglich, nicht mit SLIP/CSLIP.

1.22 Miami_deutsch.guide/NODE_GUI_INTERFACE_EOF

EOF-Modus

(Diese Option erscheint nur für serielle Schnittstellen.)

Miami hat zwei Möglichkeiten, das Ende von ankommenden Datenpaketen zu erkennen: Die effizientere (mit geringerem CPU-Zeit-Verbrauch) verwendet das EOF_MODE-Flag. Allerdings funktioniert das nur, wenn der serielle Treiber den EOF-Modus unterstützt. Viele Treiber von Drittherstellern unterstützen diesen Modus nicht.

Normalerweise sollten Sie diesen Schalter auf "auto" stehen lassen, damit Miami die Standardeinstellung benutzt. Wenn Sie genau wissen, ob Ihr Treiber den EOF-Modus unterstützt, können Sie das auch hier einstellen.

1.23 Miami_deutsch.guide/NODE_GUI_INTERFACE_SERIAL

Betriebsart

(Diese Option erscheint nur für serielle Schnittstellen.)

Die Einstellungen für die Anzahl Datenbits und Parität. Für 99% aller Provider ist 8N1 korrekt. Einige wenige Provider (z.B. einige Dialin-Punkte für Compuserve) benötigen möglicherweise 7E1 oder 7O1.

Bitte beachten Sie, daß sich diese Einstellungen nur auf die Wähl- und Loginphase beziehen. Für das (C)SLIP/PPP-Protokoll verwendet Miami immer 8N1, unabhängig davon, was Sie hier einstellen. Es ist abolut

unmöglich, PPP oder (C)SLIP über eine 7-Bit-Leitung zu nutzen - mit jeder PPP/(C)SLIP-Implementierung. Das ist keine Einschränkung in Miami.

1.24 Miami_deutsch.guide/NODE_GUI_INTERFACE_IP

IP-Typ / -Adresse

Internet-Provider bieten üblicherweise zwei Arten von Internet-Verbindungen an: solche mit einer statischen IP-Adresse, die Ihrem Amiga permanent zugewiesen wird, und (häufiger) solche, bei denen Ihr Amiga bei jeder Neu-Verbindung eine andere, dynamische IP-Adresse erhält.

Für serielle Schnittstellen:

Wenn Ihr Amiga eine feste IP-Adresse hat, wählen Sie "statisch" und geben Sie die IP-Adresse ein, die Ihnen Ihr Provider gegeben hat. Wenn Ihr Provider Ihnen eine dynamische IP-Adresse für jede Verbindung zuweist, dann wählen Sie "dynamisch", und Miami ermittelt die IP-Adresse automatisch, wenn Sie eine Verbindung aufbauen.

Wenn Sie TIA oder Slirp verwenden, müssen Sie "statisch" wählen, und die Pseudo-IP-Adresse eingeben, die TIA oder Slirp Ihrem Amiga zugewiesen hat. Mehr Informationen hierzu stehen in der TIA/Slirp-Anleitung.

Für SANA-II-Punkt-zu-Punkt-Verbindungen:

Wenn Ihr Amiga eine feste IP-Adresse hat, wählen Sie "statisch" und geben Sie die IP-Adresse ein. Wenn Ihnen die Adresse von einem BootP-Server zugewiesen wird, dann wählen Sie "BootP". Wenn der SANA-II/Treiber die IP-Adresse selbst ermittelt, dann wählen Sie "SANA-II".

Für SANA-II-Bus/Ring-Schnittstellen:

Wenn Ihr Amiga eine feste IP-Adresse hat, wählen Sie "statisch" und geben Sie die IP-Adresse ein. Wenn Ihnen die Adresse von einem BootP-Server zugewiesen wird, dann wählen Sie "BootP". Wenn Ihnen die Adresse von einem RArp-Server zugewiesen wird, dann wählen Sie "RArp".

1.25 Miami_deutsch.guide/NODE_GUI_INTERFACE_MASK

Netzmasken-Typ / -Adresse

(Diese Option erscheint nur für SANA-II-Bus-/Ring-Schnittstellen.)

Ihre Netzmaske muß korrekt konfiguriert werden, damit Miami weiß,

welche Rechner auf dem lokalen Netz sind. Es gibt drei Möglichkeiten, die Netzmaske einzustellen:

statisch

Fragen Sie Ihren Netz-Administrator nach dem korrekten Wert, und geben Sie ihn ein.

BootP

Miami versucht, den korrekten Wert von einem BootP-Server zu ermitteln.

ICMP

Miami versucht, den korrekten Wert von einem Server per ICMP-Netmask-Discovery zu ermitteln.

1.26 Miami_deutsch.guide/NODE_GUI_INTERFACE_GWAY

Gateway-Typ / -Adresse

(Diese Option erscheint nur für SANA-II-Bus-/Ring-Schnittstellen.)

Ihr Standard-Gateway muß korrekt konfiguriert werden, damit Miami weiß, wohin Packets geschickt werden sollen, die nicht für Rechner auf dem lokalen Netz bestimmt sind. Es gibt drei Möglichkeiten, das Gateway einzustellen:

statisch

Fragen Sie Ihren Netz-Administrator nach dem korrekten Wert, und geben Sie ihn ein.

BootP

Miami versucht, den korrekten Wert von einem BootP-Server zu ermitteln.

ICMP

Miami versucht, den korrekten Wert von einem Server per ICMP-Router-Discovery zu ermitteln.

1.27 Miami_deutsch.guide/NODE_GUI_INTERFACE_MULTICASTS

Multicasts

(Diese Option kann nur in der registrierten Version verändert werden.)

Miami unterstützt Level-2-Multicasting, d.h. sowohl das Senden, als auch das Empfangen von Multicast-Nachrichten.

Wenn Sie Applikationen verwenden, die Multicasting nutzen (bisher gibt es noch keine), müssen Sie möglicherweise Multicasting in Miami einschalten. Die möglichen Einstellungen sind:

inaktiv

Multicasting ist ausgeschaltet.

als Broadcasts

Multicasts werden als Link-Level-Broadcasts (oder bei Punkt-zu-Punkt-Hardware: als normale Pakete) verschickt.

als Multicasts

Multicasts werden als Link-Level-Multicasts verschickt. Diese Option ist nur mit Ethernet möglich.

Hinweis: Multicasts sollten für eine Schnittstelle nur dann eingeschaltet werden, wenn Sie Ihren Multicast-Feed direkt von dieser Schnittstelle erhalten. Wenn Sie Ihren Multicast-Feed durch einen Tunnel mittels MiamiMRouteD bekommen, sollten Sie Multicasts bei Miami's Schnittstelle normalerweise abschalten, da MiamiMRouteD Multicasts selbständig erledigt.

1.28 Miami_deutsch.guide/NODE_GUI_INTERFACE_MAPPING

Umwandlung

(Diese Option erscheint nur für SANA-II-Arcnet-Schnittstellen.)

Arcnet unterstützt zwei verschiedene Methoden, um IP-Adressen in Hardware-Adressen zu konvertieren:

Arp

Arp (Address-Resolution-Protocol) wird verwendet. Das ist die empfohlene Standard-Einstellung, und ist auch, was AmiTCP/IP verwendet.

direkt

Die untersten 8 Bits der IP-Adresse werden auf die Hardware-Adresse abgebildet. Das ist die Methode, die NetBSD 1.1 verwendet.

Wenn Sie mindestens einen NetBSD-1.1-Rechner in Ihrem Arcnet-Netz haben, können Sie die Konfiguration vereinfachen, indem Sie "direkte Umwandlung" wählen, anstatt auf allen Rechnern manuelle Arp-Einträge zu erzeugen.

In allen anderen Fällen sollten Sie auf allen Rechnern "Arp" wählen. Neuere ("current") Versionen von NetBSD 1.2 und höher unterstützen auch Arp für Arcnet. Wenn Sie eine dieser neueren NetBSD-Versionen verwenden, dann wählen Sie bitte "Arp"-Umwandlung in Miami.

1.29 Miami_deutsch.guide/NODE_GUI_INTERFACE_MTU

MTU

(Diese Option erscheint nur für serielle Schnittstellen.)

Maximum Transfer Unit, d.h. die maximale Größe von Paketen, die "am Stück" übertragen werden. Der Standard für PPP ist 1500, der Standard für (C)SLIP ist 1006.

Für serielle Leitungen ist es normalerweise ratsam, die MTU auf einen geringeren Wert als den Standardwert einzustellen, um ein besseres Antwortzeitverhalten für interaktive TCP/IP-Anwendungen zu erhalten. Gute Werte sind:

- * für Modem-Geschwindigkeiten bis zu 19200 bps: MTU=296.
- * für Modem-Geschwindigkeiten über 19200 bps: MTU=552.

Bitte beachten Sie, daß ein Ändern des MTU-Wertes im Konfigurationsfenster nicht unbedingt auch zur Folge hat, daß die maximale Paketgröße wirklich auf diesen Wert verändert wird:

(C)SLIP kann den MTU-Wert überhaupt nicht aushandeln, d.h. der konfigurierte MTU-Wert beeinflusst nur die Größe von verschickten Paketen, nicht die Größe von empfangenen Paketen.

PPP hat Konfigurationsoptionen zum Aushandeln der MTU. Miami versucht immer, den hier eingestellten Wert auszuhandeln, aber es ist möglich, daß die andere Seite das nicht zuläßt. In diesem Fall muß Miami möglicherweise die Werte, die die andere Seite vorgibt, in eine Richtung oder auch in beiden Richtungen verwenden.

Beachten Sie auch: Für PPP ist der MTU-Wert nicht kritisch, d.h. Ihre Verbindung wird auch dann noch funktionieren, wenn der von Ihnen eingestellte Wert größer oder kleiner als der optimale Wert ist. Für (C)SLIP darf der eingestellte Wert jedoch nicht größer sein als der MTU-Wert bei Ihrem Provider.

1.30 Miami_deutsch.guide/NODE_GUI_INTERFACE_STP

SANA-II-Parameter

(Diese Option erscheint nur für SANA-II-Schnittstellen.)

Bei Klicken auf den Schalter "SANA-II-Parameter" öffnet sich ein Fenster mit SANA-II-Geräte-Einstellungen. Diese Einstellungen beinhalten:

- * Die Hardware-Adresse Ihres Gerätes, mit einer Option, diese zu erzwingen. (Nur für Bus-/Ring-Geräte.) Hardware-Adressen werden

als Bytefolge in hexadezimaler Schreibweise, getrennt durch `:'` angegeben, z.B. `01:23:45:67:89:ab`.

- * Der Link-Level-Pakettyp für IP-, Arp- und RArp-Pakete. (RArp ist nicht möglich mit Arcnet, und Arp ist nicht möglich mit Punkt-zu-Punkt-Geräten.)
- * Die MTU für das Gerät.
- * Die Anzahl der IORequests, die für IP- und Arp-Pakete verwendet werden sollen. (Arp ist nicht möglich mit Punkt-zu-Punkt-Geräten.)

In meisten Fällen sollten Sie alle diese Werte auf Standardwerte initialisieren, indem Sie auf "Parameter beim Gerät erfragen" klicken (nur möglich, wenn Miami offline ist). Allerdings können Sie Werte auch erzwingen, wenn das notwendig sein sollte, z.B. wenn Sie einen neuen Gerätetyp verwenden, zu dem Miami noch nicht die korrekten Standardeinstellungen kennt.

1.31 Miami_deutsch.guide/NODE_GUI_INTERFACE_INACTIVITY

Inaktivität

Einige Internet-Provider hängen auf, wenn auf der Leitung einige Zeit lang keine Aktivität ist, um zu verhindern, daß Anwender Leitungen blockieren, die sie nicht nutzen.

Mit den "Inaktivität"-Schaltern können Sie Miami so konfigurieren, daß es Leitungsaktivität simuliert, damit Ihr Provider nicht aufhängt, auch wenn Sie die Leitung gerade nicht benutzen.

Mit dem linken Schalter stellen Sie die Art der Aktivität ein: PPP ping oder ICMP ping. PPP ping benötigt weniger Bandbreite, aber funktioniert nur mit PPP, nicht mit (C)SLIP, und hat auch nicht bei allen Providern den gewünschten Effekt. ICMP ping braucht geringfügig mehr Bandbreite, aber sollte mit allen Providern funktionieren.

Wenn Sie (C)SLIP verwenden, wählen Sie ICMP ping. Ansonsten versuchen Sie es erst mit PPP ping, und wenn Ihr Provider dann immer noch aufhängt, mit ICMP ping.

Der Schalter rechts gibt die Anzahl Minuten zwischen aufeinanderfolgenden Pings an. Sie müssen damit etwas experimentieren. Typische Werte sind 9 oder 14, um ein Auflegen nach 10 oder 15 Minuten zu verhindern.

Hinweis: Sie müssen zuerst Ihren Internet-Provider fragen, ob er einen derartigen Aktivitäts-Simulator erlaubt. Einige Provider haben Nutzungsbedingungen, die den Gebrauch derartiger Simulatoren verbieten, d.h. durch den Gebrauch würden Sie die Nutzungsbedingungen verletzen. Ich lehne jegliche rechtliche oder finanzielle Verantwortung für die Folgen ab, die aus dem unberechtigten oder falschen Gebrauch des

Aktivitäts-Simulators entstehen.

Hinweis: Es gibt viele Gründe, warum ein Modem auflegen kann. Ein Grund ist ein Inaktivitäts-Timeout beim Provider, was durch diese Funktion verhindert werden sollte. Allerdings hängen manche Modems auch gelegentlich wegen Leitungsstörungen auf. Das läßt sich durch Software nicht verhindern.

1.32 Miami_deutsch.guide/NODE_GUI_PPP

```

PPP
===

PAP / CHAP-Passwort                Die 'PAP/CHAP'-Schalter

Rückruf                             Die 'Rückruf'-Schalter

VJC                                  Der 'VJC'-Schalter

ACCM                                 Der 'ACCM'-Schalter

Schnelle Neuverbindung             Der 'Schnelle Neuverbindung'-Schalter

Escape                              Der 'Escape'-Schalter

DNS mit IPCP ermitteln             Der 'DNS mit IPCP ermitteln'-Schalter

TermReq vor dem Auflegen           Der 'TermReq vor dem Auflegen'-Schalter

```

1.33 Miami_deutsch.guide/NODE_GUI_PPP_CHAP

```

PAP / CHAP-Passwort
-----

```

PAP und CHAP sind Protokolle, die von PPP verwendet werden, um die Login-ID und das Passwort an einen PPP-Server zu übermitteln.

In den meisten Fällen sind die für PAP und CHAP verwendeten Daten für Login-ID und Passwort identisch zu denen im Wählskript. In diesem

Fall aktivieren Sie "Wie im Wählmodul".

Wenn Ihr Provider für PAP/CHAP andere Daten für Login-ID und Passwort als im Wählskript benötigt, dann deaktivieren Sie "Wie im Wählmodul", und geben Sie stattdessen PAP/CHAP-Login-ID und Passwort manuell ein.

1.34 Miami_deutsch.guide/NODE_GUI_PPP_CALLBACK

Rückruf

(Diese Option ist nur in der registrierten Version anwählbar.)

PPP unterstützt Rückruf ('Callback', 'Dialback') mit dem CBCP-Protokoll. Wenn der Server Ihres Providers dafür konfiguriert ist, können Sie mit dem Provider aushandeln, daß er Sie zurück ruft, damit Sie Telefongebühren sparen.

Abhängig von der Konfiguration bei Ihrem Provider müssen Sie entweder 'CBCP, fest' wählen (in diesem Fall ruft der Provider zu einer fest vordefinierten Nummer zurück) oder 'CBCP, variabel' (in diesem Fall ruft der Provider bei der Nummer zurück, die Sie in dem Feld darunter eingeben).

'Min-Pause' ist die Wartezeit, die der Provider warten soll, bevor er Sie zurück ruft. Diese Pause sollte lang genug sein, um Ihrem Modem Zeit zu geben, die Leitung aufzulegen, und sich neu zu initialisieren.

'Max-Pause' ist die maximale Zeit, die Miami auf einen Rückruf warten soll, bevor es aufgibt.

1.35 Miami_deutsch.guide/NODE_GUI_PPP_VJC

VJC

Van Jacobsen Compression ist ein Verfahren zur Komprimierung der Header von TCP-Paketen. Diese Option sollte normalerweise angeschaltet sein, damit PPP automatisch die Benutzung von VJC aushandelt, und VJC benutzt, wenn die andere Seite zustimmt.

Allerdings unterstützen einige alte, defekte PPP-Server VJC nicht korrekt, so daß Sie VJC möglicherweise ausschalten müssen.

VJC hat keinerlei Einfluß auf die Datenkomprimierung von Ihrem Modem, d.h. Sie sollten VJC nicht abschalten, nur weil Ihr Modem MNP-5 oder V.42bis unterstützt. VJC kann unabhängig von MNP-5 oder V.42bis genutzt werden.

1.36 Miami_deutsch.guide/NODE_GUI_PPP_ACCM

ACCM

Das PPP-Protokoll unterstützt eine Liste von Kontrollzeichen, die während der Übertragung "escaped" werden müssen, d.h. die durch eine andere Zwei-Byte-Sequenz ersetzt werden. Diese Liste heisst ACCM (Asynchronous Control Character Mask).

Der Sinn dieser Liste ist es, PPP robuster auf Leitungen zu machen, die nicht vollständig 8-bit-transparent sind, und um zu verhindern, daß Software-Handshaking das PPP-Protokoll behindert.

Standardmäßig werden nur die Zeichen 17 und 19 (Xon/Xoff) escaped, so daß PPP mit Software-Handshaking genutzt werden kann. Wenn Sie PPP durch eine telnet-Verbindung betreiben, müssen Sie möglicherweise mehr Zeichen escapen. Jedes Zeichen, das in die Liste aufgenommen wird, reduziert den Datendurchsatz von PPP um ca. 0.8%.

Sie können die ACCM-Einstellungen direkt durch Eingabe einer 32-Bit-Maske ändern, oder mit Hilfe des Popup-Schalters einzelne Kontrollzeichen getrennt aktivieren oder deaktivieren.

1.37 Miami_deutsch.guide/NODE_GUI_PPP_QUICK

Schnelle Neuverbindung

Normalerweise ist es mit Miami möglich, die Verbindung zum Provider neu aufzubauen (ohne neu zu wählen), wenn das Modem noch mit der Gegenseite verbunden ist, z.B. nachdem Sie den Amiga resettet haben, aber nur wenn der "Verwende CD"-Schalter auf der "Schnittstelle"-Seite aktiv ist.

Allerdings ist mit einigen Internet-Providern und PPP auch dann keine Neuverbindung (mit Neuaushandlung von PPP) möglich, wenn die Gegenseite beim Neuaushandeln von PPP auflegt.

"Schnelle Neuverbindung" hilft dann oft: Wenn "Schnelle Neuverbindung" aktiviert ist (durch Einstellen dieser Option auf "RAM" oder "Datei"), dann versucht Miami nicht, PPP neu auszuhandeln, sondern übernimmt alle PPP-Parameter aus einem resettefesten RAM-Bereich, der vor dem Rebooten eingerichtet worden ist (für die "RAM"-Einstellung) oder von einer Datei auf Festplatte (für die "Datei"-Einstellung). In den meisten Fällen ist es so möglich, nach einem Reset die Verbindung zum Provider neu aufzubauen.

Hinweis: Wenn Sie die "Datei"-Einstellung verwenden, und Ihr Amiga crasht (aus irgendeinem Grund, z.B. ausgelöst durch ein defektes Commodity oder einen Patch) während Miami die Neuverbindungs-Datei auf Festplatte schreibt, dann ist es möglich, daß die Festplatte invalidiert oder in irgendeiner Weise beschädigt wird. Verursacht wird

das durch Fehler und Probleme im Amiga-FileSystem, nicht in Miami.

Es ist daher sicherer, "RAM" zu verwenden, weil Miami dann keine Datei auf Festplatte erstellen muss. Allerdings funktioniert die "RAM"-Einstellung nur dann, wenn entweder gar kein Reset erfolgt ist, oder wenn nur ein "warmer" Reset erfolgt ist. Wenn Ihr Amiga sehr heftig abstürzt und Sie ihn "kalt" resetten müssen (wobei resetfeste Module überschrieben werden), dann werden auch die alten PPP-Parameter überschrieben, und mit der "RAM"-Einstellung gibt es keine korrekte Neuverbindung.

1.38 Miami_deutsch.guide/NODE_GUI_PPP_ESCAPE

Escape

PPP kann mit der Gegenseite aushandeln, daß Zeichen in den Bereichen 0-31 und 128-159 escaped werden. Das wird in der ACCM konfiguriert.

Allerdings kann es auch Situationen geben, in denen andere Zeichen escaped werden müssen, z.B. 0xFF über rlogin-Verbindungen.

In diesem Fall geben Sie bitte die zweistelligen Hexcodes der zu escapenden Zeichen (getrennt von Leerzeichen) im Escape-Feld ein. Miami escaped diese Zeichen dann beim Senden.

Beachten Sie daß, im Gegensatz zur ACCM, dieses Escapen nur in einer Richtung geschieht: beim Senden. Wenn auf dem Rückkanal vom PPP-Server zu Miami ebenfalls Zeichen escaped werden müssen, muß das beim Server konfiguriert werden.

1.39 Miami_deutsch.guide/NODE_GUI_PPP_DNSIPCP

DNS mit IPCP ermitteln

Dieser Schalter ist normalerweise an. Das bedeutet, daß Miami versucht, IPCP-Erweiterungen zur automatischen Erkennung von DNS-Servern zu nutzen.

Leider unterstützen einige defekte PPP-Server diese Option nicht korrekt, lehnen sie aber auch nicht korrekt ab, sondern verletzen einfach das Protokoll. Wenn Sie Fehlermeldungen bzgl. des Link-Level-Protokolls mit Ihrem Internetprovider bekommen, müssen Sie diese Option möglicherweise deaktivieren.

1.40 Miami_deutsch.guide/NODE_GUI_PPP_TERMREQ

TermReq vor dem Auflegen

Diese Option sollte normalerweise eingeschaltet sein. Miami sendet dann eine LCP-TermReq-Nachricht an den Provider vor dem Auflegen der Leitung. Oft bewirkt das, daß der Provider zuerst auflegt, und Ihr dadurch Modem schneller auflegt, als das sonst der Fall wäre.

Allerdings gibt es einige PPP-Server, die LCP-TermReqs nicht korrekt unterstützen. Wenn Ihnen auffällt, daß das Auflegen der Leitung sehr lange dauert, dann schalten Sie diese Option ab, und versuchen Sie, ob das Auflegen dann schneller geht.

1.41 Miami_deutsch.guide/NODE_GUI_DIALER

```

=====
                Wählmodul

                Wählskript                                Die 'Wählskript'-Liste
                Telefonnummern                            Die 'Telefonnummern'-Liste
                Max Wiederholung                          Der 'Max Wiederholung'-Schalter
                Wiederholpause                            Der 'Wiederholpause'-Schalter
                Wahlpause                                  Der 'Wahlpause'-Schalter
                Lernen                                     Der 'Lernen'-Schalter
                Login-ID / Passwort                       Die 'Login-ID' / 'Passwort'-Schalter
                Speichern                                  Die 'Speichern'-Schalter

```

1.42 Miami_deutsch.guide/NODE_GUI_DIALER_SCRIPT

Wählskript

Die Liste im oberen Bereich der "Wählskript"-Gruppe enthält das Wählskript. Sie können die Einträge ändern, indem Sie sie anklicken, und dann im Textfeld darunter verändern.

Die Schalter am unteren Rand werden verwendet, um Einträge zum Wählskript hinzuzufügen, oder aus dem Wählskript zu löschen.

Für mehr Informationen zu der vom Wahlmodul verwendeten Skriptsprache sehen Sie bitte unter
Wahlmodul-Befehlssprache
nach.

Die Liste hat ein Kontextmenü, d.h. wenn Sie die rechte Maustaste über der Liste drücken, öffnet sich ein Menü, mit dem Sie das Wählskript von/nach ASCII-Format importieren/exportieren können.

1.43 Miami_deutsch.guide/NODE_GUI_DIALER_PHONE

Telefonnummern

Die "Telefonnummern"-Gruppe funktioniert ähnlich wie die "Wählskript"-Gruppe, hat aber zwei zusätzliche Schalter: "aktiv" und "deaktiv". Aktivierte Telefonnummern haben ein ">>"-Symbol neben ihnen. Nur aktivierte Telefonnummern werden beim Wählen benutzt.

In der Demoversion ist die Anzahl der Telefonnummern begrenzt auf drei. In der registrierten Version gibt es keine derartige Beschränkung.

1.44 Miami_deutsch.guide/NODE_GUI_DIALER_MAX

Max Wiederholung

Wenn keine Verbindung mit einer der angegebenen Telefonnummern aufgebaut werden kann, wartet Miami solange, wie unter

Wiederholpause

angegeben wurde, und beginnt dann wieder von vorne. Allerdings ist ↔
die

Maximalzahl der Wiederholversuche begrenzt durch die Zahl im "Max Wiederholung"-Schalter. Danach gibt Miami auf und beendet das Wahlmodul.

1.45 Miami_deutsch.guide/NODE_GUI_DIALER_DELAY

Wiederholpause

Wenn keine Verbindung mit einer der angegebenen Telefonnummern aufgebaut werden kann, wartet Miami solange, wie im "Wiederholpause"-Schalter angegeben wurde, und beginnt dann wieder von vorne.

1.46 Miami_deutsch.guide/NODE_GUI_DIALER_RDELAY

Wahlpause

Dieser Wert gibt die Pause zwischen aufeinander folgenden Wahlversuchen an (also für verschiedene Telefonnummern). Normalerweise sollte dieser Wert null sein, damit Miami sofort die nächste Nummer wählt, wenn eine Nummer besetzt ist.

Allerdings benötigen einige Europäische Modems eine Minimalpause zwischen aufeinander folgenden Wahlversuchen. Wenn Sie eines dieser Modems besitzen, müssen Sie den Wert für "Wahlpause" groß genug für Ihr Modem einstellen.

1.47 Miami_deutsch.guide/NODE_GUI_DIALER_TEACH

Lernen

Der "Lernen"-Schalter startet das Miami-Wählmodul im interaktiven Modus (d.h. ohne ein Wählskript auszuführen), speichert den gesamten Text, der vom Anwender eingegeben wird, und der vom Modem empfangen wird, und versucht dann, daraus ein geeignetes Wählskript zu erstellen.

In den meisten Fällen wird MiamiInit dazu verwendet, ein Wählskript zu erstellen, nicht "Lernen", aber wenn Ihr Provider die Anwahlprozedur irgendwann verändert, kann es bequemer sein, nur mit "Lernen" ein neues Wählskript zu erzeugen, statt mit MiamiInit eine komplett neue Konfiguration zu erstellen.

1.48 Miami_deutsch.guide/NODE_GUI_DIALER_NAME

Login-ID / Passwort

Die im Wählskript verwendeten Daten für Login-ID und Passwort. Wenn "Wie im Wählmodul" auf der PPP-Seite aktiviert ist, dann werden diese Daten auch für PAP/CHAP verwendet.

1.49 Miami_deutsch.guide/NODE_GUI_DIALER_CAPTURE

Speichern

Wenn Sie das "Speichern"-Schalter aktivieren, und im Eingabefeld einen Dateinamen eingeben, dann speichert das Wählmodul nach dem Wählen ein vollständiges Wählprotokoll in der angegebenen Datei.

1.50 Miami_deutsch.guide/NODE_GUI_DATABASE

Datenbank

=====

Die "Datenbank"-Seite entspricht den Dateien im "db"-Verzeichnis anderer Protokollstacks, d.h. sie erlaubt Ihnen, die meisten TCP-Einstellungen auf Ihrem System zu konfigurieren, z.B. welche Server gestartet werden sollen, welche Anwender das System kennt, und andere Dinge.

Der Auswahlwechsler über der Liste wird dazu verwendet, zwischen verschiedenen Teilen der Datenbank umzuschalten. Für jeden Teil der Datenbank erscheint eine Liste und ein Satz Eingabefelder, mit denen Sie den jeweils aktuellen Eintrag verändern können.

Mit dem Kontextmenü der Datenbank-Liste können Sie Teile der Datenbank von/nach ASCII-Format importieren/exportieren. Damit ist es z.B. möglich, Ihre alten db/#?-Dateien von AmiTCP/AS-225 mit Miami weiter zu verwenden.

In der registrierten Version können Sie auch Teile der Datenbank sortieren, im/exportieren von/nach Clipboard, und die Datenbank mit einer ASCII-Datei mischen.

Sie können die Reihenfolge der Einträge in der Datenbank ändern, indem Sie sie mit der Maus seitlich aus der Liste heraus ziehen, und dann wieder zurück in die Datenbank, zur gewünschten Position ziehen. Für mehr Informationen hierzu sehen Sie bitte in der MUI-Dokumentation zum Thema "Drag&Drop" nach.

Jeder Eintrag in der Datenbank kann als "temporär" markiert werden, indem man den "Temp"-Schalter anklickt. Das bewirkt, daß der entsprechende Eintrag nicht mit abgespeichert wird, wenn man die Einstellungen speichert, und daß - in einigen Fällen - der Eintrag gelöscht wird, wenn man die Verbindung zum Provider neu aufbaut. Das kann nützlich sein, wenn einige der Einträge (z.B. dynamisch

ermittelte DNS-Server-Adressen) nicht bei der nächsten Verbindung verwendet werden sollen.

Standardmäßig markiert Miami alle dynamisch ermittelten DNS-Server-Adressen und den dynamischen Hostnamen als temporär.

Teile der Datenbank:

Protokolle	Die 'Protokolle'-Seite
Dienste	Die 'Dienste'-Seite
Rechner	Die 'Rechner'-Seite
Netzwerke	Die 'Netzwerke'-Seite
Domains	Die 'Domains'-Seite
DNS-Server	Die 'DNS-Server'-Seite
InetD	Die 'InetD'-Seite
Anwender	Die 'Anwender'-Seite
Gruppen	Die 'Gruppen'-Seite
Arp	Die 'Arp'-Seite
IP-Filter	Die 'IP-Filter'-Seite

1.51 Miami_deutsch.guide/NODE_GUI_DATABASE_PROTOCOLS

Protokolle

Liste aller unterstützten Protokolle (relativ zu IP), bestehend aus einem Protokollnamen, einer Protokoll-ID, und einer optionalen Liste von Aliases.

Diese Tabelle braucht fast nie verändert werden. Sie sollten nie

Standardeinträge aus dieser Tabelle entfernen.

1.52 Miami_deutsch.guide/NODE_GUI_DATABASE_SERVICES

Dienste

Liste aller unterstützten Dienste (TCP oder UDP), bestehend aus einem Dienst-Namen, einer Dienst-ID, einem Protokollnamen, und einer optionalen Liste von Aliases.

Einige Anwenderprogramme erfordern Änderungen an dieser Tabelle. In der Regel müssen dann Dienste zu dieser Liste hinzugefügt werden. Allerdings sollten Sie nie Standardeinträge aus dieser Liste entfernen.

Insbesondere: Wenn Sie einen Dienst in InetD ausschalten wollen, sollten Sie das nicht tun, indem Sie den Eintrag aus der "Dienste"-Liste entfernen. Die korrekte Methode ist, den Eintrag aus der "InetD"-Liste zu entfernen, nicht aus der "Dienste"-Liste. Ansonsten könnten Sie später gelegentliche Fehler von anderen Programmen bekommen.

1.53 Miami_deutsch.guide/NODE_GUI_DATABASE_HOSTS

Rechner

Liste aller Rechnernamen (und zugehörigen IP-Adressen), bestehend aus einer IP-Adresse, einem Rechnernamen, und einer optionalen Liste von Aliases.

Miami fügt automatisch Einträge für "localhost" und für den Rechnernamen Ihres Amiga zu dieser Liste hinzu. Andere Einträge können manuell hinzugefügt werden, um Name->IP-Übersetzungen schneller zu machen.

1.54 Miami_deutsch.guide/NODE_GUI_DATABASE_NETWORKS

Netzwerke

Liste aller Netzwerke, bestehend aus einem Netzwerk-Namen, einer Netzwerk-ID, und einer optionalen Liste von Aliases.

Diese Tabelle wird kaum noch genutzt, und ist nur für Kompatibilität mit alter Software und Diagnosesoftware vorhanden.

1.55 Miami_deutsch.guide/NODE_GUI_DATABASE_DOMAINS

Domains

Liste aller lokalen Domains, bestehend lediglich aus dem Domainnamen.

Diese Tabelle wird eigentlich nicht von TCP/IP benötigt, aber kann den Umgang mit Miami für den Anwender bequemer machen: Sie erlaubt dem Anwender, vollständige Rechnernamen abzukürzen, indem man nur den Namen des Rechners (ohne Domain) angibt, wenn man einen Rechner ansprechen will.

Beispiel:

Angenommen ein Rechner auf dem lokalen Netz heisst ex1.foo.edu, und sie greifen häufig auf diesen Rechner zu. Wenn Sie foo.edu zur Domain-Liste hinzufügen, dann können Sie diesen Rechner auch nur mit ex1 ansprechen.

1.56 Miami_deutsch.guide/NODE_GUI_DATABASE_DNSSERVERS

DNS-Server

Liste der DNS-Server, angegeben nur durch die jeweilige IP-Adresse.

DNS-Server werden benutzt, um zu logischen Rechnernamen die dazugehörige IP-Adresse zu ermitteln. Sie sollten mindestens einen DNS-Server in dieser Tabelle eingetragen haben, vorzugsweise einen DNS-Server bei Ihrem Provider, oder zumindest einen in der Nähe.

Wenn Miami beim Aufbau der Verbindung selbst DNS-Server findet, fügt es sie selbständig in diese Liste ein, und markiert sie als "temporär".

1.57 Miami_deutsch.guide/NODE_GUI_DATABASE_INETD

InetD

Liste aller Server die vom eingebauten InetD gestartet werden, bestehend aus einem Dienst-Namen (der einem Eintrag in der "Dienste"-Tabelle entspricht), einem Socket-Typ ("dgram" oder "stream"), einem Wartemodus ("wait", "nowait" oder "dos"), dem Anwender (bei AmigaDOS normalerweise "root"), dem Dateinamen des Servers, dem Prozessnamen des Servers, und einer Liste von Shell-Argumenten, die dem Server übergeben werden.

Der in Miami eingebaute InetD unterstützt die folgenden eingebauten

Dienste: "daytime", "time", "echo", "discard", "chargen", "finger" und "auth". "auth" ist identisch zu "identd".

Server für andere (externe) Dienste können von InetD automatisch gestartet werden, wenn man einen entsprechenden Eintrag in dieser Tabelle vornimmt. Wenn Sie externe Server installieren möchten (z.B. ftpd oder telnetd), dann sehen Sie bitte in der Dokumentation des Servers nach, welches genaue Format der "InetD"-Eintrag erfordert.

1.58 Miami_deutsch.guide/NODE_GUI_DATABASE_USERS

Anwender

Liste aller Anwender im System, bestehend aus dem Anwender-Namen, dem Passwort, der Anwender-ID, der Gruppen-ID, dem Realnamen, dem Home-Verzeichnis, und dem Shell-Aufruf.

Normalerweise benötigen Sie nur einen Eintrag in dieser Datei (für Sie selbst), es sei denn, Sie verwenden Server wie ftpd oder telnetd, die anderen Anwendern Zugriff auf Ihren Amiga erlauben.

Passwörter werden in einem verschlüsselten Format gespeichert, und werden nicht direkt in der Liste angezeigt. In der Passwort-Spalte erscheint

'-'

wenn diesem Anwender kein Passwort zugeordnet ist, d.h. wenn Login ohne ein Passwort möglich ist.

'*'

wenn für diesen Anwender kein Login möglich ist.

ein zentriertes 'x'

wenn ein gültiges Passwort für diesen Anwender existiert.

Um das Passwort zu verändern, klicken Sie auf den "Passwort"-Schalter. Wenn Sie das Eingabefeld leer lassen, wird dem Anwender kein Passwort zugeordnet (angezeigt als '-'). Wenn Sie nur das einzelne Zeichen '*' eingeben, werden Logins blockiert (angezeigt als '*'). In allen anderen Fällen wird der eingegebene Text als Passwort verwendet (angezeigt als zentriertes 'x').

Hinweis: Wenn Sie die Passwort-Datei von AmiTCP importieren, dann werden die Passwörter nicht mit übernommen, d.h. die Passwörter für alle Anwender werden als "leer" gespeichert, und müssen manuell neu eingegeben werden. Das ist notwendig, weil der in AmiTCP verwendete Passwort-Algorithmus aus rechtlichen Gründen nicht in Miami verwendet werden darf. Für mehr Informationen zu diesem Thema sehen Sie bitte in

Austausch von Passwoertern
nach.

1.59 Miami_deutsch.guide/NODE_GUI_DATABASE_GROUPS

Gruppen

Liste der Gruppen im System, bestehend aus einem Gruppennamen, einer Gruppen-ID, und einer optionalen Anwender-Liste.

Normalerweise benötigen Sie nur einen Eintrag in dieser Datei (für Sie selbst), es sei denn sie verwenden Server wie ftpd oder telnetd, die anderen Anwendern Zugriff auf Ihren Amiga erlauben.

1.60 Miami_deutsch.guide/NODE_GUI_DATABASE_ARP

Arp

Liste der manuellen Arp-Einträge im System, bestehend aus einer IP-Adresse und einer Hardware-Adresse. Die Hardware-Adresse wird in der üblichen Doppelpunkt-Hex-Notation angegeben (z.B. '01:23:45').

Arp wird nur bei Bus-/Ring-SANA-II-Geräten verwendet, und Sie brauchen manuelle Arp-Einträge auch nur dann, wenn einer der Rechner in Ihrem Netz Arp nicht unterstützt.

1.61 Miami_deutsch.guide/NODE_GUI_DATABASE_IPFILTER

IP-Filter

(Diese Option erscheint nur in der registrierten Version.)

Diese Tabelle ermöglicht Ihnen, einige der bei Ihnen ankommenden IP-Pakete heraus zu filtern, oder System-Log-Einträge zu erzeugen, wenn derartige Pakete ankommen. Das erlaubt es Ihnen, einen sehr einfachen Firewall zu implementieren, oder Informationen zu erhalten, wenn jemand versucht, in Ihren Rechner "einzubrechen".

Die Tabelle besteht aus einer Folge von Regeln. Jedes ankommende Paket wird mit jeder der Regeln verglichen, in der Reihenfolge von oben nach unten. Die erste Regel, die auf das Paket zutrifft, bestimmt, ob das Paket ausgefiltert wird, und ob ein Log-Eintrag erzeugt wird. Weitere Regeln werden dann nicht mehr überprüft.

Jeder Tabelleneintrag besteht aus einem Protokoll ('tcp', 'udp' oder '*' für 'ein beliebiges Protokoll'), einem Dienst (einem Namen, der in

der Dienste-Tabelle erscheint, '*' für 'ein beliebiger Port', oder '\$' für 'ein beliebiger Service-Port', d.h. ein Port, der nicht im Bereich von 1024-5000 liegt), einer IP-Adresse (der Source-IP-Adresse des Pakets), einer Netzmaske (die den Bereich der IP-Adressen definiert), und zweier Parameter, die bestimmen, was geschehen soll: Sie können Zugriff erlauben oder verbieten ("y" oder "n"), und optional einen Log-Eintrag erstellen ("y" oder "n").

Beachten Sie, daß Log-Einträge nur für 'tcp'-Dienste erstellt werden, nicht für 'udp'-Dienste.

Hier ist ein Beispiel für eine sinnvolle Startkonfiguration für den IP-Filter:

```
* * 127.0.0.1 (leere Maske) y n
tcp auth *.*.*.* (leere Maske) y n
* $ *.*.*.* (leere Maske) y y
```

Das bewirkt:

Die erste Zeile stellt sicher, daß alle lokal verschickten Pakete (d.h. von Ihrem Amiga zu Ihrem Amiga) ohne Logging erlaubt werden.

Die zweite Zeile erlaubt auch ankommende 'auth'-Anfragen ohne Logging. Das ist nützlich, weil 'auth'- ('identd'-) Anfragen von so vielen httpd-, ftpd- und ircd-Servern gestellt werden, daß sie vermutlich nicht mit Logeinträgen für jede derartige Anfrage belästigt werden möchten.

Die dritte Zeile erlaubt alle übrigen externen Anfragen, aber erzeugt jeweils einen Log-Eintrag, damit Sie erkennen, wenn jemand versucht, auf Ihren Rechner zuzugreifen. Es ist wichtig, daß Sie den Dienst als '\$' angeben, nicht als '*', und zwar weil ftp Rückwärts-Verbindungen (also vom Server zum Client) beim Upload und Download benutzt. Wenn Sie den Dienst als '*' angeben, erhalten Sie einen Log-Eintrag jedes mal wenn Sie eine Datei von/zu einem ftp-Server up/downloaden.

Alle übrigen Pakete (d.h. Pakete von aussen, die zu einem Port zwischen 1024 und 5000 geschickt werden), verwenden die implizite Standardregel, also Zugriff erlauben, und keinen Log-Eintrag erstellen.

1.62 Miami_deutsch.guide/NODE_GUI_TCPIP

TCP/IP

=====

Hostname

Die 'Hostname'- ←
Gruppe

Realname / Username

	Die 'Realname' und 'Username' -Schalter ↔
ICMP verwenden	Der 'ICMP verwenden' -Schalter ↔
Verwende BootP	Der 'BootP-Schalter
DNS-Server überprüfen	Der 'DNS-Server überprüfen' -Schalter ↔
Pseudo-IP	Der 'Pseudo-IP' -Schalter ↔
T/TCP	Der 'T/TCP' -Schalter ↔
Domain automatisch hinzufügen	Der 'Domain automatisch hinzufügen' -Schalter ↔
'down' wenn Offline	Der ' 'down' wenn Offline' -Schalter ↔
Schutz vor Dauer-Ping	Der 'Schutz vor Dauer-Ping' -Schalter ↔
Zeit ermitteln	Die 'Zeit ermitteln' -Schalter ↔

1.63 Miami_deutsch.guide/NODE_GUI_TCPIP_HOSTNAME

Hostname

In den meisten Fällen sollten Sie den Schalter "dynamisch" einschalten. Miami ermittelt dann automatisch den Hostnamen Ihres Amiga durch "reverse DNS lookup", nachdem die Verbindung aufgebaut wurde.

Einige Provider unterstützen allerdings kein "reverse DNS lookup", oder weisen ihren Kunden feste Hostnamen zu, die nicht im DNS gelistet sind. In diesem Fall schalten Sie "dynamisch" aus, und geben Sie den Hostnamen manuell ein.

1.64 Miami_deutsch.guide/NODE_GUI_TCPIP_NAME

Realname / Username

In diesen Eingabefeldern sollten Sie Ihren Realnamen (z.B. "Joe Smith"), und den Anwendernamen auf Ihrem Amiga (z.B. "jsmith") eingeben.

Obwohl Sie theoretisch hier beliebige Namen verwenden können, ist es üblich, die korrekten Namen zu verwenden, keine Phantasienamen.

Einige Programme fragen Miami nach Anwender-Informationen zu Ihrem Anwendernamen. Damit diese Programme korrekt funktionieren, sollten Sie darauf achten, daß im "Anwender"-Teil auf der "Datenbank"-Seite ein Eintrag existiert, der dem hier angegebenen Usernamen entspricht.

1.65 Miami_deutsch.guide/NODE_GUI_TCPIP_ICMP

ICMP verwenden

Wenn dieser Schalter aktiviert ist, verwendet Miami ICMP-"ping"-Pakete, um die Gültigkeit von IP-Adressen, DNS-Servern usw. zu überprüfen.

Dieser Schalter sollte normalerweise eingeschaltet sein, weil er zusätzlich vor Fehlkonfigurationen schützt.

Allerdings kann es sein, daß Sie diesen Schalter ausschalten müssen, wenn Sie mit Hilfe eines TCP-Emulators wie TIA verbunden sind, da nicht alle TCP-Emulatoren ICMP unterstützen.

1.66 Miami_deutsch.guide/NODE_GUI_TCPIP_BOOTP

Verwende BootP

Wenn Ihr Provider dynamische IP-Adressen verwendet, gibt es verschiedene Möglichkeiten für Miami, die korrekte (dynamische) IP-Adresse zu finden.

Für PPP-Verbindungen geschieht das normalerweise als Teil des PPP-Protokolls. Für (C)SLIP-Verbindungen ist das aber nicht möglich, d.h. (C)SLIP-Verbindungen verwenden manchmal ein Protokoll mit dem Namen "BootP". Eine weitere Möglichkeit ist, die IP-Adresse aus dem Anwahl-Protokoll zu entnehmen.

Wenn Sie MiamiInit zum Konfigurieren der Leitung benutzt haben, dann lassen Sie diesen Schalter am besten in der Standardeinstellung. Wenn Sie Miami manuell konfiguriert haben, sollten Sie "BootP" zunächst eingeschaltet lassen, und dann später testen, ob es auch ohne "BootP" noch funktioniert.

Wenn Miami Ihre IP-Adresse auch ohne BootP erkennen kann, sollte "BootP" ausgeschaltet sein, da das Aufbauen der Verbindung dann schneller geht.

1.67 Miami_deutsch.guide/NODE_GUI_TCPIP_VERIFYDNS

DNS-Server überprüfen

Normalerweise versucht Miami, die IP-Adressen aller DNS-Server auf Gültigkeit zu überprüfen. Allerdings kann das Probleme mit einigen Internet-Providern verursachen, wenn deren DNS-Server eine schlechte Verbindung haben oder kurz nach Verbindungsaufbau noch nicht auf Anfragen reagieren.

Wenn Sie den Schalter "DNS-Server überprüfen" deaktivieren, dann überspringt Miami die DNS-Überprüfung beim Verbindungsaufbau.

1.68 Miami_deutsch.guide/NODE_GUI_TCPIP_FAKEIP

Pseudo-IP

Wenn Sie mit Hilfe eines TCP-Emulators wie TIA oder Slirp verbunden sind, und dieser Emulator Ihnen keine "echte" IP-Adresse zuweist, sondern eine Pseudo-IP-Adresse, dann sollten Sie diesen Schalter aktivieren.

Wenn dieser Schalter aktiviert ist, ermittelt Miami Ihren Hostnamen auf der Basis der Gateway-IP-Adresse, nicht ihrer lokalen Pseudo-IP-Adresse.

1.69 Miami_deutsch.guide/NODE_GUI_TCPIP_TTCP

T/TCP

(Diese Option ist nur in der registrierten Version anwählbar.)

T/TCP (TCP for Transactions) ist eine Erweiterung zu TCP, die die

Geschwindigkeit von Netzwerkanwendungen signifikant erhöhen kann, insbesondere für Web-Browser, wenn sowohl der Browser als auch der Server T/TCP unterstützen.

Registrierte Anwender sollten diese Option einschalten, um von dem Geschwindigkeitszuwachs zu profitieren. Allerdings haben einige wenige PPP-Server Probleme mit den erweiterten TCP-Paketen, die T/TCP erzeugt. Wenn Miami also nach dem Einschalten von T/TCP nicht mehr korrekt funktioniert, dann schalten Sie diese Option aus - oder suchen Sie sich einen neuen Provider.

1.70 Miami_deutsch.guide/NODE_GUI_TCPIP_ADDDOMAIN

Domain automatisch hinzufügen

Wenn dieser Schalter aktiviert ist, fügt Miami automatisch die Domain Ihres Hostnamen (d.h. alles hinter dem ersten '.') zu Miami's "Domains"-Datenbank hinzu.

Weder Miami noch irgendwelche Programme benötigen einen derartigen Eintrag, aber es kann für den Anwender bequemer sein, Abkürzungen für Rechnernamen zu verwenden. Für mehr Details sehen Sie unter

Die 'Datenbank'-Seite
nach.

1.71 Miami_deutsch.guide/NODE_GUI_TCPIP_DOWN

'down' wenn Offline

(Diese Option ist nur in der registrierten Version anwählbar.)

In der unregistrierten Version unterbricht Miami grundsätzlich alle TCP-Verbindungen, wenn die Schnittstelle offline geht. In der registrierten Version erhält Miami die Verbindungen in so einer Situation aufrecht. Das hat den Vorteil, daß Sie möglicherweise schnell eine neue Modem-Verbindung aufbauen, und Ihre bisherigen TCP-Verbindungen weiter nutzen können.

Der Nachteil ist, daß das Aufrechterhalten von TCP-Verbindungen bei einer Schnittstelle im Offline-Zustand auch zur Folge hat, daß eine Applikation nicht mehr erkennen kann, ob Miami online oder offline ist, d.h. wenn eine Applikation versucht, eine Verbindung aufzubauen, während die Schnittstelle offline ist, wird kein Fehler erzeugt, sondern die Applikation wartet nur auf einen Timeout.

Wenn Ihnen dieses Verhalten nicht gefällt, und Sie lieber hätten,

wenn im Offline-Zustand Fehlermeldungen erzeugt werden, dann aktivieren Sie diese Option.

1.72 Miami_deutsch.guide/NODE_GUI_TCPIP_PING

Schutz vor Dauer-Ping

(Diese Option ist nur in der registrierten Version anwählbar.)

Miami enthält eine einfache Heuristik, um die Effekte von "Denial-of-Service-Attacks" zu vermindern, die durch Dauer-Pings ("ping flooding") entstehen:

Wenn diese Option aktiviert ist, und ein Anwender versucht, Ihren Rechner mit Dauer-Pings zu belästigen (entweder mit sehr grossen Pings, oder mit sehr schnell aufeinander folgenden Pings), dann erzeugt Miami einen Log-Eintrag, um Sie davon zu informieren, und erzeugt dann für einige Zeit keine Ping-Antworten an diesen Anwender mehr, bis der Anwender für einige Zeit mit den Dauer-Pings aufgehört hat.

Beachten Sie, daß es unmöglich ist, die Dauer-Pings ganz zu verhindern, d.h. den Anwender davon abzuhalten, Ihre Modem-Kapazität zu verschwenden. Das einzige, was Miami in so einer Situation tun kann, ist, keine Antworten mehr zu schicken, und Sie zu informieren (damit Sie einen neuen Modemport anwählen können). Es ist Miami nicht möglich, die Verschwendung der Modembandbreite an sich zu verhindern. Das ginge nur durch einen Filter bei Ihrem Internet-Provider.

1.73 Miami_deutsch.guide/NODE_GUI_TCPIP_GETTIME

Zeit ermitteln

Wenn Ihr Amiga nicht mit einer batteriegepufferten Echtzeituhr ausgestattet ist, sollten Sie den Schalter "Zeit ermitteln" aktivieren, und Namen oder IP-Adresse eines Servers eingeben, der den "time"-Dienst unterstützt. Wenn Sie nicht sicher sind, welchen Server Sie eingeben sollen, versuchen Sie es mit dem Server, den Sie für Email oder News verwenden.

Wenn Sie diese Funktionen nutzen, dann achten Sie darauf, daß die Environment-Variablen "ENV:TZ" korrekt eingestellt ist, d.h. normalerweise zu einem Text der Form "EST5" oder "EST4EDT" (während der Sommerzeit). Das ist wichtig, da der Server die Zeit im GMT- (UTC-) Format übermittelt, und dieses Format von Miami an die lokale Zeitzone angepaßt werden muß.

1.74 Miami_deutsch.guide/NODE_GUI_EVENTS

Ereignisse

=====

Miami erlaubt es, auf verschiedene Ereignisse wie "offline", "online" usw. auf verschiedene Art zu reagieren, z.B. durch das Ausführen von ARexx-Skripts, das ikonifizieren des Miami-Fensters usw.

Die genauen Ereignisse, auf die Miami reagieren kann, sind:

Start

Programmstart.

Ende

Programmende.

aktives Offline

Abbauen der Verbindung, ausgelöst vom Anwender, z.B. durch klicken auf den "Offline"-Schalter, oder durch den ARexx-Befehl "OFFLINE".

passives Offline

Abbauen der Verbindung, wenn das Modem oder der Provider auflegen.

Online

Aufbauen der Verbindung, d.h. erfolgreicher Aufbau einer Verbindung zum Internet, einschließlich dem Start aller benötigten Protokolle.

erfolgloser Online-Versuch

Ein Versuch, eine Verbindung aufzubauen, ist aus irgend einem Grund fehlgeschlagen, z.B. weil alle Telefonleitungen besetzt waren, oder weil die maximale Anzahl Wiederholversuche erreicht wurde.

Miami kann in der folgenden Art reagieren. Nicht alle Optionen machen Sinn als Reaktion auf jedes Ereignis, so daß nur eine Untermenge aller Optionen für jedes Ereignis tatsächlich anwählbar sind:

ARexx

ARexx-Skript starten.

Verstecken

Miami-Fenster schliessen, und stattdessen ein Piktogramm anzeigen.

Auto-Online

Versuch, automatisch eine Verbindung aufzubauen.

Blinken

Bildschirm aufblinken lassen, oder Signalton ausgeben, wie in den Amiga-System-Preferences definiert.

Anzeigen

Miami-Fenster öffnen, und Piktogramm entfernen.

In der Demoversion von Miami ist die Option "ARexx" nicht verfügbar, und "auto-online" ist nicht als Reaktion auf "passives Offline"

verfügbar.

1.75 Miami_deutsch.guide/NODE_GUI_MODEM

===== Modem	
Init-String	Der 'Init-String'-Schalter
Exit-String	Der 'Exit-String'-Schalter
Wählpräfix	Der 'Wählpräfix'-Schalter
Wählsuffix	Der 'Wählsuffix'-Schalter
Nullmodem	Der 'Nullmodem'-Schalter

1.76 Miami_deutsch.guide/NODE_GUI_MODEM_INIT

Init-String

Der Initialisierungsstring für Ihr Modem, normalerweise eingestellt von MiamiInit.

1.77 Miami_deutsch.guide/NODE_GUI_MODEM_EXIT

Exit-String

Der String, der beim Programmende zum Modem geschickt wird. Die meisten Anwender benötigen diese Option nicht, aber sie kann nützlich sein, wenn mehrere Programme einen Modemport teilen, und das Modem auf Standardeinstellungen zurück gesetzt werden muß, wenn Miami beendet wird.

1.78 Miami_deutsch.guide/NODE_GUI_MODEM_PREFIX

Wählpräfix

Der von Ihrem Modem verwendete Befehl zum Wählen, d.h. der String, der der Telefonnummer vorangestellt wird. Normalerweise ist das "ATDT" oder "ATDP".

1.79 Miami_deutsch.guide/NODE_GUI_MODEM_SUFFIX

Wählsuffix

Der String, der an die Telefonnummer angehängt wird, um den Wählbefehl zu vervollständigen. Das ist normalerweise '\r' .

1.80 Miami_deutsch.guide/NODE_GUI_MODEM_NULLMODEM

Nullmodem

Miami nimmt normalerweise an, daß Sie ein Modem an Ihrem seriellen Port angeschlossen haben. Wenn Ihr Amiga mittels Nullmodem-Kabel direkt mit einem anderen Rechner verbunden ist, müssen Sie diesen Schalter aktivieren. Er verhindert daß Miami Modembefehle ("AT-Befehle") verschickt, und auf Antworten wie "OK" oder "CONNECT" wartet.

Wenn "Nullmodem" aktiviert ist, ändert sich die Bedeutung des "Verwende CD"-Schalters auf der "Schnittstelle"-Seite:

- * Wenn Ihr Rechner mit einem Computer verbunden ist, der eine Login-Sequenz benötigt, um SLIP/PPP zu starten, sollten Sie "Verwende CD" deaktivieren. Miami führt dann erst das auf der "Wählmodul"-Seite definierte Wählskript aus, aber ohne eine Telefonnummer zu wählen. Diese Option ist nützlich, wenn Sie mit einer Unix- oder Linux-Box verbunden sind, die einen getty mit Login/Passwort-Abfrage auf dem seriellen Port betreibt.
- * Wenn Ihr Rechner mit einem Computer verbunden ist, der auf dem seriellen Post direkt SLIP/PPP benutzt (z.B. ein anderer Amiga mit Miami), dann sollten Sie "Verwende CD" aktivieren. Miami überspringt dann das Wählskript und versucht sofort, das Protokoll auszuhandeln.

1.81 Miami_deutsch.guide/NODE_GUI_LOGGING

Protokollierung

=====

Konsole

Der 'Konsole'-Schalter

Datei

Der 'Datei'-Schalter

syslog.library verwenden

Der 'syslog.library verwenden'-Schalter

Telefon-Log

Die 'Telefon-Log'-Schalter

1.82 Miami_deutsch.guide/NODE_GUI_LOGGING_CONSOLE

Konsole

Im diesem Eingabefeld können Sie den AmigaDOS-Dateinamen eines Fensters eingeben, das Miami für System-Log-Nachrichten verwendet. Diese Datei wird offen gehalten, nachdem die erste Systemnachricht ausgegeben wurde, d.h. Sie sollten die "CON:"-Optionen "/AUTO/CLOSE" verwendet, damit Sie das Fenster schliessen können, ohne daß alte Systemnachrichten verloren gehen.

1.83 Miami_deutsch.guide/NODE_GUI_LOGGING_FILE

Datei

In diesem Eingabefeld können Sie den AmigaDOS-Dateinamen einer Datei angeben, in der Miami System-Log-Nachrichten speichert. Wenn die Datei bereits existiert, fügt Miami neue Nachrichten am Ende an, d.h. alte Dateiinhalte werden nicht gelöscht.

1.84 Miami_deutsch.guide/NODE_GUI_LOGGING_SYSLOG

syslog.library verwenden

Wenn Sie diesen Schalter aktivieren, versucht Miami, auf

syslog.library zuzugreifen, um das System-Log zu erzeugen.
 syslog.library ist Teil des SysLog-Pakets von Petri Nordlund.

1.85 Miami_deutsch.guide/NODE_GUI_LOGGING_PHONE

Telefon-Log

Miami kann Online- und Offline-Ereignisse protokollieren, um bei der Verwaltung von Telefonrechnungen zu helfen.

Mit den beiden "Telefon-Log"-Schaltern kann man das Telefon-Logging einschalten, und den Namen der Datei angeben, an der Miami neue Log-Einträge anfügt.

Im Moment wird nur das ASCII-Format unterstützt, mit den folgenden Einträgen:

```
Online: 27.07.1996 17:48:11 (5551234)
Passive offline: 27.07.1996 17:48:11
Active offline: 27.07.1996 17:48:11
Reconnect: 27.07.1996 17:48:11
```

Der "Online"-Eintrag enthält die gewählte Telefonnummer in "()".
 "Reconnect" tritt auf, wenn Miami eine Verbindung aufbaut, ohne neu zu wählen, z.B. nach einem Reset des Rechners.

Der Unterschied zwischen "passivem" und "aktivem" Offline ist, daß "aktives" Offline freiwillig ist, d.h. das Resultat eines "OFFLINE"-ARexx-Befehls, eines Klicks auf den "Offline"-Schalter usw. Ein "passives" Offline tritt auf, wenn das Modem auflegt, oder der Internet-Provider die Verbindung abbaut.

1.86 Miami_deutsch.guide/NODE_GUI_GUI

GUI (Benutzeroberfläche)

=====

Ende-Requester

Die 'Ende-Requester'-Schalter

Offline-Requester

Der 'Offline-Requester'-Schalter

Fehler-Requester

Der 'Offline-Requester'-Schalter

Wählmodul

Die 'Wählmodul'-Schalter

1.87 Miami_deutsch.guide/NODE_GUI_GUI_REQQUIT

Ende-Requester

Sie können konfigurieren, wann Miami einen Ende-Requester anzeigen soll:

- * immer
 - * wenn noch Programme laufen, die Miami benutzen
 - * wenn Miami online ist
- oder Kombinationen davon.

1.88 Miami_deutsch.guide/NODE_GUI_GUI_REQOFFLINE

Offline-Requester

Wenn Sie diesen Schalter aktivieren, fragt Miami vor dem Auflegen der Leitung noch einmal nach.

1.89 Miami_deutsch.guide/NODE_GUI_GUI_REQERRORS

Fehler-Requester

Normalerweise zeigt Miami einen Fehlerrequester an, wenn beim Wählen oder bei der Konfiguration der Verbindung ein Fehler auftritt. Wenn Sie diesen Schalter ausschalten, werden derartige Fehler ignoriert, und es erscheint kein Requester.

1.90 Miami_deutsch.guide/NODE_GUI_GUI_DIALER

Wählmodul

Das Standard-Wählfenster hat drei Teile: Einen Hilfetext oben,

einige Schalter darunter, und das Dialogfenster ganz unten. Mit den drei "Wählmodul"-Schaltern können Sie jeden dieser Teile einzeln einschalten oder ausschalten.

Wenn das Dialogfenster ausgeschaltet wird, zeigt Miami stattdessen eine einzelne Zeile an, in der der gerade ausgeführte Befehl erscheint.

Mit dem 'Fenster aktivieren'-Schalter teilen Sie Miami mit, daß Miami Wählfenster und Fehler-Requester automatisch aktivieren soll, wenn diese erscheinen.

1.91 Miami_deutsch.guide/NODE_GUI_MISC

Sonstiges
=====

Es gibt drei weitere Schalter in Miami, die noch nicht in den vorigen Abschnitten beschrieben worden sind:

- * "Online": Miami beginnt mit dem Wählen, und versucht, eine Verbindung aufzubauen.
- * "Offline": Miami bricht die Verbindung ab und legt auf.
- * Eine Liste auf der linken Seite des Miami-Fensters. Mit dieser Liste können Sie eine der Konfigurations-Seiten anwählen.

1.92 Miami_deutsch.guide/NODE_DIALERLANG

Wählmodul-Befehlssprache

Die folgenden Befehle werden vom Wählmodul unterstützt:

ABORT "Text1", "Text2", ...

Gibt eine Liste von Texten an, bei deren Empfang Miami den Wählvorgang ganz abbricht, z.B. "NO DIALTONE" vom Modem.

ASKPASSWORD

Erzeugt einen Requester, der Sie nach dem Passwort fragt.

DELAY Sekunden

Wartet die angegebene Anzahl Sekunden.

DIALNEXT "Text1", "Text2", ...

Gibt eine Liste von Texten an, bei deren Empfang Miami die nächste Telefonnummer wählt, z.B. "BUSY" vom Modem.

PARSEPASSWORD "Endzeichen"

Parsed alle vom Modem ankommenden Zeichen bis zu (nicht

einschließlich) <Endzeichen>, und ersetzt dann das aktuelle Passwort durch diesen Text. Dieser Befehl kann nützlich sein für Einmal-Passwort-Systeme, die das Passwort für die nächste Verbindung während des Login-Vorgangs senden.

REDIAL "Text1","Text2",...

Gibt eine Liste von Texten an, bei deren Empfang Miami die Leitung auflegt, und die aktuelle Telefonnummer noch einmal wählt, z.B. "BUSY" vom Modem.

SAVECONFIG

Speichert die aktuelle Konfiguration (Einstellungen) auf Disk. Dieser Befehl wird normalerweise in Verbindung mit PARSEPASSWORD benutzt, um die Einstellungen mit dem neuen Passwort zu speichern.

SEND "Text"

Sender <Text> zum Modem. Ein Linefeed/Carriage Return wird nicht automatisch angefügt. Miami erkennt die folgenden Standard-Kontrollsequenzen: \", \\, \r, \n. Ausserdem werden "\u" und "\p" unterstützt, um die aktuelle Login-ID oder das aktuelle Passwort zu senden.

SEENDBREAK

Sendet ein "Break"-Signal zum seriellen Port. Einige Terminalserver verwenden das Break-Signal, um in den Befehlsmodus umzuschalten.

SENDPAD "text",Länge

Schickt <text> zum Modem, aufgefüllt mit Leerzeichen, bis die in <Länge> angegebene Länge erreicht wird. Beispiel: 'SENDPAD "abc",5' würde "abc " senden.

SENDPASSWORD

Sendet das aktuelle Passwort, gefolgt von einem "\r".

SENDUSERID

Sendet die aktuelle Login-ID, gefolgt von einem "\r".

TIMEOUT Sekunden

Gibt an, wie lange Miami beim nächsten WAIT oder WAITPPP wartet, bevor das Wählskript abgebrochen wird.

WAIT "Text"

Wartet darauf, daß "Text" vom Modem empfangen wird.

WAITPPP

Wartet darauf, daß der Server auf PPP umschaltet.

Bei den Befehlen "ABORT", "DIAL" und "DIALNEXT" kann auch das Schlüsselwort "TIMEOUT" (ohne die Anführungszeichen) angegeben werden, statt eines Textes in Anführungszeichen, z.B.

ABORT "NO CARRIER",TIMEOUT

Das bedeutet, daß Miami das Wählskript abbricht, wenn ein Timeout auftritt. Andere Optionen sind, die aktuelle Nummer erneut anzuwählen, oder die nächste Nummer zu wählen, wenn ein Timeout auftritt.

1.93 Miami_deutsch.guide/NODE_AREXX

ARexx-Schnittstelle

Der Name des ARexx-Ports von Miami ist "MIAMI.1". Im Moment unterstützt Miami alle Standard-ARexx-Befehle für MUI-Applikationen ("QUIT", "HIDE", "DEACTIVATE", "SHOW", "ACTIVATE", "INFO", "HELP") und die folgenden zusätzlichen Befehle:

CHANGEDB

Neulesen der Datei "ENVARC:MiamiChangeDB", um die Einstellungen auf den neuesten Stand zu bringen. Bitte sehen Sie unter

Client-Einstellungen
für nähere Informationen nach.

GETONLINETIME

Gibt die Anzahl Sekunden, die Miami online ist, in der 'result'-Variable zurück.

GETSETTINGSNAME

Gibt den Dateinamen der aktuellen Konfigurationsdatei zurück.

ISONLINE

Testet ob Miami online ist, und setzt den Fehlercode ("RC") entsprechend. 1 bedeutet: Miami ist online. 0 bedeutet: Miami ist offline.

LOADSETTINGS file/a

Lädt die angegebene Einstellungen-Datei.

ONLINE

Aufbau der Verbindung. Gleiche Bedeutung wie das Anklicken des "Online"-Schalters.

OFFLINE

Abbau der Verbindung und auflegen. Gleiche Bedeutung wie das Anklicken des "Offline"-Schalters.

1.94 Miami_deutsch.guide/NODE_EXCONFIG

Austauschen von Einstellungen

Die Miami-Einstellungen (Konfigurationsdaten) werden in einer IFF-Datei gespeichert, deren Format zur Zeit absichtlich nicht dokumentiert ist. Allerdings erlaubt Miami das Importieren und Exportieren von Einstellungen auf verschiedene Art:

Format für den Vertrieb	Importieren/exportieren von ↔ Einstellungen für den Vertrieb
Austausch von Passwörtern	Austausch von Passwort-Dateien.
Client-Einstellungen	Sonder-Einstellungen für Clients

1.95 Miami_deutsch.guide/NODE_EXCONFIG_DIST

Format für den Vertrieb
=====

Miami hat eine Funktion, um Einstellungen in einem ASCII-Format zu exportieren, das für weitere Verbreitung und Vertrieb geeignet ist, z.B. zum Uploaden auf Aminet, oder zur Weitergabe an andere Anwender, die den gleichen Internet-Provider verwenden. Das Format kann auch von Internet-Providern dazu benutzt werden, vorkonfigurierte Miami-Einstellungen für neue Kunden zu erstellen.

Das ASCII-Dateiformat besteht aus einem Header, gefolgt von einer variablen Anzahl Parameter.

Beim Exportieren von Dateien nimmt Miami nur solche Parameter mit in die exportierte Datei auf, die sich auf den Provider beziehen, nicht solche, die sich auf die Systemkonfiguration des Anwenders beziehen, oder die in irgendeiner Form sicherheitsrelevant sind. Das heisst, man kann die Einstellungen gefahrlos exportieren und weitergeben, ohne daß das Risiko besteht, daß dabei geheime Informationen wie Passwörter mit heraus gegeben werden.

Beim Importieren von Dateien unterstützt Miami allerdings auch anwenderbezogene Informationen wie Passwörter, damit Provider Installationsskripte schreiben können, die den Anwender nach Login-ID, Passwort usw. fragen, und dann eine ASCII-Einstellungen-Datei schreiben, die alle von Miami benötigten Informationen enthält.

Um eine Vorstellung davon zu bekommen, wie eine derartige ASCII-Datei aussieht, exportieren Sie einfach Ihre aktuellen Einstellungen. Das allgemeine Format ist:

- * ein 2-zeiliger Header. Beide Zeilen beginnen mit einem "\$"-Zeichen. Verändern Sie diesen Header nicht.
- * Eine variable Anzahl von Zeilen, die mit einem ";" beginnen. Diese Zeilen sind Kommentare und können frei verändert werden.
- * eine variable Anzahl von Zeilen mit Parametern.

Die meisten Parameter werden jeweils auf einer einzelnen Zeile angegeben. Diese Zeilen sehen folgendermaßen aus:

PARAMETER=Wert

Einige Parameter (z.B. das Wählskript) benötigen mehrere Zeilen.
In diesem Fall wird folgendes Format verwendet:

```
PARAMETER=%  
erster Wert  
zweiter Wert  
dritter Wert  
%
```

Das heisst, daß ein einzelnes "%" einen mehrzeiligen Parameter anzeigt, und ein weiteres "%" als einziges Zeichen in einer Zeile das Ende der Liste anzeigt.

Die Parameter können innerhalb der Datei in beliebiger Reihenfolge auftreten. Sie sollten keine Annahmen machen, daß Miami die Parameter immer in einer bestimmten Reihenfolge speichert.

Liste der gültigen Parameter: Ein (m) zeigt einen mehrzeiligen Parameter an. Ein (i) bedeutet, daß der Parameter nur importiert, aber nicht exportiert wird. Ein (r) bedeutet, daß der Parameter nur in der registrierten Version unterstützt wird, und von der unregistrierten Version ignoriert wird. "(m)" und "(i)" sind nicht Teil der ASCII-Datei.

Wenn als Format "A / B" angegeben ist, bedeutet daß, das gültige Werte einzelne Zeichen sind: entweder "A" oder "B".

DEVNAME= (i)
Gerätename

UNIT= (i)
Gerätenummer

BAUD= (i)
Geschwindigkeit des seriellen Ports

PROTOCOL=
P / S (PPP oder SLIP)

FLOWCONTROL= (i)
H / S (Hardware- (RTS/CTS) oder Software- (Xon/Xoff)
Handshaking)

EOFMODE= (i)
Y / N / A (Ja / Nein / Auto)

SERMODE=
8N1 / 7E1 / 7O1

MTU=
Zahl

IPTYPE=
D / S (dynamisch oder statisch)

IP=
1.2.3.4

CD= (i)
Y / N (Verwende CD)

BOOTP=
Y / N (Verwende BootP)

INACTIVITY=
N / I / P (Inaktivitäts-Typ: nichts, ICMP, PPP)

INACTIVITYDELAY=
Minuten

PAPNAME= (i)
Username

PAPPWD= (i)
Passwort

PAPSAME=
Y / N

CALLBACKTYPE= (r)
NONE / CBCPFIXED / CBCPVARIABLE

CALLBACKPHONE= (i) (r)
phone_number

CALLBACKMINDELAY= (r)
integer

CALLBACKMAXDELAY= (r)
integer

ACCM=
000a0000

VJC=
Y / N

QUICKRECONNECT=
Y / N

DIALNAME= (i)
Login-ID

DIALPWD= (i)
Passwort

INITSTRING= (i)
Modem_Init_String

DIALPREFIX= (i)
Wähl_Präfix

DIALSUFFIX= (i)
Wähl_Suffix

DIALSCRIPT= (m)
Wähl_Skript

DIALNUMBERS= (i) (m)
Telefon_Nummern

DIALMAXREPEAT=
Max_Wiederholung

DIALREPEATDELAY=
Wiederholpause

DIALREDIALDELAY=
Wahlpause

HOSTDYNAMIC=
Y / N (Hostname dynamisch: ja / nein)

HOSTNAME= (i)
Hostname

REALNAME= (i)
Realname

USERNAME= (i)
Username

DOICMP=
Y / N

FAKEIP=
Y / N

TTCP= (r)
Y / N

DBHOSTS= (m)
Host_Datenbank

DBNETWORKS= (m)
Netzwerk_Datenbank

DBDOMAINS= (m)
Domain_Datenbank

DBDNSSERVERS= (m)
DNS_Server_Datenbank

1.96 Miami_deutsch.guide/NODE_EXCONFIG_PASSWORDS

Austausch von Passwörtern

=====

Miami erlaubt den freien Austausch von Dateien im db-Verzeichnis von Unix/AmiTCP, mit einer Ausnahme: Die passwd-Datei kann zwar importiert werden, aber dabei werden die Passwörter nicht übernommen, sondern gelöscht, d.h. die Passwörter müssen nach dem Importieren neu manuell eingegeben werden.

Der Grund dafür ist: AmiTCP (zumindest bis zur Version 4.3) verwendet den DES-Algorithmus zur Passwort-Verschlüsselung. DES ist ein kryptographisch starker Verschlüsselungs-Algorithmus, der unter die US-Export-Beschränkungen fällt. Ein Programm, das DES implementiert, darf nicht ohne individuelle Genehmigung aus den USA exportiert werden, und die US-Regierung erteilt derzeit keine derartigen Genehmigungen.

Das führt dazu, daß jeglicher Export von AmiTCP aus den USA illegal ist, einschließlich des Downloadens des AmiTCP-Archivs von einem FTP-Server in den USA zu einem Computer ausserhalb der USA. Aus diesem Grund darf AmiTCP nicht auf alle Aminet-Server upgeloadet werden, was die Verfügbarkeit von AmiTCP stark einschränkt.

Für Miami wäre die Situation noch schlimmer gewesen: da ich Miami in den USA entwickle (nicht in Finnland wie NSDi), hätte ich Miami niemandem ausserhalb der USA weitergeben dürfen, egal auf welche Art es vertrieben würde. Ich habe mich daher entschlossen, DES in Miami nicht zu verwenden, sondern einen anderen Verschlüsselungsalgorithmus, der nicht unter die US-Export-Beschränkungen fällt.

Miami verwendet eine iterierte Version von MD5 zur Passwort-Verschlüsselung. Dieser Algorithmus ist kryptographisch stark, d.h. es ist ausser der vollständigen Suche kein Algorithmus bekannt, um MD5 zu brechen, genau wie bei DES. Allerdings ist MD5, im Gegensatz zu DES, ein Einweg-Algorithmus, d.h. verschlüsselte Texte können nicht entschlüsselt werden, und deshalb fällt MD5 nicht unter die US-Export-Beschränkungen.

Das bedeutet, daß Miami frei in die USA importiert und aus den USA exportiert werden darf, auf Aminet und andere Server upgeloadet werden darf, und sowohl in den USA, als auch außerhalb der USA verwendet werden darf (es sei denn ein Land verbietet die Benutzung von MD5).

Es tut mir leid, daß Anwender deshalb möglicherweise mehrere getrennte Passwort-Dateien verwalten müssen, aber ich sehe keinen anderen Weg, dieses Problem zu umgehen.

1.97 Miami_deutsch.guide/NODE_EXCONFIG_CLIENTS

Client-Einstellungen.

=====

Einige TCP/IP-Clients (also Programme, die unter Miami laufen

sollen), z.B. Amitalk, erfordern Änderungen an den Dateien, die die meisten Protokollstacks im "db"-Verzeichnis speichern. In den meisten Fällen müssen Einträge zu den "services" oder "inetd.conf"-Dateien hinzugefügt werden.

Mit Miami können Sie die entsprechenden Änderungen direkt durch die graphische Benutzeroberfläche vornehmen, d.h. sie wählen die "Datenbank"-Seite an, wählen den korrekten Teil der Datenbank an (z.B. "Dienste"), und fügen die benötigten Einträge hinzu.

In manchen Situationen kann es bequemer sein, diesen Vorgang zu automatisieren, z.B. damit ein Installer-Skript eines TCP/IP-Clients alle benötigten Änderungen automatisch erledigen kann, ohne den Anwender damit zu belästigen. Mit Miami geht das folgendermaßen:

- * Zuerst fügen Sie eine Zeile am Ende der Datei "ENVARC:MiamiChangeDB" an, die folgendermaßen aussieht:

```
ADD services ntalk 518/udp
```

 oder

```
ADD inetd ntalk dgram udp wait root Servers:talkd (talkd)
```

 Wenn Miami gestartet wird, lädt es automatisch den Inhalt dieser Datei (wenn sie existiert), paßt die Einstellungen entsprechend an, und speichert die Einstellungen wieder ab.
- * Wenn Miami bereits läuft, wenn der Client installiert wird, und Miami die Einstellungen sofort ändern soll, dann sollten Sie Miami den "CHANGEDB" ARexx-Befehl schicken, nachdem Sie die o.a. Datei geändert haben.

Zusammenfassend: Um von einem Installer-Skript Miami automatisch für einen Client zu konfigurieren, verwenden Sie die folgenden Befehle:

```
echo >>ENVARC:MiamiChangeDB "ADD services ntalk 518/udp"
rx "address MIAMI.1;CHANGEDB"
```

Wenn Miami bereits läuft, werden die Änderungen sofort übernommen. Ansonsten übernimmt Miami die Änderungen beim nächsten Programmstart.

1.98 Miami_deutsch.guide/NODE_UTILITY

Hilfsprogramme

MiamiArp

MiamiArp

MiamiFinger

MiamiFinger

MiamiIfConfig

MiamiIfConfig

MiamiMapMBone	MiamiMapMBone
MiamiMRInfo	MiamiMRInfo
MiamiMRouted	MiamiMRouted
MiamiMTrace	MiamiMTrace
MiamiNetStat	MiamiNetStat
MiamiPing	MiamiPing
MiamiResolve	MiamiResolve
MiamiRoute	MiamiRoute
MiamiSysCtl	MiamiSysCtl
MiamiTCPDump	MiamiTCPDump
MiamiTraceRoute	MiamiTraceRoute

Andere Hilfsprogramme sind für zukünftige Versionen geplant, aber noch nicht verfügbar.

1.99 Miami_deutsch.guide/NODE_UTILITY_ARP

MiamiArp

=====

Anzeige und Kontrolle zur Adressumwandlung.

Gebrauch:

arp hostname

Zeigt den aktuellen Arp-Eintrag für <hostname> an.

arp [-n] -a

Zeigt alle aktuellen Arp-Einträge an. Wenn "-n" angegeben wird, werden alle Einträge nicht symbolisch, sondern numerisch angezeigt.

```
arp -d hostname
    Löscht den Arp-Eintrag für <hostname>.

arp -s hostname hw_addr [temp] [pub]
    Erzeugt einen Arp-Eintrag für <hostname> mit der Hardware-Adresse
    <hw_addr>. Der Eintrag ist permanent, es sei denn das Wort "temp"
    wird angegeben. Wenn das Wort "pub" angegeben wird, agiert dieser
    Rechner als Arp-Server für den angegebenen Host.

arp -f dateiname
    Liest Befehle aus der Datei <dateiname> und führt sie aus.
```

1.100 Miami_deutsch.guide/NODE_UTILITY_FINGER

MiamiFinger
=====

MiamiFinger zeigt Informationen zu System-Anwendern an.

Gebrauch: MiamiFinger [-l] [anwender][@rechnername]

Optionen sind:

-l
 Zeigt die Ausgabe im langen Format an (für nicht-lokale Rechner:
 schickt die "/W"-Option zum finger-Daemon).

Wenn kein Rechnername angegeben wird, wird "localhost" angenommen.

Wenn ein Anwender angegeben wird, werden Informationen zu diesem
Anwender angezeigt. Ansonsten werden die Standardinformationen für den
fingerd angezeigt, zu dem die Verbindung aufgebaut wird. In vielen
Fällen bedeutet das, daß allgemeine Systeminformationen und/oder eine
Liste der eingeloggten Anwender angezeigt werden.

Diese Implementierung von MiamiFinger unterstützt T/TCP für
schnellere finger-Antworten.

1.101 Miami_deutsch.guide/NODE_UTILITY_IFCONFIG

MiamiIfConfig
=====

Konfigurieren von Netz-Schnittstellen-Parametern

Hinweis: die meisten Optionen von MiamiIfConfig sollten derzeit nicht
mit Miami benutzt werden, weil Miami die Werte bereits korrekt
einstellt. Spielen Sie bitte mit diesem Hilfsprogramm nicht herum !
Sie sollten genau die Bedeutung der einzelnen Optionen kennen, bevor
Sie versuchen, Schnittstellenoptionen zu verändern.

Die wahrscheinlich einzigen nützlichen Optionen derzeit sind "up" und "down", um Zugriff auf die Schnittstelle vorübergehend zu verhindern. Beachten Sie, daß das nicht dazu führt, daß das Modem auflegt. Abgesehen von diesen Optionen sollten Sie MiamiIfConfig nur benutzen, um sich Optionen anzusehen, nicht um sie zu ändern.

Gebrauch: MiamiIfConfig interface [alias | -alias] [af [address [dest_addr]] [up] [down] [netmask mask]] [metric n] [arp | -arp] [broadcast address] [link0 | -link0] [link1 | -link1] [link2 | -link2]

interface

Derzeit entweder "lo0" oder "mi0"

alias/-alias

Die angegebene Adresse als Alias für eine existierende Adresse ansehen, d.h. die aktuelle Adresse nicht überschreiben.

af

Address-Familie: derzeit wird nur "inet" unterstützt.

address

Eine Protokoll-Level-Adresse. Für die Adress-Familie "inet" ist das eine IP-Adresse in Punkt-Notation (z.B. 123.45.67.89).

dest_addr

Die Protokoll-Level-Zieladresse. Wird nur für Punkt-zu-Punkt-Schnittstellen genutzt.

up/down

Markiert die Schnittstelle als "up" oder "down" (aktiv oder inaktiv).

netmask

Ändert die Netzmaske für die Schnittstelle.

metric

Ändert die Metrik (Priorität) für die Schnittstelle. Für einen Einzel-Schnittstellen-Stack wie Miami hat das keine Auswirkungen.

arp/-arp

Schaltet Arp für diese Schnittstelle ein oder aus. Diese Option sollte nicht mit Miami genutzt werden. Verwenden Sie stattdessen die Miami-GUI, um die Art der Adress-Umwandlung anzugeben.

broadcast

Setzt die Broadcast-Adresse für diese Schnittstelle.

linkx/-linkx

Setzt oder löscht die Link-Level-Flags 0, 1 oder 2. Diese Flags werden derzeit von Miami nicht genutzt.

1.102 Miami_deutsch.guide/NODE_UTILITY_MAPMBONE

MiamiMapMBone
=====

Multicast connection mapper

Usage: MiamiMapMBone [-d debug_level] [-f] [-g] [-r retry_count] [-t timeout_count] [starting_router]

MiamiMapMBone versucht, alle Multicast-Router anzuzeigen, die vom angegebenen Start-Multicast-Router erreicht werden können. Wenn kein Start-Multicast-Router auf der Kommandozeile angegeben wird, wird localhost angenommen.

Die Optionen haben die folgende Bedeutung:

-d debug_level

Gibt die Debug-Stufe an. Wenn die Debug-Stufe größer als der Standardwert 0 ist, werden zusätzliche Debug-Nachrichten angezeigt.

-f

Setzt die "Flooding"-Option. "Flooding" erlaubt die rekursive Suche nach benachbarten Multicast- Routern, und ist standardmäßig eingeschaltet, wenn kein Start-Router angegeben wird.

-g

Erzeugt einen Graph im GraphEd-Format.

-n

Schaltet DNS-Lookup für die Namen von Multicast- Routern aus.

-r retry_count

Setzt die Anzahl Wiederholversuche bei der Nachbar-Suche. Der Standardwert ist 1.

-t timeout_count

Gibt die Anzahl Sekunden bei der Nachbar-Suche an, bevor eine neue Anfrage gestellt wird. Der Standardwert ist 2 Sekunden.

1.103 Miami_deutsch.guide/NODE_UTILITY_MRINFO

MiamiMRInfo
=====

Zeigt Konfigurations-Informationen für einen Multicast-Router an.

Usage: MiamiMRInfo [-d debug_level] [-r retry_count] [-t timeout_count] [multicast_router]

MiamiMRInfo versucht, Konfigurations-Informationen für den angegebenen Multicast-Router anzuzeigen. Wenn kein Router angegeben wird, wird localhost verwendet.

Die Optionen haben die folgende Bedeutung:

- d debug_level
Gibt die Debug-Stufe an. Wenn die Debug-Stufe größer als der Standardwert 0 ist, werden zusätzliche Debug-Nachrichten angezeigt.
- r retry_count
Setzt die Anzahl Wiederholversuche bei der Nachbar-Suche. Der Standardwert ist 3.
- t timeout_count
Gibt die Anzahl Sekunden bei der Nachbar-Suche an, bevor eine neue Anfrage gestellt wird. Der Standardwert ist 4 Sekunden.

1.104 Miami_deutsch.guide/NODE_UTILITY_MROUTED

MiamiMRouteD

=====

IP Multicast Routing Daemon

Usage: MiamiMRouteD [-p] [-c config_file] [-d debug_level]

MiamiMRouteD ist ein Programm, daß Sie möglicherweise im Hintergrund starten müssen ("run MiamiMRouteD"), um einen Multicast-Feed zu erhalten oder weiterzuleiten. Für mehr Erklärungen hierzu siehe unten.

Die Optionen haben die folgende Bedeutung:

- p
Startet MiamiMRouteD im 'non-pruning mode'. Diese Option sollte nur zum Testen verwendet werden.
- c config_file
Gibt an, welche Konfigurationsdatei verwendet werden soll. Die Standarddatei ist "Miami;MiamiMRouteD.config".
- d debug_level
Gibt die Debug-Stufe an. Der Standard ist 0 (keine Debug-Informationen).

MiamiMRouteD ist ein sehr komplexes und leistungsfähiges Programm zum Empfang und Weiterleiten von Multicast-Feeds. Es wird mit einer getrennten Konfigurationsdatei konfiguriert, deren Format teilweise unten beschrieben wird. Da Miami allerdings nur ein einzelnes Interface unterstützt, können nur wenige Features von MiamiMRouteD sinnvoll mit Miami genutzt werden, und nur wenige Anwender benötigen MiamiMRouteD überhaupt.

Die am häufigsten verwendeten Konfigurationen sind:

- * Sie erhalten Ihren Multicast-Feed direkt von einer Schnittstelle,

die Broadcasts oder Multicasts unterstützt, z.B. Ethernet oder Arcnet. In diesem Fall starten Sie MiamiMRouted NICHT. Schalten Sie stattdessen Multicasting in Miami ein, auf der "Schnittstelle"-Seite.

- * Sie erhalten Ihren Multicast-Feed durch einen IP-Tunnel, möglicherweise per PPP von Ihrem Provider. In diesem Fall schalten Sie Multicasting in Miami für Ihr PPP/SLIP-Interface aus, konfigurieren Sie MiamiMRouted für einen Tunnel zu Ihrem Provider (siehe unten), und starten Sie MiamiMRouted, nachdem Sie Miami gestartet haben.

Die Konfigurationsdatei für MiamiMRouted ist eine standard-ASCII-Textdatei. Jede Zeile kann einen Befehl enthalten. Der einzige derzeit interessante Befehl ist "tunnel". Mit diesem Befehl können Sie einen IP-Tunnel zu Ihrem Provider konfigurieren, um Multicasts zu senden und zu empfangen. Der wichtige Teil der Syntax ist:

```
tunnel <local-addr> <remote-addr>
```

Für <local-addr> können Sie Ihre IP-Adresse oder einen Interface-Namen (für Miami immer "mi0") angeben. <remote-addr> ist die IP-Adresse des Rechners auf der anderen Seite des Multicast-Tunnels, z.B.

```
tunnel mi0 1.2.3.4
```

erzeugt einen Multicast-Tunnel zum Rechner 1.2.3.4.

1.105 Miami_deutsch.guide/NODE_UTILITY_MTRACE

MiamiMTrace

=====

Zeigt den Multicast-Pfad von einer Quelle zum Empfänger an

```
Usage: MiamiMTrace [-g gateway] [-i if_addr] [-l] [-M] [-m max_hops]
[-n] [-p] [-q nqueries] [-r resp_dest] [-s] [-S stat_int] [-t ttl] [-v]
[-w waittime] source [receiver] [group]
```

MiamiMTrace ist ein Hilfsprogramm sehr ähnlich zu MiamiTraceRoute, aber für Multicast-Adressen, nicht Unicast-Adressen. Sehen Sie bitte bei

MiamiTraceRoute

für mehr Informationen zu MiamiTraceRoute nach.

"group" gibt die zu verwendende Multicast-Adresse an. "source" und "receiver" sind IP-Unicast-Adressen, die den Start- und Endpunkt des zu verfolgenden Multicast-Pfads angeben. Wenn "group" nicht angegeben wird, wird 224.2.0.1 verwendet. Wenn "receiver" nicht angegeben wird, wird localhost angenommen.

Die Optionen haben die folgende Bedeutung:

-g gateway

Schickt die Trace-Anfrage direkt an den angegebenen Multicast-Router, statt sie per Multicasting zu verschicken. Das Gateway muß dann der last-hop-Router auf dem Pfad von der angegebenen Quelle zum Empfänger sein.

- i if_addr
Verwendet die angegebene Adresse für die lokale Schnittstelle (auf einem multi-homed host), um die Trace-Anfrage zu verschicken, und als Standard für Empfänger- und Antwort-Adresse.
 - l
Endlosschleife mit Ausgabe der Statistik für Paketrate und Paketverlust auf dem Multicast-Pfad, nach jeweils 10 Sekunden (siehe auch '-S stat_int').
 - M
Schickt die Antwort immer per Multicasting, statt zuerst Unicasting zu probieren.
 - m max_hops
Gibt die Maximallänge des Multicastpfads an, die vom Empfänger zum Sender zurück verfolgt wird. Der Standard ist 32 (unendlich für das DVMRP-Routing-Protokoll).
 - n
Gibt Adressen numerisch statt symbolisch und numerisch aus (spart einen DNS-Lookup für jeden Router auf dem Pfad).
 - q nqueries
Gibt die Anzahl Versuche pro Rechner auf dem Pfad an. Der Standard ist 3.
 - p
Wartet passiv auf Multicast-Antworten von Traces, die von anderen initiiert wurden. Funktioniert am besten auf Multicast-Routern.
 - r resp_dest
Schickt die Trace-Antwort an den angegebenen Host, statt an den Host, auf dem MiamiMTrace läuft, oder an eine andere Multicast-Adresse als die, die für diesen Zweck registriert ist (224.0.1.32). other than the one registered for this purpose (224.0.1.32).
 - s
Gibt die Ausgabe in kurzer Form an, d.h. nur mit dem Multicast-Pfad, aber ohne Statistiken.
 - S stat_int
Gibt die Pause zwischen aufeinanderfolgenden Statistiken an (Standard ist 10 Sekunden).
 - t ttl
Gibt die ttl (time-to-live, Anzahl Rechner auf dem Pfad) für Multicast-Trace-Anfragen und Antworten an. Der Standard ist 64, außer für lokale Anfragen zur "all routers"-Multicast-Gruppe. Dafür ist der Standard 1.
-

- v
"Verbose mode". Gibt Intervallzeiten beim Trace und in der Statistik an.
- w waittime
Gibt die Wartezeit auf eine Trace-Antwort in Sekunden an.

1.106 Miami_deutsch.guide/NODE_UTILITY_NETSTAT

MiamiNetStat
=====

MiamiNetStat ist ein Programm zum Anzeigen von Konfigurationsparametern und Statistiken. Es ist fast identisch in Funktionalität zu der "netstat"-Version, die mit 4.4BSD ausgeliefert wird, hat aber einige zusätzliche Funktionen zur Anzeige von Link-Level-Statistiken.

Gebrauch:

- * MiamiNetStat [-AaDnN] [-f Adressfamilie]
- * MiamiNetStat [-dimnNrs] [-f Adressfamilie]
- * MiamiNetStat [-dnN] [-] [-I Schnittstelle]
- * MiamiNetStat [-s] [-] [-L Schnittstelle]
- * MiamiNetStat [-s] [-g]
- * MiamiNetStat [-p Protokoll]

Der MiamiNetStat-Befehl zeigt symbolisch die Inhalte verschiedener netzwerkbezogener Datenstrukturen an. Es gibt verschiedene Ausgabeformate für die auszugebenden Informationen, abhängig davon, welche Optionen man angegeben hat.

Mit der ersten Befehlsform erhält man eine Liste aller aktiven Sockets für jedes Protokoll.

In der zweiten Form wird der Inhalt einer der Datenstrukturen angezeigt, entsprechend der angegebenen Option.

Mit der dritten Form zeigt MiamiNetStat Informationen zum Paketfluß auf der angegebenen Schnittstelle an.

Die vierte Form zeigt die Link-Level-Konfiguration oder (bei Angabe des "-s"-Flags) Link-Level-Statistiken für die angegebene Netzwerk-Schnittstelle an.

Die fünfte Form zeigt Informationen zu virtuellen Schnittstellen (für Multicasting) und Multicast-Routing-Statistiken an.

Die sechste Form zeigt Statistiken zum angegebenen Protokoll an.

Die Optionen haben die folgende Bedeutung:

- A
Wird mit der Standardanzeige verwendet, und zeigt die Adressen aller zu den Sockets gehörenden Protokoll-Kontroll-Blöcke an. Wird zur Fehlersuche verwendet.
 - a
Wird mit der Standardanzeige verwendet, und zeigt den Zustand aller Sockets an. Normalerweise werden die Sockets von Serverprozessen nicht angezeigt.
 - d
Wird zusammen mit einer Schnittstellenoption (i oder I) verwendet, und zeigt die Anzahl der nicht weitergeleiteten Pakete an.
 - D
Wird mit der Standardanzeige verwendet, und zeigt die Anzahl der übertragenen Bytes pro aktiver TCP-Verbindung an.
 - f Adressfamilie
Beschränkt Statistiken oder Adress-Kontroll-Block-Ausgaben auf die angegebene Adressfamilie. Nur die Adressfamilie "inet" wird derzeit erkannt.
 - g
Zeigt die Tabelle der virtuellen Schnittstellen und die Multicast-Routing-Tabelle an. Zusammen mit der '-s'-Option zeigt diese Option Multicast-Routing-Statistiken an. Beide Optionen sind nur sinnvoll, wenn MiamiMRouteD läuft.
 - I Schnittstelle
Zeigt Informationen zur angegebenen Schnittstelle an.
 - i
Zeigt Informationen zu allen konfigurierten Schnittstellen an.
 - m
Zeigt Statistiken der Speicherverwaltungsroutinen an. (Das Netzwerk-Kernel hat einen eigenen Speicher-Pool.)
 - n
Zeigt Netzwerk-Adressen als Zahlen an. (Normalerweise interpretiert MiamiNetStat Adressen, und versucht, sie symbolisch anzuzeigen.) Diese Option kann mit allen Ausgabeformaten verwendet werden.
 - N
Zeigt Netzwerk-Adressen nur dann symbolisch an, wenn der symbolische Name ohne vorhergehende DNS-Anfrage bereit steht. Ansonsten wird die Netzwerk-Adresse als Zahl angegeben. Diese Option kann mit allen Anzeigeformaten benutzt werden.
 - p Protokoll
Zeigt Statistiken zum angegebenen Protokoll an. Als Protokoll kann entweder ein Protokollname oder ein Alias angegeben werden. Eine leere Ausgabe bedeutet normalerweise, daß es keine interessanten
-

Statistiken gibt. Das Programm gibt einen Fehler aus, wenn das Protokoll nicht erkannt wird, oder es für das Protokoll keine Statistikroutine gibt.

-r

Zeigt die Routing-Tabellen an. Wenn "-s" auch angegeben wird, werden stattdessen Routing-Statistiken angezeigt.

-s

Zeigt die Statistiken für alle Protokolle an. Wenn diese Option wiederholt wird, werden Ausgaben mit einem Wert von Null unterdrückt.

In der Standardanzeige werden für aktive Sockets beide Adressen, die Längen der Sende- und Empfangswarteschlangen (in Bytes), das Protokoll und der interne Protokollzustand angezeigt. Die Adressformate haben die Form "Host.Port" oder "Netzwerk.Port", wenn die Adresse eines Sockets ein Netzwerk, aber keine Hostadresse angibt. Die Host- und Netzwerkadressen werden mit Hilfe von Informationen aus den "Hosts"- und "Netzwerke"-Datenbanken symbolisch angezeigt, wenn möglich. Wenn kein symbolischer Name für die Adresse bekannt ist, oder die "-n"-Option angegeben wurde, wird die Adresse numerisch ausgegeben, je nach Adressfamilie.

Die Schnittstellen-Anzeige gibt kumulative Statistiken bzgl. der Anzahl der übertragenen Pakete, der Anzahl Fehler, und der Anzahl Kollisionen aus. Die Netzwerk-Adresse der Schnittstelle und die maximale Paketgröße ("mtu") werden ebenfalls angezeigt.

Die Routing-Tabellen-Anzeige gibt die verfügbaren Routen und deren Status an. Jede Route hat einen Zielhost oder ein Zielnetzwerk, und ein Gateway zum Weiterleiten der Pakete. Das "Flags"-Feld gibt einige binär gespeicherte Informationen zur Route aus.

1

RTF_PROTO1 Protokoll-spezifisches Routing-Flag #1

2

RTF_PROTO2 Protokoll-spezifisches Routing-Flag #2

3

RTF_PROTO3 Protokoll-spezifisches Routing-Flag #3 (Bedeutung für TCP: Timeout für Route läuft).

C

RTF_CLONING Bei Benutzung neue Routen erzeugen

D

RTF_DYNAMIC Dynamisch (per redirect) erzeugt

G

RTF_GATEWAY Ziel nur über Zwischenrechner erreichbar

H

RTF_HOST Host-Eintrag (ansonsten für ein Netzwerk)

L

RTF_LLINFO Gültige Übersetzung von Protokoll- zu Link-Adresse

M

RTF_MODIFIED Dynamisch (per redirect) modifiziert

P

RTF_PRCLONING Route wird von Protokolls gecloned.

R

RTF_REJECT Host oder Netzwerk nicht erreichbar

S

RTF_STATIC Manuell hinzugefügt

U

RTF_UP Route benutzbar

W

RTF_WASCLONED Route wurde durch das Clonen einer anderen Route erzeugt.

X

RTF_XRESOLVE Externer Server übersetzt Protokoll- in Link-Adresse

Direkte Routen werden für jede Schnittstelle am lokalen Host erzeugt: Das Gateway-Feld für derartige Routen enthält die Adresse der Schnittstelle. Das refcnt-Feld gibt die Anzahl der aktiven Nutzer einer Route an. Verbindungsorientierte Protokolle verriegeln normalerweise eine einzelne Route für die Dauer der Verbindung, während nicht-verbindungsorientierte Protokolle die Route jedesmal neu ermitteln. Das Use-Feld gibt die Anzahl Pakete an, die über diese Route verschickt wurden. Der Interface-Eintrag gibt an, welches Interface für diese Route verwendet wird.

Mit der Option "-L" zeigt MiamiNetStat die Link-Level-Konfiguration an, z.B. den aktuellen Status der IPCP- und LCP-Unterprotokolle für PPP, für die angegebene Schnittstelle.

Mit der Optionskombination "-sL" zeigt MiamiNetStat Link-Level-Statistiken an, einschließlich Informationen zu verschiedenen Pakettypen und Prüfsummenfehlern, für die angegebene Schnittstelle.

Derzeit unterstützt Miami nur zwei Schnittstellen:

lo0

Die "local loopback"-Schnittstelle

mi0

Die PPP/(C)SLIP-Schnittstelle unter Verwendung des in Miami eingebauten Schnittstellentreibers oder der aktuellen SANA-II-Schnittstelle.

1.107 Miami_deutsch.guide/NODE_UTILITY_PING

MiamiPing

=====

Schickt Pakete an Netzwerk-Rechner, und wartet auf die Antwort.

Gebrauch: MiamiPing [-Rdfnqrv] [-c count] [-i wait] [-l preload] [-p pattern] [-s packetsize] hostname

Optionen:

-c count

Nach Senden und Empfang von <count> Paketen abbrechen.

-d

Setzt die SO_DEBUG-Option auf dem benutzten Socket.

-f

Dauer-Ping ('flood ping'). Schickt Pakete so schnell ab, wie sie zurückkommen, aber mindestens hundert mal pro Sekunde. Für jeden gesendeten Ping wird ein "." angezeigt, und für jeden empfangenen Ping ein Backspace. Damit kann man schnell erkennen, wie viele Pakete verloren gegangen sind. Hinweis: Der Mißbrauch dieser Option für "Denial-of-Service-Attacks" ist illegal.

-i wait

Wartet <wait> Sekunden zwischen aufeinanderfolgenden Paketen. Der Standard ist, eine Sekunde zu warten. Diese Option ist inkompatibel zu "-f".

-l preload

Verschickt <preload> Pakete so schnell wie möglich, und geht dann zum normalen Verhalten über.

-n

Nur numerische Ausgabe.

-p pattern

Man kann bis zu 16 Füllbytes angeben, mit denen das zu verschickende Paket aufgefüllt wird. Das ist sinnvoll, um datenabhängige Netzprobleme zu diagnostizieren. "-p ff" füllt z.B. das Paket vollständig mit binären Einsen auf.

-q

Stille Ausgabe. Nichts wird angezeigt, ausser der Statistik-Zeilen bei Start und Ende.

-R

Route protokollieren. Fügt die RECORD_ROUTE-Option in Ping-Pakete ein, und zeigt den Route-Puffer bei empfangenen Paketen an. Beachten Sie, daß der IP-Header nur Platz für neun derartige Routen hat. Viele Hosts ignorieren oder löschen diese Option.

-r

Normale Routing-Tabellen umgehen, und Daten direkt an einen direkt

angeschlossenen Host schicken. Wenn der Host nicht mit einem direkt angeschlossenen Netz verbunden ist, wird ein Fehler zurück gegeben. Diese Option kann benutzt werden, um einen lokalen Rechner durch eine Schnittstelle anzupingen, die keine Route enthält (z.B. nachdem die Schnittstelle von routed abgemeldet worden ist).

`-s packetsize`

Gibt die Anzahl Datenbytes pro Paket an. Der Standard ist 56, was zu 64 ICMP-Datenbytes führt, wenn man es mit den 8 Bytes für den ICMP-Header kombiniert.

`-v`

Ausführliche Ausgabe. Empfangene ICMP-Pakete ausser Ping-Antwort-Paketen werden mit ausgegeben.

1.108 Miami_deutsch.guide/NODE_UTILITY_RESOLVE

MiamiResolve

=====

Ermittelt zu einem Hostnamen die IP-Adresse oder umgekehrt.

Gebrauch:

MiamiResolve ip_address

Ermittelt zu ip_address den Hostnamen und alle IP-Adressen, und zeigt diese an.

MiamiResolve host_name

Ermittelt zu host_name alle IP-Adressen, und zeigt diese an.

MiamiResolve -s port_number

Ermittelt zu einer Portnummer alle zugehörigen Service-Namen, und zeigt diese an.

MiamiResolve -s service_name

Ermittelt zu einem Service-Namen die Portnummer und alle zugehörigen Service-Namen, und zeigt diese an.

1.109 Miami_deutsch.guide/NODE_UTILITY_ROUTE

MiamiRoute

=====

Manuelles Manipulieren der Routing-Tabellen.

Gebrauch: MiamiRoute [-nqv] command modifiers args

Optionen:

-n
Verhindert, daß Host- und Netznamen symbolisch angezeigt werden in der Ausgabe. (Die Umwandlung zwischen symbolischen Namen und den numerischen Äquivalenten kann recht zeitaufwendig sein, und kann das korrekte Funktionieren des Netzes erfordern. Es kann also ratsam sein, diese Umwandlung auszuschalten, insbesondere wenn man sich in der Reparatur eines Netzwerkfehlers befindet).

-q
Unterdrückt jegliche Ausgabe.

-v
Gibt zusätzliche Details aus.

Befehle:

add
Route hinzufügen

flush
Alle Routen entfernen. Seien Sie sehr vorsichtig bei der Benutzung dieses Befehls. Er entfernt auch einige von Miami's Standard-Routen. Wenn Sie das dann nicht wieder hinterher manuell reparieren, müssen Sie Miami neu starten, um normal weiter arbeiten zu können.

delete
Löscht eine einzelne Route.

change
Ändert Charakteristika einer Route (z.B. das Gateway).

get
Zeigt zu einem Ziel die zugehörige Route an.

monitor
Zeigt kontinuierlich die Änderungen an den Routing-Tabellen an, sowie fehlgeschlagene Route-Suchen und vermutete Netsplits.
Hinweis: ohne eine Implementierung von "routed" ist dieser Befehl nicht sehr nützlich.

Der MiamiRoute-Befehl wird normalerweise mit einem Einzel-Interface-Stack wie Miami nicht benötigt, und ist sehr komplex und schwierig zu benutzen. Für eine vollständige Beschreibung aller Optionen sehen Sie bitte in der BSD-Dokumentation des "route"-Befehls nach.

Die vermutlich einzige nützliche Anwendung des "MiamiRoute"-Befehls derzeit ist das Ansehen von Routen, z.B. um die Paketlaufzeiten oder MTU-Werte zu erhalten. Um das zu tun, verwenden Sie die Syntax:

```
MiamiRoute get hostname
```

Um sich die vollständige Routing-Tabelle anzusehen, verwenden Sie den Befehl "MiamiNetStat -r", nicht MiamiRoute.

1.110 Miami_deutsch.guide/NODE_UTILITY_SYSCTL

MiamiSysCtl
=====

MiamiSysCtl erlaubt Ihnen das Ansehen und Ändern einiger interner Variablen von Miami.

Gebrauch:

MiamiSysCtl [-n] variable
Zeigt den Wert von <variable> an.

MiamiSysCtl [-n] -w variable=value
Ändert den Wert von <variable> auf <value>.

MiamiSysCtl [-n] -a
Zeigt eine Liste aller Variablen.

MiamiSysCtl [-n] -A
Zeigt eine Liste aller Variablen, plus zusätzliche Informationen zu Domains, auf die nicht durch MiamiSysCtl zugegriffen werden kann.

Optionen:

-n
Zeigt nur numerische Ausgaben an.

Erläuterung aller Variablen:

net.inet.ip.forwarding/redirect
Diese Optionen haben keine Bedeutung in einem Einzel-Schnittstellen-Stack wie Miami.

net.inet.ip.ttl
Gibt die Standard-ttl (time-to-live) für von Miami verschickte Pakete an. Sollte auf dem Standardwert 64 stehen.

net.inet.ip.rtxpire/rtminexpire/rtmaxcache
Kontrolliert Timing und Größe für Route-Cloning. Sie sollten diese Werte nicht ändern.

net.inet.op.sourceroute
Kontrolliert das Verhalten bei Paketen, die eine IP-Source-Route enthalten. Diese Option ist nur wichtig für Router.

net.inet.icmp.maskrepl
Gibt an, ob Miami eine Netzmaske als Antwort auf ICMP-Masken-Anfragen verschicken soll. Wenn die Netzmaske korrekt konfiguriert ist und diese Option eingeschaltet wird, dann kann ein anderer Rechner auf dem lokalen Netz in MiamiInitSANA2 oder Miami automatisch die korrekte Netzmaske per ICMP erkennen.

net.inet.tcp.rfc1323

Schaltet die RFC1323-TCP-Erweiterungen ein. Diese Erweiterungen kollidieren mit PPP/SLIP-VJC, sollten also nur für Bus-Schnittstellen eingeschaltet werden, nicht für Punkt-zu-Punkt-Schnittstellen.

net.inet.tcp.rfc1644

Schaltet T/TCP ein.

net.inet.tcp.mssdflt

Setzt die normale maximale Segmentgröße für TCP. Wenn Ihre Interface-MTU größer ist als 552, sollten sie diesen Wert erhöhen, um Path MTU Discovery besser zu nutzen.

net.inet.tcp.rttdeflt

Diese Option kontrolliert das Zeitverhalten von TCPs Retransmission und sollte nicht verändert werden.

net.inet.tcp.keepidle/keepintvl

Diese Optionen kontrollieren das Zeitverhalten von TCPs Keep-Alive-Timer und sollten nicht verändert werden.

net.inet.tcp.sendspace/recvspace

Diese Optionen definieren die Standardgröße von TCPs Sende- und Empfangsfenstern und sollten nicht verändert werden.

net.inet.udp.checksum

Schaltet UDP-Prüfsummen bei allen abgeschickten Paketen ein. Diese Option sollte immer eingeschaltet sein.

net.inet.udp.maxdgram/recvspace

Diese Optionen kontrollieren Grenzwert für UDP-Pakete und sollten nicht verändert werden.

dns.cache.size

Kontrolliert die Größe von Miami's eingebautem DNS-Cache.

dns.cache.flush

Wenn man diesen Wert auf 1 setzt entleert Miami den eingebauten DNS-Cache.

inetd.retrytime/toomany/cntintvl/maxbuiltin

Diese Optionen kontrollieren InetD's eingebauten Schutz gegen SYN-Flooding. Wenn Sie einen sehr intensiv genutzten Web-Server auf Ihrem Rechner betreiben, sollten Sie diese Parameter erhöhen (insbesondere "toomany"), damit Clients keine Fehlermeldungen erhalten, wenn Ihr Rechner überlastet ist.

socket.maxqlen

Diese Option gibt die Länge der Socket-Verbindungs-Warteschlange für einen listen()-Parameter von 5 an. Der Standard ist 7, aber wenn Sie mit einem sehr schnellen Netz verbunden sind, und ausreichend Speicher haben, sollten Sie diesen Wert erhöhen, um die möglichen Effekte von SYN-Flooding zu reduzieren.

1.111 Miami_deutsch.guide/NODE_UTILITY_TCPDUMP

MiamiTCPDump
=====

MiamiTCPDump ermöglicht es, Netzwerk-Pakete zu filtern und dann auszugeben.

Gebrauch: MiamiTCPDump [-adflnNOqStvx] [-c count] [-F file] [-i interface] [-r file] [-s snaplen] [-T type] [-w file] [expression]

Optionen:

- a
Versuche, Netzwerk-Adressen und Broadcast-Adressen in Namen umzuwandeln.
 - c count
Beenden nach <count> Paketen.
 - d
Compilierten Packet-Matching-Code in lesbarer Form ausgeben, und dann beenden.
 - dd
Compilierten Packet-Matching-Code als Programmfragment ausgeben. dann beenden.
 - ddd
Compilierten Packet-Matching-Code als Dezimalzahlen plus Anzahl ausgeben.
 - f
Nicht-lokale Internet-Adressen numerisch ausgeben, nicht symbolisch.
 - F file
<file> wird als Eingabe für den Filter-Ausdruck verwendet. Ein weiterer Filter-Ausdruck in der Kommandozeile wird ignoriert.
 - i interface
Verwende <interface> (derzeit "lo0" oder "mi0"). Wenn nicht angegeben, sucht MiamiTCPDump auf der System-Schnittstellen-Liste nach der konfigurierten Schnittstelle (außer loopback) mit der niedrigsten Nummer. Bei gleichen Nummern wird die erste Schnittstelle verwendet. Derzeit ist das Resultat dieser Suche immer "mi0".
 - l
Zeilenpufferung für die Standardausgabe einschalten. Nützlich, wenn man sich die Daten während der Ausgabe ansehen möchte.
 - n
Verhindert, daß Adressen (d.h. Hostadressen, Portnummern usw.) in Namen umgewandelt werden.
-

- N
Verhindert, daß Domainnamen bei Hostnamen mit angegeben werden.
Z.B.: wenn man dieses Flag angibt, gibt MiamiTCPDump "nic" statt
"nic.ddn.mil" aus.
- O
Packet-Matching-Code-Optimizer nicht starten. Das ist nur
sinnvoll, wenn Sie einen Bug im Optimizer vermuten.
- q
Kurze Ausgabe. Gibt weniger Protokollinformationen aus, d.h. die
Zeilen sind kürzer.
- s snaplen
Übernimmt <snaplen> Bytes von jedem Paket statt des Standards
(68). 68 Bytes sind angemessen für IP, ICMP, TCP und UDP, aber
können Protokollinformationen von Nameservern oder NFS-Paketen zu
früh abschneiden (siehe unten). Pakete, die wegen einer zu kleinen
<snaplen> angeschnitten wurden, werden in der Ausgabe mit "[proto]"
angezeigt, wobei <proto> der Name des Protokolls ist, bei dem das
Paket abgeschnitten wurde. Beachten Sie, daß eine größere
<snaplen> die Verarbeitungszeit pro Paket erhöht, und die
effektive Puffergröße verringert. Sie sollten <snaplen> auf den
kleinsten Wert stellen, der noch alle Protokollinformationen
enthält, an denen Sie interessiert sind.
- S
Gibt absolute anstelle von relativen TCP-Sequenznummern aus.
- T type
Erzwingt, daß selektierte Pakete als Typ <type> interpretiert
werden. Derzeit bekannte Typen sind
- * rpc (Remote Procedure Call)
 - * rtp (Real-Time Applications protocol)
 - * rtcp (Real-Time Applications control protocol),
 - * vat (Visual Audio Tool),
 - * wb (distributed White Board).
- t
Uhrzeit nicht auf jeder Zeile ausgeben.
- tt
Unformatierte Uhrzeit auf jeder Zeile ausgeben.
- v
Etwas ausführliche Ausgabe. Es werden z.B. die time-to-live und
die type-of-service-Informationen in einem IP-Paket mit ausgegeben.
- vv
Noch ausführlichere Ausgabe. Es werden z.B. zusätzliche Felder
von NFS-Antwort-Paketen mit ausgegeben.
- w file
-

Speichert die Pakete in <Datei>, statt sie zu analysieren. Das erlaubt eine spätere Auswertung mit der Option "-r". Die Standardausgabe wird verwendet, wenn <file> als "-" angegeben wird.

-x

Gibt jedes Paket (ausser dem Link-Level-Header) hexadezimal aus, bis zu <snaplen> pro Paket.

<expression> gibt an, welche Pakete ausgegeben werden sollen. Wenn keine <expression> angegeben wurde, werden alle Pakete ausgegeben. Ansonsten werden nur die Pakete ausgegeben, auf die <expression> zutrifft.

Die Syntax von <expression> ist zu umfangreich und komplex, um sie hier vollständig zu dokumentieren. Für eine vollständige Beschreibung der Syntax und aller Details des Ausgabeformats sehen Sie bitte in der Dokumentation der frei erhältlichen BSD-Version von "tcpdump" nach. Hier sind einige Beispiele für gültige Ausdrücke:

"host sundown"

Gibt alle Pakete aus, die beim host "sundown" ankommen oder von dort abgeschickt werden.

"host helios and (hot or ace)"

Gibt alle Pakete aus zwischen "helios" und "hot" oder "ace".

"ip host ace and not helios"

Gibt alle IP-Pakete aus zwischen "ace" und jedem Host außer "helios".

"tcp[13] & 3 != 0"

Gibt die Start- und End-Pakete (SYN und FIN) für jede TCP-Verbindung aus.

"icmp[0] != 8 and icmp[0] != 0"

Gibt alle ICMP-Pakete aus, die nicht Echo-Requests/Replies sind (also keine Ping-Pakete).

1.112 Miami_deutsch.guide/NODE_UTILITY_TRACEROUTE

MiamiTraceRoute

=====

Gibt den Weg aus, den Pakete zu einem Rechner nehmen.

Gebrauch: MiamiTraceRoute [-m max_ttl] [-n] [-p port] [-q nqueries] [-r] [-s src_addr] [-t tos] [-v] [-w waittime] host [packetsize]

Optionen:

-m max_ttl

Gibt die maximale time-to-live (maximale Anzahl Schritte) aus für abgeschickte Pakete. Der Standard ist 30.

- n
Gibt Adressen numerisch statt symbolisch und numerisch aus (spart einen DNS-Lookup pro Gateway entlang des Pfads).
- p port
Gibt den Basis-UDP-Port an, der in Paketen verwendet wird (Standard ist 33434). MiamiTraceRoute hofft, dass niemand auf UDP-Ports <port>+nhops-1 beim Zielhost Verbindungen annimmt (damit eine ICMP PORT_UNREACHABLE-Nachricht zurückgegeben wird, um das Tracing zu beenden). Wenn jemand auf Verbindungen in diesem Portbereich wartet, kann diese Option benutzt werden, um einen ungenutzten Portbereich auszuwählen.
- q nqueries
Setzt die Anzahl Packets pro "ttl" auf <nqueries>. Standard ist drei.
- r
Normale Routing-Tabellen umgehen, und Daten direkt an einen direkt angeschlossenen Host schicken. Wenn der Host nicht mit einem direkt angeschlossenen Netz verbunden ist, wird ein Fehler zurück gegeben.
- s src_addr
Verwendet die angegebene IP-Adresse (angegeben als IP-Adresse, nicht als Hostname) als Quell-IP-Adresse in verschickten Paketen. Auf Rechnern mit mehr als einer IP-Adresse kann man so eine andere IP-Adresse erzwingen, als die von dem Interface, auf dem das Paket verschickt wird. Wenn die angegebene IP-Adresse keine gültige IP-Adresse für diesen Host ist, wird ein Fehler zurückgegeben, und nichts wird verschickt.
- t tos
Setzt den type-of-service in Paketen auf den angegebenen Wert (Standard ist 0). Der Wert muß im Bereich von 0 bis 255 sein. Diese Option kann verwendet werden um zu erkennen, ob verschiedene types-of-service zu unterschiedlichen Pfaden führen würden.
- v
Ausführliche Ausgabe. Empfangene ICMP-Pakete ausser TIME_EXCEEDED und UNREACHABLE werden ausgegeben.
- w
Gibt die maximale Wartezeit für eine Antwort an. (Standard ist 3 Sekunden).

1.113 Miami_deutsch.guide/NODE_COMPATIBILITY

Kompatibilität

Bisher funktioniert Miami mit allen AmiTCP-Clients und -Servern, mit denen es getestet wurde, mit einer Ausnahme:

Die AmiTCP-4.x-Version von "telnet" funktioniert normalerweise nicht mit Miami. Das liegt daran, daß diese "telnet"-Version einige undokumentierte Eigenschaften von "TCP:" ausnutzt, die nicht von Miami emuliert werden können.

Es gibt dafür zwei Lösungen:

- * Verwenden Sie eine andere telnet-Version, z.B. "AmTelnet", einen schönen MUI-basierten telnet-Client, erhältlich von www.vapor.com, die telnet-Version, die im Aminet in comm/tcp erhältlich ist, ein Terminalprogramm zusammen mit telser.device, oder "napsaterm" im telnet-Modus.
- * Installieren Sie die Version des "inet-handler", die AmiTCP 4.0demo beiliegt, erzeugen Sie einen entsprechenden Mountlist-Eintrag für "TCP:", und geben Sie "mount TCP:" ein bevor Sie Miami starten (wobei natürlich immer noch auf den Miami-Stack zugegriffen wird), statt die in Miami eingebaute Version von "TCP:" zu verwenden.

1.114 Miami_deutsch.guide/NODE_RESTRICTIONS

Einschränkungen

Die Demoversion hat die folgenden Einschränkungen:

- * Nach 60 Minuten legt Miami die Leitung auf. SANA-II-Verbindungen werden nach 30 Minuten unterbrochen.
 - * Die "Ereignisse"-Optionen "Auto-Online nach passivem Offline" und das Starten von ARExx-Skripten sind nicht verfügbar.
 - * Die Anzahl der Telefonnummern im Wählmodul ist beschränkt auf drei.
 - * Telefon-Logging ist nicht möglich.
 - * Die GUI-Optionen sind nicht anwählbar.
 - * Multicasting und T/TCP sind nicht aktiv.
 - * Der IP-Filter steht nicht zur Verfügung.
 - * Dauer-Ping-Schutz ist nicht anwählbar.
 - * Das Sortieren, Mischen und der Clipboard-Im/Export der Datenbank sind nicht möglich.
 - * PPP-Rückruf ist nicht anwählbar.
 - * Die Paket-Monitor-Funktion (für externe Paket-Monitore wie MiamiTCPDump) funktioniert nicht.
-

- * System-Log-Ereignisse können nicht an syslog.library exportiert werden.
- * Die Hilfsprogramme MiamiIfConfig, MiamiRoute, MiamiSysCtl und MiamiTCPDump, sowie die Libraries miamibpf.library und miamicap.library können nicht verwendet werden.
- * Die Programmdatei wurde nicht mit allen Optimier-Optionen kompiliert, und es gibt keine separate 020-Version.

1.115 Miami_deutsch.guide/NODE_HISTORY

Entwicklung

Version 2.1

siebente öffentliche Version

- * Volle Unterstützung für Multicasting Level-2 plus IGMP implementiert, und Kernel upgedated für Kompatibilität zu MRouted 3.5/3.8.
 - * MiamiMapMBone, MiamiMRInfo, MiamiMRouted und MiamiMTrace hinzugefügt.
 - * Option "-s" in MiamiResolve implementiert, um Service-Namen zu ermitteln.
 - * Optimierten IP-Prüfsummen-Code in der registrierten 020-Version von Miami implementiert, für eine deutlichere Geschwindigkeits-Steigerung bei Ethernet.
 - * Option implementiert, um die Daten für die Schnelle Neuverbindung bei PPP in eine Datei umzuleiten, statt in einen residenten RAM-Bereich.
 - * Verbesserung zu VJC implementiert, wie von der IETF-PPP-Arbeitsgruppe vorgeschlagen.
 - * Diverse Optimierungen in der Task-Kommunikation, um Massendatentransfer zu beschleunigen.
 - * Eine Bramo/Petersen-TCP-Korrektur hat sich nicht mit FreeBSD-TCP-Änderungen vertragen. Nach dem Beheben dieses Problems ist der Durchsatz auf Ethernet um ca. 80% angestiegen.
 - * "dead code" entfernt.
 - * Diverse Compiler-Optimieroptionen für schnellere, kürzere 000/020-Programmdateien aktiviert.
 - * Umfangreiche Änderungen am Interface-Code in Vorbereitung für Miami Deluxe.
-

- * Timing zwischen dem Senden von Massendaten und interaktiven Daten verbessert. Telnet-Verbindungen sollten jetzt während ftp-Uploads keine Timeouts mehr verursachen.
- * Scrollcode für das Wählfenster im Simple-Refresh-Mode verbessert.
- * Diverse kleinere Korrekturen an der GUI, z.B. fehlende Hilfe-Blasen.

Version 2.0n

Zwischenversion

- * Bug im Ausgabecode für Punkt-zu-Punkt-Schnittstellen beseitigt, der dazu führen konnte, daß Miami bei einigen Devices (z.B. magplip.device) Pakete 'verschluckt'. Die Probleme mit iwan.device sollten jetzt auch beseitigt sein.

Version 2.0m

Zwischenversion

- * "draser.device" zur Liste der bekannten Devices in MiamiInit hinzugefügt.
- * Kompatibilität der BootP-Implementierung in MiamiInitSANA2/Miami mit einigen BootP/DHCP-Servern verbessert. Das sollte mit einigen Kabelmodem/Ethernet-Setups helfen.
- * Bug in der Xon/Xoff-Handhabung mit PPP beseitigt.

Version 2.0l

interne Version

Version 2.0k

Zwischenversion

- * Autokonfiguration von RArp und Arp über 1201-Arcnet hat nicht mit NetBSD funktioniert - behoben.
- * Wegen eines Compiler/Optimizer-Bugs ist der WaitSelect-Aufruf manchmal zu früh zurückgekehrt, wenn ein Timeout von weniger als einer Sekunde angegeben wurde - behoben.

Version 2.0j

Zwischenversion

- * Auf der 'Ereignis'-Seite haben alle Cycle-Gadgets nun eine feste Breite, um mehr Platz für die anderen Gadgets zu schaffen.
- * Workaround, um kompatibel mit der defekten Rückruf-Implementierung in einigen Win-NT-4.0-Servern zu sein. Dieser Workaround sollte gelegentliche Fehlermeldungen bzgl. des "Link-Level-Protokolls" nach einem Rückruf beseitigen.

Version 2.0i

Zwischenversion

- * Größe des Username-Felds in der Datenbank von 40 auf 128 erhöht.
-

- * Miami sollte bei Angabe von 0.0.0.0 als Gateway nicht mehr crashen.
- * Miami und MiamiInit sollten jetzt schneller auf Benutzereingaben reagieren, wenn das Modem Miami mit Daten überflutet.
- * MiamiInit beginnt jetzt bei A2232-Karten immer mit 19200 bps.
- * Schreibfehler in deutschem Catalog von MiamiInitsSANA2 beseitigt.
- * Fehler behoben, der dazu führen konnte, daß "auth"-Prozesse sich aufhängen, und Miami dann beim Beenden hängen bleibt.
- * Weitere Änderung am PPP-Code, um zu defekten oder falsch konfigurierten Win-NT-Servern kompatibel zu sein.
- * Noch einmal kleinere Änderungen, um die Kompatibilität mit iwan.device zu verbessern.

Version 2.0h

Zwischenversion nur für MiamiInitsSANA2

- * Kleinere Änderungen, um Fehler in iwan.device zu umgehen.

Version 2.0g

Zwischenversion

- * Kleinere Anpassungen an MiamiInitsSANA2 und Miami. Vielleicht funktionieren iwan.device und ciwan.device jetzt - vielleicht aber auch nicht...
- * MiamiAirport konnte crashen, wenn eine SANA-II-Verbindung angehalten wurde, nachdem eine Einstellungsdatei für eine serielle Verbindung geladen wurde - behoben.
- * Einige ToolTypes haben seltsamerweise nicht funktioniert, wenn Miami von Workbench gestartet wurde - sollte behoben sein...
- * Catalog-Versionsnummern auf 7 erhöht (Vorbereitung für Übersetzungen).

Version 2.0f

Zwischenversion

- * MiamiTraceRoute ist auf 68000/010-Rechnern gecrasht - behoben.
 - * Ein weiterer Fall von 'Hängen beim Quit' behoben, der aufgetreten ist, wenn Miami beendet werden sollte, während ein Programm Daten zum TCP: Dos-Gerät geschickt hat.
 - * Option IDEBUG in MiamiInitsSANA2 implementiert.
 - * fossil.device wieder von der Liste der EOF-Mode-kompatiblen Treiber entfernt.
 - * Verbesserung im Workaround, um bei T/TCP-Paketen mit defekten Linux-Servern zurecht zu kommen. Miami schickt jetzt
-

SFP-Pakete nur noch an die Rechner, die zuvor einer TAO-Aushandlung schon zugestimmt haben. Wenn Sie mit älteren Miami-Versionen und T/TCP Probleme hatten, probieren Sie es bitte jetzt noch einmal.

- * NSD-Überprüfungen in Miami, MiamiInit und MiamiInitsANA2 implementiert, wenn NSDPatch läuft.
- * Nach einem PPP-Callback konnte es passieren, daß Miami versehentlich Teile der PPP-Neuinitialisierung überspringt, was dann zu Enforcer/Poolwatch-Hits und anderen Problemen geführt hat - behoben.

Version 2.0e

Zwischenversion

- * Kleinere Änderungen in InetD.
- * MiamiNetStat hat die Ausgabe nach einem Ctrl-C neu begonnen - behoben.
- * ToolType "DISABLECHAP" implementiert, um für Provider mit defekter PPP-Config manuell die CHAP-Unterstützung abzuschalten. Sigh...
- * Das AppIcon unterstützt nun Drag&Drop für Einstellungen-Dateien.
- * Miami reagiert nun wesentlich schneller auf Klicks auf den "Abbrechen"-Schalter, wenn das Modem initialisiert wird.
- * Viele Änderungen im Cleanup-Code. 'Hänger' wegen 'aufgehängter' Clients sollten jetzt nicht mehr auftreten.

Version 2.0d

Zwischenversion

- * Fehler in MiamiPing behoben, der dazu führen konnte, daß MiamiPing in eine Endlosschleife bei der Ausgabe der Statistiken läuft.

Version 2.0c

Zwischenversion

- * Alle eingebauten InetD-Dienste verwenden nun einen 2-Minuten-Timeout beim Empfangen und Senden, um zu verhindern, daß bössartige Clients Ressourcen unbegrenzt blockieren.
- * Stackgröße von "Miami Airport" erhöht. Das sollte die Crashes verhindern, über die einige Anwender nach einem PPP-Rückruf berichtet haben.

Version 2.0b

Zwischenversion

- * MiamiArp wieder mit ins Archiv gelegt.
 - * Kleinere Fehler im Installer-Skript behoben.
 - * Die "nächste wählen"-Funktion hat nicht korrekt
-

funktioniert, wenn man eine "Wahlpause" ungleich null angegeben hat.

- * Einige weitere Funktionen in der Im/Export-Liste des Vertriebs-Dateiformats hinzugefügt.
- * "nächste wählen" und einige andere Funktionen konnten Crashes verursachen, wenn sie innerhalb von "Lernen" aktiviert wurden - behoben.
- * Eine "Wahlpause" ungleich null hat dazu geführt, daß "Wiederholpause" ignoriert wurde - behoben.

Version 2.0a

Zwischenversion

- * Fehler in einem der Icons behoben.

Version 2.0

sechste öffentliche Version

- * Unterstützung für T/TCP im Kernel implementiert.
 - * Unterstützung für T/TCP in den IdentD, Daytime und Time-Diensten im eingebauten InetD implementiert.
 - * SANA-II-Unterstützung für Punkt-zu-Punkt-Geräte und Bus-Geräte implementiert.
 - * Unterstützung für Level-1-Multicasting implementiert.
 - * Bug in MiamiNetStat behoben, in der Art, wie zu IP-Adressen der Hostname ermittelt wurde.
 - * Miami hat das letzte Zeichen aus der PPP-"escape"-Liste ignoriert - behoben.
 - * Miami zeichnet nun Textframes um die Programmseiten herum.
 - * Der eingebaute DNS-Cache verwendet nun das TTL-Feld als Timeout für Cache-Einträge.
 - * Kleine Änderung im endlichen Automaten für PPP, um das Zeitverhalten bei statischen IP-Adressen zu verbessern, wenn PAP/CHAP benutzt wird.
 - * Option "'down' wenn Offline" hinzugefügt, um die Schnittstelle und die Routen als 'down' zu markieren, wenn Miami offline geht.
 - * Datagrams mit einer Größe zwischen (Windowgröße-35) und (Windowgröße) wurden nicht korrekt verschickt - behoben.
 - * Neu kompiliert mit SAS/C 6.57.
 - * "NO DIALTONE" als Synonym für "NO DIAL TONE" hinzugefügt.
 - * Miami fügt nun einen abschließenden '/' an das Home-Verzeichnis des Anwenders an, wenn notwendig.
-

- * Es gab eine Inkompatibilität zwischen 4.4BSD und 4.3BSD in der Art, wie Socket-Flags gehandhabt wurde, was dazu führte, daß AWS nicht korrekt funktioniert hat - behoben.
 - * Inkompatibilität zwischen dem DNS-Resolver und einigen Versionen von named behoben, die dazu führen konnte, daß alle unbekannt Hostnamen in eine pseudo-zufällige IP-Adresse "resolved" wurden.
 - * Seit Version 1.1 hat Miami fälschlicherweise auch in der registrierten Version TCP-Verbindungen beim Auflegen abgebrochen - behoben.
 - * Der eingebaute IdentD gibt nun den Anwendernamen nur bei solchen Verbindungen an, die wirklich existieren, und gibt "UNIX" als Betriebssystem zurück. Das ist notwendig für Kompatibilität mit neueren Versionen von ircd. IdentD sollte nun voll RFC-compliant sein.
 - * Das Ändern eines internen InetD-Eintrag in einen externen Eintrag hat nicht immer funktioniert, wenn Miami online war - behoben.
 - * Wählmodul-Befehle sind jetzt unabhängig von Gross/Kleinschreibung, wie es immer schon hätte sein sollen.
 - * MiamiRegister speichert jetzt die Registrierdatei auch dann, wenn sie korrekt verschickt wurde - für den Fall, daß sie später noch einmal übertragen werden muß.
 - * Path-MTU-Discovery im Miami-TCP/IP-Kernel implementiert.
 - * Standard-TCP-Window-Größe von 8kB auf 16kB erhöht, wie bei den meisten anderen neueren 4.4BSD-Implementierungen.
 - * Das Busy-Window hat nun eine MUI-ID, so daß die Snapshot-Funktion benutzt werden kann.
 - * Option "-D" zu MiamiNetStat hinzugefügt, um die Datenmenge anzuzeigen, die über eine bestehende TCP-Verbindung geschickt wurde.
 - * Fehler im eingebauten InetD behoben: FingerD und InetD hatten nicht immer einen korrekten Cleanup ausgeführt, wenn der Client die Verbindung unterbrochen hat, ohne Daten zu senden. Dieser Fehler war bereits teilweise in 1.1d behoben.
 - * MiamiInit fügt nun ein "\r" an das Ende des Modem-Init-Strings an, wenn der Anwender das vergißt.
 - * Parameter "Wahlpause" hinzugefügt, um die zwangsweise Pause zwischen aufeinanderfolgenden Wahlversuchen zu erlauben, die von einigen europäischen Modems benötigt wird.
 - * Bei ausgeschaltetem Wahldialog zeigt das Wahlfenster nun auch die Nachrichten "Wiederholpause" und "Wahlpause" an.
-

- * Hintergrund im Demo-Requester auf den MUI-Text-Hintergrund geändert.
 - * Für SLIP war die Anzahl der in NetStat ausgegebenen IP-Pakete immer null.
 - * Die "lastlog"-Zeit in usergroup.library ist auf Lokalzeit anstelle von GMT gestellt worden, was dazu geführt hat, daß "finger" eine falsche Zeit ausgegeben hat - behoben.
 - * Ein Fehler in getpwent() hat dazu geführt, daß "finger name" nicht funktioniert hat.
 - * Kleiner Fehler in PPP-LCP beseitigt: Miami hat manchmal ein überflüssiges NAK für 'Magic Number' verschickt.
 - * Datenkontrolle im Ausgabekanal von SLIP/PPP verbessert.
 - * MiamiPing zum Archiv hinzugefügt.
 - * MiamiTraceRoute zum Archiv hinzugefügt.
 - * MiamiSysCtl zum Archiv hinzugefügt.
 - * MiamiIfConfig zum Archiv hinzugefügt.
 - * MiamiRoute zum Archiv hinzugefügt.
 - * Miami setzt die MTU-Werte nicht mehr auf die Standardwerte von 1500 oder 1006 zurück, wenn das Protokoll verändert wird.
 - * Miami zeigt den "initialisiere Modem"-Requester nicht mehr an, wenn eine existierende Verbindung fortgesetzt wird, oder eine Nullmodem-Verbindung verwendet wird.
 - * Das Auflegen ist nun etwas schneller.
 - * "fossil.device" zur Treiberliste in MiamiInit hinzugefügt, und EOF-Mode dafür aktiviert.
 - * EOF-Mode=auto sollte nun auch korrekt mit A2232-Karten und allen möglichen Unit-Zuordnungen funktionieren.
 - * Der PPP-Empfänger löst jetzt einen VJC-Restart aus nach dem Empfang eines defekten Pakets, um TCP-Sequenz-Fehler zu vermeiden, wenn Pakete fehlen, und das ID-Feld komprimiert ist.
 - * Heuristik fuer Puffer-Alignment in PPP und SLIP verbessert. Das sollte in vielen Fällen die CPU-Belastung verringern.
 - * Fehler beseitigt: Beim Wählen wurde manchmal nicht der komplette Wählstring in das Wählfenster geechoet.
 - * Unterstützung für GetSocketEvents() wieder aktiviert.
-

- * 68020-Version von Miami (nur für reg. Anwender) dem Archiv beigelegt.
 - * Kleine Änderung in der Art, wie PAP/CHAP-Fehlermeldungen angezeigt werden.
 - * Modem-Init/Exit-Strings können jetzt bis zu 80 Zeichen lange sein.
 - * Kleine Änderung am Rendering-Code, um ein mögliches Refresh-Problem mit simple-refresh-Fenstern in OS 3.x zu beseitigen.
 - * Kleiner Fehler beseitigt in der Art, wie ENV:TZ bei der Europäischen Sommerzeit genutzt wird.
 - * Workaround für defekte Win-NT-Server wieder geändert, um solche Win-NT-Server zu unterstützen, bei denen PAP ausgeschaltet ist.
 - * Unterstützung für vorgeschlagene Erweiterung zur IPCP/Option IP_ADDRESS implementiert.
 - * PPP-Authentication-Phase ist nun formell getrennt von der NCP-Config-Phase. Das sollte beim Workaround für Bugs in Win-NT-Servern helfen.
 - * Option "-N" zu MiamiNetStat hinzugefügt, um die gecacheten Hostnamen anzuzeigen, aber zusätzliche DNS-Lookups zu verhindern.
 - * Puffergröße für serial.device von 4 kB auf 16 kB erhöht.
 - * Miami speichert die Position der Balancing Bar im Wählfenster jetzt mit ab.
 - * ToolType "AREXX" hinzugefügt, um ein AREXX-Skript beim Start zu aktivieren.
 - * Syslog-Meldungen enthalten jetzt System-Datum und -Zeit.
 - * AREXX-Befehle GETONLINETIME und LOADSETTINGS implementiert.
 - * ToolType ACCMBUG hinzugefügt, um einen Bug in einigen Xylogics-Servern zu umgehen (z.B. bei den Server, die InternetFCI verwendet).
 - * "MiamiResolve" und "MiamiFinger" dem Archiv hinzugefügt.
 - * Miami fragt nun den Anwender, ob eine Einstellungsdatei von MiamiInit oder MiamiInitSANA2 importiert werden soll, und speichert dann die Einstellungen nach dem Start, wenn noch keine Einstellungen-Datei existiert.
 - * Wählbefehle ASKPASSWORD und SENDPAD implementiert.
 - * Datenbank-Seite neu designed, um sie intuitiver in der
-

Nutzung zu machen.

- * Fehler im eingebauten InetD beseitigt: FingerD und IdentD konnten in manchen Fällen ihre internen Puffer überschreiben, und Crashes verursachen.
- * Simple Heuristik zum Dauer-Ping-Schutz implementiert.
- * miamibpf.library, miamipcap.library und MiamiTCPDump dem Archiv beigelegt.
- * Menu-Help für das Hauptmenü implementiert.
- * Einzel-Eintrag-Cache für TCP und UDP PCBs in hashlistenbasierten Multi-Eintrag-Cache erweitert.
- * Fehlende "Standard"-MUI-ARexx-Befehle implementiert.
- * Callback-Hook-Mechanismus für Paket-Monitore implementiert.
- * Unterstützung für Shellskripte auf der Ereignisse-Seite implementiert.
- * Verbesserungen in WaitSelect() um die Kompatibilität mit anderen Protokollstacks zu erhöhen.
- * Konfigurierbaren IP-Filter implementiert.
- * Automatischen IP-Filter für ungültige IP-Adressen (wie 0.*.*.*) implementiert.
- * Option "Konsolename" auf der Ereignisse-Seite implementiert.
- * Unterstützung für Petri Nordlund's SysLog-Paket implementiert.
- * Unterstützung für CBCP-PPP-Rückruf implementiert.
- * Einige Verbesserungen im endlichen Automaten von PPP.
- * Menüeintrag "Iconifizieren" implementiert.
- * Viele kleinere Änderungen, die ich schon wieder vergessen habe :-).

Version 1.1d

Zwischenversion

- * Fehler behoben, der dazu führen konnte, daß AuthD/IdentD in einigen Fällen sehr viel CPU-Zeit verbraucht hat.

Version 1.1b

Zwischenversion

- * Mehr als drei Telefonnummern wurden in der registrierten Version nicht immer korrekt unterstützt - behoben.

Version 1.1a

Zwischenversion

- * Die meisten Einträge in MiamiInit waren "um einen daneben". Behoben.

Version 1.1

fünfte öffentliche Version

- * Der Fehler, daß manchmal der ARexx-Befehl "QUIT" nicht funktioniert hat, wenn die Benutzeroberfläche ikonifiziert war, sollte behoben sein.
- * Wenn der Wählerdialog ausgeschaltet war, hat das Text-Clipping im Wählerfenster nicht immer korrekt funktioniert - behoben.
- * Einige zusätzliche Modem-Definitionen in MiamiInit aufgenommen.
- * Einige MUI-Hintergrundmuster in Miami korrigiert.
- * Das Wählerfenster hat jetzt einen zusätzlichen Rahmen von 1 Pixel Breite, damit Zeichensätze mit ungewöhnlichen Dimensionen korrekt dargestellt werden.
- * Kleiner Grafikfehler behoben, der auftreten konnte, wenn der Wählerdialog ausgeschaltet war.
- * MiamiInit versucht nun, die korrekte MTU zu ermitteln, und diese an Miami zu übergeben.
- * Ein weiteres Speicherleck beseitigt.
- * Designänderung auf der "Allgemeines"-Seite.
- * MiamiRegister kann nun auch dann von Miami gestartet werden, wenn Miami von der Shell mit Angabe eines Pfades gestartet wurde.
- * Zwei kleine Änderungen am TCP/IP-Kernel zur Leistungsverbesserung.
- * Warnmeldung eingebaut, wenn eine alte Einstellungsdatei mit unvollständiger Datenbank geladen wird (wie Miami 1.0.1 und 1.0.2 sie erzeugt hat).

Version 1.0.3

vierte öffentliche Version

- * Änderung im DNS-Resolver: Bisher hat Miami bei Hostnamen, die nicht in einem '.' endeten, zuerst angenommen, daß es Abkürzungen sind, und jeweils die definierten Domainnamen zum DNS-Lookup angehängt, bevor ein DNS-Lookup nur mit dem Hostnamen durchgeführt wurde. Jetzt, nach der Änderung, wird jeder Hostname, der mindestens ein '.' enthält, als "fully qualified" angesehen, und DNS-Lookups werden erst nur für diesen Hostnamen durchgeführt. Erst wenn das erfolglos ist, werden zum DNS-Lookup Domains angehängt. Diese neue Strategie ist anders als die normalerweise von BSD verwendete, aber scheint DNS-Lookups insgesamt zu beschleunigen, vor allem, wenn Abkürzungen selten benutzt werden, und für mindestens eine vordefinierte Domain ein

- DNS-Lookup recht langsam sein kann. Einfacher formuliert: Die "looking up host name"-Phase von Webbrowsern sollte jetzt kürzer sein als vorher, besonders wenn mehr als eine Domain bei Miami konfiguriert ist.
- * Kleiner Bugfix im "daytime"-Service in Bezug auf ENV:TZ.
 - * GUI-Konfigurationsseite implementiert. Auf dieser Seite kann die Benutzeroberfläche genauer konfiguriert werden.
 - * Der "Ende"-Requester ist jetzt konfigurierbarer, ein "Offline"-Requester ist neu hinzugekommen, und alle Fehlerrequester für Protokollfehler können abgeschaltet werden.
 - * Die Hilfe-Texte und Schalter im Wählmodul können nun getrennt abgeschaltet werden, und es ist möglich, das Wählfenster durch ein kleines, einzeiliges Statusfenster zu ersetzen, in dem nur der gerade ausgeführte Befehl angezeigt wird.
 - * Fehlercodes von gethostbyname() wurden nicht korrekt in errno und herrno zurückgegeben.
 - * "Escape"-Funktion auf der PPP-Konfig-Seite neu implementiert, um andere Zeichen als 0-31 und 128-159 zu escapen.
 - * Das Ausführen des ARexx-Befehls "QUIT" von einem von Miami aus gestarteten ARexx-Skript ist jetzt korrekt möglich.
 - * syslog() hat nicht korrekt mit allen Programmen funktioniert.
 - * syslog() protokolliert jetzt die Prozess-ID mit, wenn ein Programm das verlangt.
 - * Wenn Miami ohne Einstellungsdatei gestartet wurde, ist die Datenbank nicht korrekt initialisiert worden.
 - * Unterstützung für den InetD-Server-Typ "dos" implementiert.
 - * Das Hauptfenster ist nun ein AppWindow, d.h. wenn man ein Icon einer Einstellungsdatei auf das Fenster zieht, dann lädt Miami die zugehörige Datei.
 - * Wenn Miami von der Workbench gestartet wird, wird jetzt zunächst die Datei "ENV:Sys/def_MiamiApp.info" als AppIcon benutzt. Nur wenn dieses Icon nicht existiert oder Miami von der Shell gestartet wurde, verwendet Miami das eingebaute AppIcon.
 - * Es gab einen weiteren Fall, in dem das Speicherbelegungsproblem von Miami 1.0 auch noch in 1.0.2 auftrat.

Version 1.0.2

dritte öffentliche Version

- * Nur kleinere Änderungen an MiamiRegister, um den

Miami-Registrier-Server über den neuen Hostnamen anzusprechen.

Version 1.0.1

zweite öffentliche Version

- * Abfrage um einige Probleme mit einigen Annex-Terminal-Servern zu umgehen.
- * Die Kontrollsequenzen "\p" und "\u" haben fälschlicherweise ein "\r". Die Kontrollsequenzen "\p" und "\u" haben fälschlicherweise ein "\r" angehängt.
- * Das ARexx-Skript für "passives offline" wird jetzt ausgeführt, bevor Miami versucht, eine neue Verbindung aufzubauen.
- * Miami hat noch etwas debugging-Ausgabe erzeugt, wenn auf eine ARexx-Message geantwortet wurde.
- * PPP signalisiert kein "down"-Event mehr an Miami, wenn IPCP den "Open"-Zustand verläßt. sondern nur noch, wenn LCP dies tut. Dadurch sollten einige Probleme mit langsamen Terminalservern beseitigt werden, die versuchen, IPCP neu zu konfigurieren, nachdem die PPP-Konfiguration schon beendet ist.
- * ARexx-Befehle "SHOW", "HIDE" und "GETSETTINGSNAME" implementiert.
- * PPP-Option "Get DNS from IPCP" implementiert. Damit kann man die DNS-Erkennung per IPCP abschalten. Einige defekte PPP-Server reagieren auf IPCP-DNS-Erweiterungen weder mit korrekten Daten, noch mit einem korrekten "Reject".
- * Miami fragt jetzt explizit nach MUI >=3.3.
- * Ein Timing-Problem in der unregistrierten Version konnte dazu führen, daß Miami versucht, den gleichen Puffer immer wieder zu allokatieren, ohne ihn freizugeben, so daß Miami sehr schnell alles verfügbare RAM verbraucht hat.
- * Wenn ToolTypes benutzt wurden, um die MiamiInit-Konfiguration oder eine ASCII-Konfiguration zu importieren, wurde die interne Datenbank nicht korrekt initialisiert.
- * Einige Kombinationen von Slirp/TIA mit Slip/PPP wurden nach dem Importieren der MiamiInit-Konfiguration in Miami nicht korrekt konfiguriert.
- * Kleinere Änderungen in MiamiInit.
- * Ein kleines Speicherleck im eingebauten InetD beseitigt.

Version 1.0

erste öffentliche Version

1.116 Miami_deutsch.guide/NODE_FUTURE

Die Zukunft

Meine kurzfristigeren Pläne für Miami beinhalten:

- * Transparente Unterstützung für SOCKS V4 and V5 Clients.
- * Ein neues API zur Handhabung von Online/Offline-Übergängen, kontrolliert von Clients.

ISDN ist ein anderer Schwerpunkt. Externe ISDN-Terminal-Adapter werden schon vollständig unterstützt. Das gleiche gilt für die ISDN-Master-Karte in den meisten Modi, mit Ausnahme von "synchronem PPP im HDLC-Modus". Dieser Modus erfordert neue Treiber und ein neues API, aber einige Leute arbeiten daran...

Die einzige größere (absichtliche) Einschränkung von Miami ist, daß nur eine einzige Schnittstelle genutzt werden kann. Ein Nachfolgeprogramm "Miami Deluxe" ist für später dieses Jahr geplant. Meine derzeitigen Pläne für Miami Deluxe beinhalten Unterstützung für PPP-Dial-in-Verbindungen, mehrere Schnittstellen, und vermutlich viele andere Funktionen, die für Router-Betrieb nützlich sind, wie Socks-Daemon-Unterstützung, IP-Masquerading, und Firewall-Funktionen.

Es wird für registrierte Miami-Anwender einen Rabatt bei der Registrierung von Miami Deluxe geben.

1.117 Miami_deutsch.guide/NODE_SUPPORT

Unterstützung

Es gibt verschiedene Möglichkeiten, technische Unterstützung und Updates zu bekommen:

Email

kruse@nordicglobal.com

normale Post

Holger Kruse
c/o Nordic Global Inc.
P.O. Box 780248
Orlando FL 32878-0248
USA

WWW

<http://www.nordicglobal.com/Miami.html>

Mailing-Listen

Schicken Sie "SUBSCRIBE miami-announce-ml" oder "SUBSCRIBE miami-talk-ml" im Text einer Email an

"amiga-lists@nordicglobal.com".

1.118 Miami_deutsch.guide/NODE_ACKNOWLEDGEMENTS

Danksagungen

Herzlichen Dank an

- * Die Tester früher Alpha- und Beta-Versionen Karl Bellve, Mike Fitzgerald, Adam Hough, Daniel Saxer, Stefan Stuntz und Oliver Wagner.
 - * Karl Bellve und Daniel Saxer für Ihre große Unterstützung.
 - * NSDi für das erste frei erhältliche TCP/IP-Protokoll-System für AmigaOS und eine sehr brauchbare Programmierschnittstelle.
 - * James Cooper, Steve Krueger und Doug Walker für das SAS/C Entwicklungssystem und Ihre große Unterstützung.
 - * Stefan Stuntz für sein hübsches User-Interface-Paket MUI.
 - * Klaus Melchior für die MUI custom class "Busy.mcc".
 - * Robert Reiswig für das Ausleihen von Computerhardware.
 - * der University of California für ihre erfolgreiche Arbeit an dem exzellenten BSD-Netzwerk-Code.
 - * Reinhard Spisser und Sebastiano Vigna for ihre Amiga-Portierung von "makeinfo".
 - * Paul Trauth, dem Sieger des Miami-Logo-Contest, for seine hübsche Image-Sammlung.
 - * John Pszeniczny for seine Variationen des "Miami"-Logos.
 - * allen Anwendern, die sich dazu entscheiden, sich für Miami zu registrieren.
-